UNIVERSITY LUMIÈRE LYON 2

DOCTORAL SCHOOL INFOMATHS

INFORMATIQUE ET MATHÉMATIQUES (ED 512)

P H D T H E S I S

to obtain the title of

PhD of Science

Specialty : Computer Science

Defended by

Marian-Andrei RIZOIU

Semi-supervised structuring of complex data

Thesis Advisers : Stéphane LALLICH and Julien VELCIN

prepared at the ERIC laboratory

defended on June 24^{th}, 2013

Jury :

Reviewers :	Adrião Dória Neto	-	Universidade Federal do Rio Grande do Norte
	Christel Vrain	-	LIFO Laboratory, Univ. Orleans
Advisers :	Stéphane Lallich	-	ERIC Laboratory, Univ. Lyon 2
	Julien Velcin	-	ERIC Laboratory, Univ. Lyon 2
Examiners :	Maria Rifqi	-	LIP6 Laboratory, Univ. Panthéon-Assas
	Frédéric Precioso	-	I3S Laboratory, Univ. Nice Sophia Antipolis
	Gilbert Ritschard	-	Institute IDEMO, Univ. de Genève

Semi-Supervised Structuring of Complex Data

Abstract :

The objective of this thesis is to explore how complex data can be treated using unsupervised machine learning techniques, in which additional information is injected to guide the exploratory process. The two main research challenges addressed in this work are (a) leveraging semantic information into data numerical representation and into the learning algorithms and (b) making use of the temporal dimension when analyzing complex data. The main research challenges are derived, through a dialectical relation between theory and practice, into more specific learning tasks, which vary from (i) detecting typical evolution patterns to (ii) improving data representation by using semantics to (iii) embedding expert information into image numerical description or to (iv) using semantic resources (*e.g.*, WordNet) when evaluation topics extracted from text. The methods we privilege when tackling with our learning tasks are unsupervised and, mainly, semi-supervised clustering. Therefore, the general context of this thesis lies at the intersection of the two large domains of complex data analysis and semi-supervised clustering.

We divide our work into four parts. The first is dedicated to the temporal component of data, in which we propose a temporal clustering algorithms, with contiguity constraints, and use it to detect typical evolutions. The second part is dedicated to semantic reconstruction of the description space of the data, and we propose an unsupervised feature construction algorithm, which replaces highly correlated pairs of features with conjunctions of literals. In the third part, we tackle the problem of constructing a semantically-enriched image representation starting from a baseline representation, and we propose two approaches toward leveraging external expert knowledge, under the form of non-positional labels. We dedicate the fourth part of our work to textual data, and more precisely towards the task of topic extraction, using an overlapping text clustering algorithm, topic labeling, using frequent complete phrases, and semantic topic evaluation, by using an external concept hierarchy. We add a fifth part, which describes the applied part of our work, `CommentWatcher`, an open-source platform for analyzing online discussion forums.

Keywords : complex data analysis, semi-supervised clustering, semantic data representation, temporal clustering, topic extraction, semantic-enriched image representation, feature construction

Structuration semi-supervisée des données complexes

Résumé :

L'objectif du travail présenté dans cette thèse est d'explorer comment les données complexes peuvent être analysées en utilisant des techniques d'apprentissage automatique non-supervisé, dans lequel des connaissances supplémentaires sont introduits pour guider le processus exploratoire. Ce travail de recherche traite deux grandes problématiques : d'une part, l'utilisation d'informations sémantiques dans la construction de la représentation numérique ainsi que dans les algorithmes d'apprentissage automatique, et d'autre part, l'utilisation de la dimension temporelle dans l'analyse de données complexes. De ces problématiques de recherche ont émergé, au travers d'une relation dialectique entre la théorie et la pratique, des tâches plus précises, à savoir : (i) la détection d'évolutions typiques, (ii) l'amélioration de la représentation des données en utilisant leur sémantique, (iii) l'introduction d'informations expertes dans la représentation numérique des images et (iv) l'utilisation de ressources sémantiques additionnelles (comme WordNet) pour l'évaluation des thématiques extraites à partir du texte. Les méthodes qu'on privilégie dans notre travail sont des méthodes non-supervisées et, notamment, des méthodes semi-supervisées. Par conséquent, le contexte générale de cette thèse cette thèse se situe à au croisement des domaines de l'analyse de données complexes et du clustering semi-supervisé.

Nous divisons notre travail en quatre parties. La première partie est dédiée à la dimension temporelle des données, et dans cette optique nous proposons un algorithme de clustering temporel avec des contraintes de contiguïté, que nous appliquons à la détection d'évolutions typiques. La deuxième partie quant à elle s'intéresse à la reconstruction sémantique de l'espace de représentation, tâche pour laquelle nous proposons un algorithme non-supervisé de construction d'attributs, dont le principe de base est de remplacer les paires d'attributs hautement corrélés par des conjonctions de ceux-ci. Dans la troisième partie, nous traitons le problème de la construction de représentations numériques sémantiquement enrichies des images et pour cela nous proposons deux approches qui utilisent des connaissances expertes sous la forme d'annotations. Enfin, la dernière partie de nos travaux théoriques est dédiée aux données textuelles, plus précisément aux tâches d'extraction de thématiques à l'aide d'un algorithme de clustering avec recouvrement, de nommage de thématiques par des expressions intelligibles, ainsi qu'à l'évaluation sémantique des thématiques, en utilisant une hiérarchie de concepts. A ce travail vient s'ajouter une cinquième partie pratique, dont l'aboutissement est la plateforme `CommentWatcher` qui permet d'analyser les forums de discussion en ligne.

Mots clés : analyse de données complexes, clustering semi-supervisé, représentation sémantique de données, clustering temporel, extraction de thématiques, représentation des images enrichi avec de la sémantique, construction non-supervisée des attributs

Acknowledgments

First of all, I would like to thank my two PhD advisers, Stéphane Lallich and Julien Velcin, for accompanying me on this four-year journey. They have shown me what research means and they have taught me how to appreciate it. They have shown me how one can be rigorous, while creative, and they have become, in the meanwhile, my role-models.

I would also like to thank Christel Vrain and Adrião Dória Neto for accepting to dedicate their time and energy towards reviewing my thesis. I equally thank the examiners in my jury, Maria Rifqi, Frédéric Precioso and Gilbert Ritschard, for doing me the honor of participating to the jury of my thesis.

A special thanks goes towards Julien Ah-Pine, who, in addition to being a good and supportive friend, has kindly accepted to read this manuscript and to provide educated opinions, which often made me see things in a new light. I also acknowledge my work colleagues and friends, Adrien and Mathilde, who transformed the office into an exhilarating place.

And last, but not least, I thank my friends and family for putting up with me.

Contents

Chapter 1

Introduction

Contents

1.1 The big picture

The early stages of the Web (*i.e.*, the *Web 1.0*) was made out of static pages, user could consult their content, but not contribute to it. The *Web 2.0* allowed users to interact and collaborate with each, while dynamically generating the content of web pages. The new paradigm contributed to the change of the way in which information is produced, shared and consumed. Users read, watch, listen existing material, then they react, post, describe and tag, therefore enriching the available information. All this freely accessible information is a non-exhaustible source of data. Internet-originating data is just one example of a broader class of data, called **complex data**. Complex data are heterogeneous data (*e.g.*, text, images, video, audio *etc.*), which are further interlinked through the structure of the complex document (*i.e.*, the webpage, in the case of Internet) in which they reside. These data have a big dimensionality and very often they have a temporal dimension attached. The temporal aspect is particularly important for news articles or online social network postings.

The difficulties of dealing with the complex data originating from the *Web 2.0* (*i.e.*, the immense quantities of unstructured and semi-structured heterogeneous data) are the central points of the main applications related to the Internet, such as *Information Search and Retrieval* (finding useful information in the enormous amounts of available data is still the most prevalent user task on the internet), *Categorization* (a collective effort to organize the available information, *e.g.*, folksonomies such as Delicious[1]), or *Recommender Systems* (recommending new content based on the habits of the user inferred from the currently viewed content).

The difficulties introduced by the *Web 2.0* led to the emergence of the *Semantic Web*[2], which is linked to converting the current unstructured and semi-structured documents into a "Web of data", by including machine-readable semantic content into web pages. The purpose of the *Semantic Web* is to provide a common framework that allows information to be shared

1. https://delicious.com/
2. *Semantic Web* and *Web 3.0* are often used as synonyms, their definition is not yet standardized.

and reused across application, enterprise and community boundaries. It involves publishing in languages specifically designed for data (such RDF [3], OWL [4] and XML [5]). The machine-readable descriptions enable content managers to add meaning to the content. In this way, the machine can process information at a semantic level, instead of text, thereby obtaining more meaningful results. This semantic information is gathered in knowledge repositories, such as freely accessible ontologies (*e.g.*, DBpedia [6] [Bizer *et al.* 2009], Freebase [7]). One of the main challenges of the Semantic Web is obtaining a semantic representation of data. The main problem of representing data of different natures (*e.g.*, image, text) is that low-level features used to digitally represent data are far removed from the semantics of the content.

Our work: research challenges and privileged methods. The main research challenge of the work presented in this thesis is **leveraging semantics when dealing with complex data**. Chapters 4, 5 and 6 approach the problems of introducing human knowledge (*e.g.*, labels, knowledge repositories) into the learning process and semantically reconstructing the description space of data. We distinguish between two sub-challenges: (i) *translating data into a semantic-aware representation space*, which deals with constructing a representation space that better embeds the semantics and which can be used directly with classical machine learning algorithm, and (ii) *injecting knowledge into machine learning algorithms*, which deals with modifying the machine learning algorithms so that they take into account semantics while inferring knowledge.

The second research challenge of this thesis is **leveraging the temporal dimension of complex data**. The temporal dimension is more than just another descriptive dimension of data, since it profoundly changes the learning problem. The description of data becomes contextualized (*i.e.*, a certain description is true during a given time frame) and new learning problems arise: following the temporal evolution of individuals, detecting trends, topic burstiness, popular events tracking, *etc.* The temporal dimension is intimately related to the interactive aspect of the Web 2.0. We approach the temporal dimension in Chapter 3, where we develop a clustering algorithm in which we take time into account to construct temporally coherent clusters. In the work presented in this thesis, we deal with each of these research challenges individually. We currently have undergoing work (detailed in Chapter 8), which will allow to integrate together our two research challenges.

The methods we privilege when tackling with our two major research problems are unsupervised and, mainly, **semi-supervised clustering**. Semi-supervised clustering [Davidson & Basu 2007] is essentially an unsupervised learning technique, in which partial knowledge is leveraged in order to guide the clustering process. Unlike semi-supervised learning [Chapelle *et al.* 2006], where the accent is on dealing with missing data in supervised algorithms, semi-supervised clustering is used when the expert knowledge is incomplete or in such low quantity that it would be impossible to apply supervised techniques. We use semi-supervised partial knowledge to model the semantic information and the temporal dimension when analyzing complex data. Therefore, the general context of this thesis lies at the intersection

3. http://www.w3.org/RDF/
4. http://www.w3.org/OWL/
5. http://www.w3.org/XML/
6. http://www.dbpedia.org
7. http://www.freebase.com/

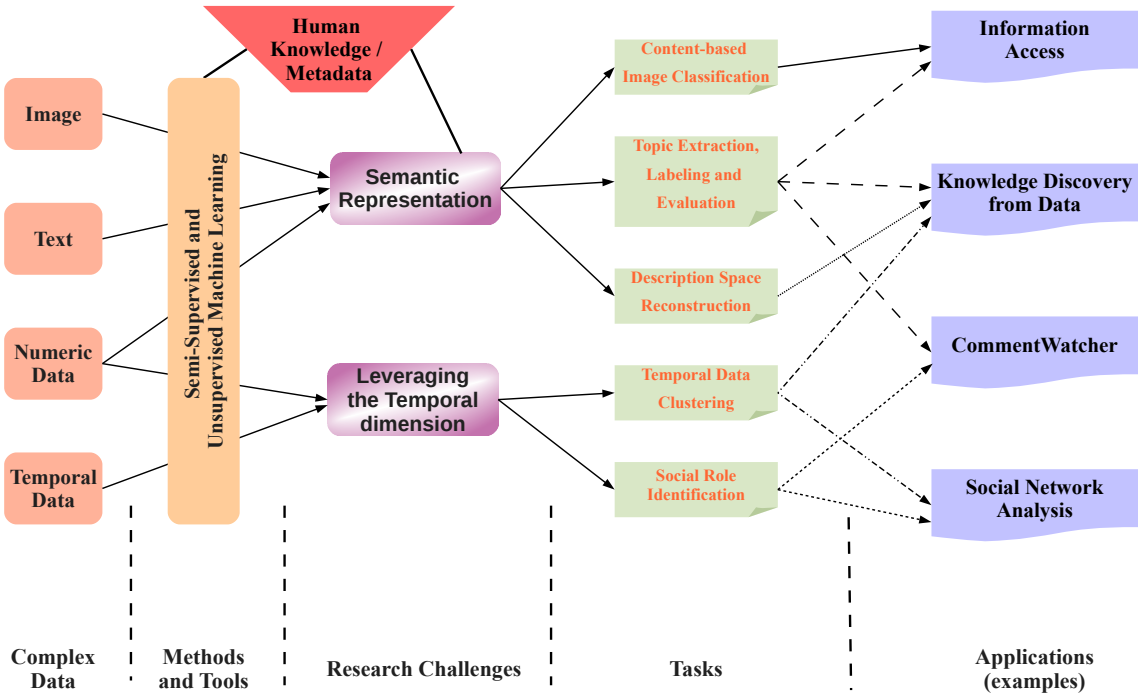


Figure 1.1 – The schema structuring the work in this thesis: starting with the input complex data and ending with examples of applications.

of the two large domains of **complex data analysis** and **semi-supervised clustering**.

The remainder of this chapter presents an overview of the research project and the detailed motivations of our work (in Section 1.2), we describe the organization of the different parts of our work (in Section 1.3), followed by a detailed plan of the manuscript (in Section 1.4). The research work presented in this thesis was performed at the ERIC Laboratory in Lyon, France, in the Data Mining and Decision team.

1.2 Research project

Complex Data Mining is a very vast domain, touching Computer Vision, Natural Language Processing, Artificial Intelligence and even Sociology (*e.g.*, constructing and analyzing online social networks). I have, therefore, derived the two central research challenges and focused on more specific learning tasks, which vary from (i) *detecting typical evolution patterns* to (ii) *improving data representation by using semantics* or to (iii) *embedding expert information into image numerical description*. My research project was built incrementally, through a dialectical relation between theory and practice. The two advanced together, influencing each other along the process. The before mentioned learning tasks I interested in, are partially motivated by the specific problems and applications needed by the different research projects in which I was involved (see in Appendix A).

Figure 1.1 presents the schema of the work presented in this manuscript. We start from a subset of types of complex data (*i.e.*, we interest in text, image, numeric data and data with a temporal dimension) (on the left of the schema). In the work we performed, we

analyze each type of data independently. A perspective of our work is a broader integration of all the information provided by complex data, in order to take profit from every available piece of information. At the right side of the schema in Figure 1.1 are the final abstract applications of our work, such as *Information Access*, *Knowledge Discovery from Data*, *Social Network Analysis* or `CommentWatcher`, an online media analysis tool, the result of our applied work. In between we present, from right to left (from the purpose of our work, *i.e.*, the output, to the input), (a) the more specific learning tasks we approach in our work, (b) the research challenges that derive in the learning tasks and (c) methods and tools we privilege in order to attain our research challenges. The arrows indicate how the different parts were used in our research. For example, we interest in leveraging semantics into the construction of the representation for images, text or numeric data. Similarly, the research challenge of *leveraging the temporal dimension* is derived into the two specific tasks of *temporal data clustering* and *social role identification*. These can be, in turn, used in application as *Knowledge Discovery from Data*, *Social Network Analysis* or into `CommentWatcher`, the developed software.

The motivations of our work The motivations behind our work can be resumed at different abstraction levels.

On an **applied level**, our work is related to the various research needs of the projects I was involved (more details in Annex A). The task of *detecting typical evolution patterns* is in relation with the interest of researched in Political Sciences, involved in the ImagiWeb project [8]. Another example is the multi-sided link between the research collaboration with the Technicolor laboratories [9], the Crtt-Eric project, our work concerning the textual dimension, the task of *Social Role Identification* and `CommentWatcher` (which presented in detail in Chapter 7).

On the **problems and solutions level**, our work was motivated by the need to propose solutions for a series of specific learning tasks. We proposed new algorithms (*e.g.*, the temporal clustering algorithms **TDCK-Means**, the feature construction algorithm **uFC**), new measures (*e.g.*, the temporal-aware dissimilarity measure), parameter choice heuristics (*e.g.*, the χ^2 hypothesis testing-based heuristic), *etc.*

On the **research challenges level**, at the core of our research work are the two research challenges detailed earlier: (a) embedding semantics into data representation and machine learning algorithms and (b) leveraging the temporal dimension.

The **abstract applications level**. At a meta level, our work is motivated and can be used in application as Information Access, Knowledge Discovery from Data or Social Network Analysis.

The three most important original ideas in our work Throughout this manuscript, the reader will find a number of original proposals. In the following, we single out three of the most important ideas of our research.

Taking into account both the temporal dimension and the descriptive dimension into a clustering framework. The resulted clusters are coherent from both the temporal and the

8. http://eric.univ-lyon2.fr/~jvelcin/imagiweb/
9. https://research.technicolor.com/rennes/

descriptive point of view. Constraints are added to ensure the entity segmentation contiguity.

Unsupervised construction of a feature set based on the co-occurrences issued from the dataset. This allows adapting a feature set to the dataset's semantics. The new features are constructed as conjunctions of the initial features and their negations, which renders the result comprehensible for the human reader.

Using non-positional user labels (denoting objects) to filter irrelevant visual features and to construct a semantically aware visual vocabulary for a "bag-of-feature" image representation. We use the information about the presence of objects in images to detect and remove features unlikely to belong to the given object. Dedicated visual vocabularies are constructed, resulting in a numerical description which yields higher object categorization accuracy.

1.3 The constituent parts of the thesis

Given the great diversity of the approached subjects, I divide my work into four distinct, yet complementary parts. The four parts deal, respectively, with (a) the temporal dimension, (b) semantic data representation, and the different natures of complex data, *i.e.*, (c) image and (d) text. Each part is dealt with in an individual chapter, which contains an overview of the state of the art of the domain, the proposals, conclusions about the work and some plans for future work. A fifth chapter is dedicated to the practical aspects of my work, most notably `CommentWatcher`, an open-source platform for online discussion analysis. Therefore, each of the five chapters can be seen as autonomous, while remaining connected the **directive guidelines**, the **transverse links** between them and the **conceptual articulation**. Each of these is further detailed in the following paragraphs.

Directive guidelines The core research challenges are translated into directive guidelines, that run throughout my research: (i) human comprehension, (ii) translating data of different natures into a semantic-aware description space and (iii) devising algorithms and methods that embed semantics and the temporal component.

In each of our proposals, we consider crucial to generate **human comprehensible outputs**. Black-box approaches exist for many problems (*e.g.*, Principal Component Analysis is a solution for re-organizing the description space), but the semantic meaning of their output is not always clear and, therefore, the latter are difficult to interpret. Our proposals are developed with human comprehensibility in mind.

Another directive guideline of our work is **translating data of different natures into a semantic-aware description space**, which we call throughout this manuscript the Numeric Vectorial Space. Constructing such a description space usually consists in (a) rendering the data into a common usable numeric format, which succeeds in capturing the information present in the native format, and in (b) efficiently using external information for improving the numeric representation.

Finally, a central axis of our research is **devising algorithms and methods that embed semantics and the temporal component**, based on unsupervised and semi-supervised techniques. Often additional information and knowledge is attached to the data, under the form of (a) user labels, (b) structure of interconnected documents or (c) external

knowledge bases. We use this additional knowledge at multiple instances, usually using semi-supervised clustering techniques. We also use semi-supervised constraints to model the temporal dependencies in the data.

Transverse links There are multiple **transverse links** between the individual parts of our research. Our work with textual data is intimately linked with the software `CommentWatcher`. The text from online discussion forums is retrieved, we extract topics from it and infer a social network using the forum's reply-to relation. The social network is modeled and visualized as a multidigraph, in which links between nodes are associated to topics. Furthermore, the temporal-driven clustering we propose is applied to detect social roles in the social network. Another transverse link concerns our feature construction algorithms, which was initially motivated by the need to re-organize the user label set we use to create the semantic-enabled image representation. We also have ongoing work which deals with embedding the temporal dimension into this feature construction algorithm. The idea is to detect if features are correlated with a certain time lag.

Conceptual articulation of the different parts It is noteworthy that the work presented in this thesis is not a blueprint of an integrated complex data analysis system. Realizing such a system would have been possible in the context of a very specific (applied) problem, which is not the case of the different collaborations and projects in which I was involved. Whatsoever, a conceptual articulation exists between all the parts of our work: data of different natures is translated into a common semantic-aware numeric format, which is afterwards used together with the temporal dimension or with external knowledge bases. During the next chapters, we evolve the schematic representation of our work in Figure 1.1 to a complete conceptual integration of our proposals. At the end of each chapter, the reader is shown how the work presented in the given chapter can be conceptually integrated with the work in previous chapters. We incrementally evolve the schema in Figure 1.1 in Figures 3.1 (p. 30), 4.17 (p. 93), 5.13 (p. 122) and 6.13 (p. 165) into the complete schema in Figure 8.1 (p. 182).

1.4 Content of the different chapters

Excepting the current chapter, this manuscript is structured over six chapters, as follows.

In **Chapter 2**, we present a general overview of complex data mining. Starting from the specificities of complex data, we identify some of the difficulties of analyzing them and we present some of the solutions existing in the literature. We present the field of semi-supervised clustering in a similar fashion: we start from the necessities of semi-supervised clustering, the advantages and difficulties. We present the taxonomy and briefly present some of the most relevant existing approaches. All along this chapter, we position our work in the broader context of these two domains.

In **Chapter 3**, we leverage the temporal dimension of the complex data and we apply our proposals to the learning task of *detecting typical evolution patterns*. We propose a new temporal-aware dissimilarity measure and a segmentation contiguity penalty function. We combine the temporal dimension of complex data with a semi-supervised clus-

tering technique. We propose a novel time-driven constrained clustering algorithm, called TDCK-Means, which creates a partition of coherent clusters, both in the multidimensional space and in the temporal space. We also show how this temporal clustering algorithm can be applied to a different task: *finding behavioral roles in an online community*.

In **Chapter 4**, we regroup our research concerning the task of *semantic description space reconstruction*. We seek to construct, in an unsupervised way, a new description space which embeds some of the semantics present in a given dataset. The constructed features (*i.e.*, the dimensions of the new description space) are, at the same time, comprehensible for a human user. We propose two algorithms that construct the new features as conjunctions of the initial primitive features or their negations. The generated feature sets have reduced correlations between features and succeed in catching some of the hidden relations between individuals in a dataset. We also propose a method based on statistical testing for setting the values of parameters.

Chapter 5 presents our research concerning image data. We are particularly interested in the task of *improving image representation using semi-supervised visual vocabulary construction*. We present the "bag-of-features" representation, one of the most widely used methods for translating images from their native format to a numeric Vectorial Space. We are interested in using expert knowledge, under the form of non-positional labels attached to the images, in the process of creating the numerical representation. We propose two approaches: the first one is a label-based visual vocabulary construction algorithm, while the second deals with filtering the irrelevant features for a given object, in order to improve object categorization accuracy.

Chapter 6 presents in detail our research concerning textual data, and more precisely, we are interested in the task of *topic extraction and evaluation*. After a presentation of the "bag-of-words" representation, we make an in-depth review of topic extraction and evaluation literature, while referencing methods related to our general domain of interest (*e.g.*, incorporating the temporal dimension or external semantic knowledge). We complete this bibliographic research with the presentation of a textual clustering-based topic extraction system and a topic evaluation systems based on an external semantic knowledge base. At the end of the chapter, we present some applications of this system to the Ontology Learning process, and to topic improvement by removing spurious words.

Chapter 7 presents the practical prototype production. The most prominent produced software is `CommentWatcher`, an open source tool for analyzing discussions on web forums. Constructed as a web platform, `CommentWatcher` features (i) automatic fetching of forums, using a versatile parser architecture, (ii) topic extraction from a selection of texts and (iii) a temporal visualization of extracted topics and the underlying social network of users. It aims both the media watchers (it allows quick identification of important subjects in the forums and user interest) and the researchers in social media (who can use it to constitute temporal textual datasets).

In the last chapter, **Chapter 8**, we draw some general conclusions about our work. We also present in this chapter the work we are currently undergoing and plan other research of near term and long term future.

Chapter 2

Overview of the Domain

Contents

The purpose of this chapter is to present a general overview and familiarize our reader, if not already the case, with the two large domains around which our work revolves: Complex Data Mining (in Section 2.1) and Semi-Supervised Clustering (in Section 2.2). We discuss for each domain the motivations, the difficulties that arise and some of the solutions present in the literature. All along this chapter, we relate our work to the domain and position our proposals relative to existing solutions.

2.1 Complex Data Mining

In this section, we present a general overview of the domain of Complex Data Mining. Using the example of a Wikipedia article, we incrementally single out the particularities of complex data. In Section 2.1.1, we identify and summarize the most important five specificities that define complex data, and we point out how our proposals address them. We further detail some of the identified specificities (in Sections 2.1.2, 2.1.3 and 2.1.4), by presenting the difficulties they pose and some of the existing solutions in the literature. Each of these subsections ends with a paragraph in which we position our work.

Complex Data Mining is a very vast domain, incorporating a large range of related problems. A definition of Data Mining is the computational process of discovering patterns in large data sets involving methods issued from the domains of artificial intelligence, machine learning, statistics, and database systems. Complex Data Mining is the application of Data Mining to Complex Data, *i.e.*, data with a series of particularities that we identify later in this section, and which is a non-standard input for classical data mining algorithms.

Vectorial description space. The input data format for classical data mining algorithms is the (attribute, value) pair format. In this format, each individual is described by a set of measurements over a set of attributes. Each measurement is (a) a real value (**numeric** attributes), (b) a choice from a set of available options (**categorical** attributes) or (c) a value of true or false (**boolean** attributes). When considering the numeric variables, this format can be associated with a multidimensional **vectorial description space**, in which each individual is described by its measurements vector. The assumption is that, in this multidimensional description vectorial space, machine learning algorithms are efficient (*e.g.*, it is separable by means of a classification algorithm).

Complex Data Complex data are profoundly heterogeneous data. Excepting the classic (attribute, value) numeric format, a complex document can contain data of different natures (*e.g.*, text, images, video, audio *etc.*). These data are interlinked through the structure (*e.g.*, titles, paragraphs, sections) of the complex document in which they reside. In addition, complex data can have expert knowledge attached (*e.g.*, expert categorize documents using labels). Sometimes, complex data have attached *a temporal dimension*: it either records the evolution of an entity/object over time, or the complex document suffers modifications over time.

Each of the evoked particularities are facets of the considered complex document and they all must be taken into account in the learning process in order to infer complete knowledge [Zighed *et al.* 2009]. On a more abstract level, *semantics are the main challenge when dealing with complex data*. Complex data come in high volumes and many different types, and the main objective is to piece together the underlying knowledge and recreate the semantic links. Semantics are crucial for the comprehension of the generated results, especially from a human point of view. The semantic representation of complex data can be improved by using the freely accessible resources of the new Semantic Web. Knowledge repositories are increasingly available, most often under the form of (a) ontologies, such as the general purpose DBpedia [1] [Bizer *et al.* 2009] and Freebase [2], which are further interlinked by projects such as *Linking Open Data* [3] of (b) more specialized datasets, issued from the domains of Social Sciences and Humanities (*e.g.*, History, Communication sciences, Sociology *etc.*). It is not uncommon to tap into these distributed external repositories in order to introduce semantics into the learning process. Leveraging semantics and the temporal dimension into the analysis of complex data are the main research challenges of our work, presented in this thesis.

Let's take an example! Figure 2.1 depicts the complex document which is the Wikipedia article [4] about the country of France. The different components of the complex document are highlighted and numbered. The documents is **structured**, having a title (denoted by number 6), subtitles, a table of content *etc.* The main description is in the main column, while additional information is given in the side (right) column. In some applications, additional information can be derived from the structure of the complex documents (*e.g.*, the

1. http://www.dbpedia.org
2. http://www.freebase.com/
3. http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
4. http://en.wikipedia.org/wiki/France

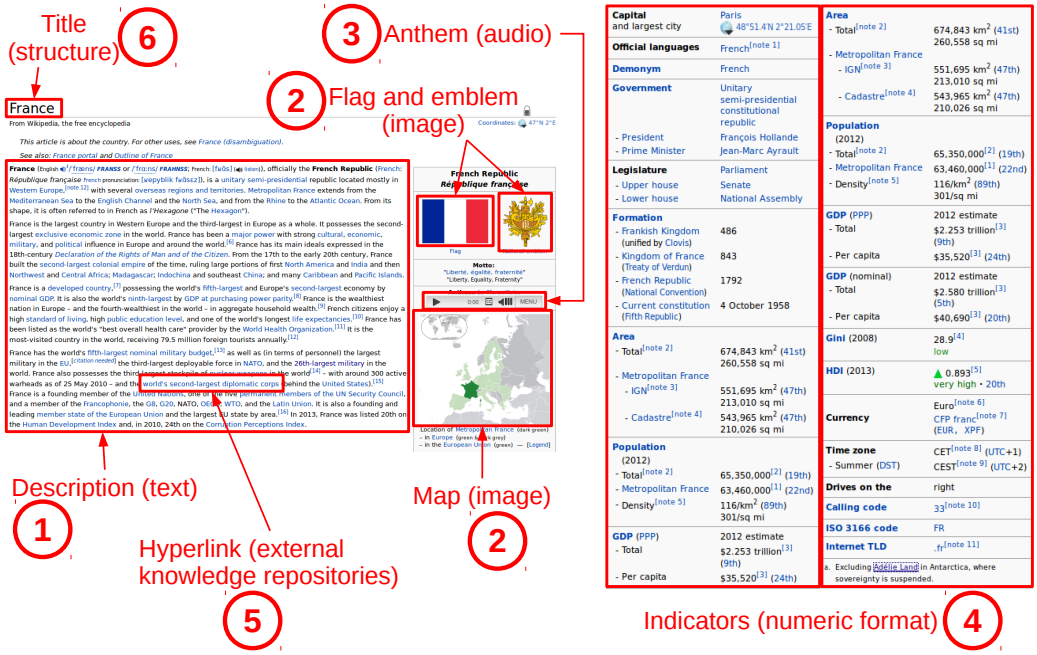


Figure 2.1 – A Wikipedia article is a complex document, containing text (1), images (2), audio (3), numeric indicators (4), links to other pages (5) and a structure (6).

structure of a social network, *i.e.*, how users are interlinked, is sometimes more informative than the content posted by the users).

In Figure 2.1, **text** (numbered 1) is mainly used to give the information about geographic position, history, political system *etc.* Data of other natures could be used to complete the information. **Images** (number 2) are added to portray the country's flag, emblem and geographic map. **Audio data** (number 3) is included to give the national anthem and even **video** is used to present specific events (*e.g.*, in the original Wikipedia article, a video sequence is used to show the French territorial evolution from 985 to 1947). Information like the country's surface, population, gross domestic product (GDP), geographic coordinates *etc.* are given in an (attribute, value) **numerical format** (number 4). Finally, hyperlinks (number 5) are present in the text, most often linking this article with other articles. Hyperlinks can also point towards **external resources** in a knowledge ontology, therefore linking the complex document to structured information and, also, with more semantics. The complex data can also have **a temporal dimension**. In the Wikipedia article, the data can be updated yearly with the latest information about political events. Furthermore, the track of past values for social and economic indicators are good hints for current events (*e.g.*, high levels of debt and leverage in the banking system were early indicators of the economic crisis of 2008).

2.1.1 Specificities of complex data

In the following, we summarize and structure the before mentioned specificities of complex data.

– **diverse nature of data.** (text, image or audio/video) Dealing with non-numeric

data raises problems, out of which we mention the fact that (a) they are not directly "understandable" by a machine (*i.e.*, they need to be translated first to a numerical space) and (b) the numerical space in which they are translated captures few semantic information and, consequently, machine learning algorithms exhibit low performances. Some approaches (more details in Section 2.1.2) deal with this problem by using data of different natures (*e.g.*, images and text) to better guide the learning process.

- **additional information** (*e.g.*, external knowledge) External information or resources might be available to complete the semantic information present in the data. This additional information can be under the form of (a) expert provided tags of labels or (b) interlinked knowledge repositories (*i.e.*, ontologies).
- **temporal/dynamic dimension.** It often happens that the same entity is described according to the same characteristics at different times or different places (*e.g.*, a patient may often consult several doctors, at different moments of time). These different data are associated with the same entity and the complex data describes the evolution of the entity in the given description space. A special kind of temporal data is the dynamic data, which is available as a stream (this data cannot be stored and it must be analyzed online). We give more details about the temporal dimension and dynamic data in Section 2.1.3.
- **high dimensionality.** Taking into account data of multiple natures and external knowledge repositories raises dimensionality problems. This dimensionality problem can either concern the high volumes of data that need to be dealt with (the *"scalability"* problem), or, most often, the high dimensionality of the description space (the *"curse of dimensionality"*). In Section 2.1.4, we present in detail how this problem affects the learning process and some existing solutions.
- **distributed and diverse sources.** The complex data can originate from different sources, which, furthermore, do not need to be collocated. This is not a new problem (*e.g.*, in older times, the same information could be found in different books, in different libraries), but it has been exacerbated with the arrival of the Web 2.0. Information is nowadays essentially distributed into many sources, instead of being centralized in libraries. The retrieval paradigm also shifted from classification (*e.g.*, sorting books in a library based on a set of criteria) to searching (*e.g.*, modern days web-search engines query multiple distributed knowledge repositories to compile an answer).

Positioning our work In our work, we have addressed multiple learning tasks related to the specificities of complex data identified here above. We deal with complex data of two **different natures.** We deal with image data in Chapter 5, in which we propose a method to introduce semantic knowledge into the image numerical representation. We deal with textual data in Chapter 6, in which we present address the tasks of topic extraction, topic labeling and topic evaluation. Topic labeling is important for the human comprehension of extracted labels, whereas for the topic evaluation we employ semantic knowledge.

Leveraging **semantic information** into data numerical representation and into the learning algorithms is one of the central research challenges of this thesis. In Chapter 5, we deal specifically with embedding semantic information under the form of labels into the numeric representation of images. In Chapter 6 we use external semantic resource (*i.e.*, WordNet) for mapping the statistically constructed topics to a semantic-aware structure

and for evaluating and improving topics extracted from text. The second research challenge that we address in our work is the **temporal dimension** of complex data. In Chapter 3 we propose a new temporal-aware constrained clustering algorithm (TDCK-Means), which constructs temporally coherent clusters and contiguously segments the temporal observations belonging to an entity.

In the following subsections, we further detail some of the specificities of complex data (*i.e.*, dealing data of different natures, the temporal dimension and the dimensionality), we show some of the difficulties associated with each one and some of the solutions present in the literature.

2.1.2 Dealing with complex data of different natures

Traditional Data Mining algorithms (*e.g.*, clustering algorithm, classification tree learning algorithms *etc.*) were not designed to deal with data of diverse natures. Text and image are the two natures of data most widely used (for example in Internet), but other are also popular, like the audio and video. The difficulty in analyzing data of different natures is that, while they are easy to be transformed, stored and reproduced into/from a digital format, *this format captures little semantic information* needed by machine learning algorithms. Therefore, one of the main challenges when dealing with data of different natures is to translate them into a semantic-aware numeric description space, on which the results of a machine learning algorithm are "relevant". In our context, we consider results "relevant" when, in addition to being the answer to a given task, they are also comprehensible for a human being. Therefore, we summarize the difficulties related to the most common natures of complex data, as follows:

- **text.** The morphological and syntactic rules of languages are not directly machine "comprehensible". Furthermore, in most representations, the text is encoded at the level of a character and the presence of a given character (*e.g.*, the character 'b' or the space) gives almost no hints about the subject of the text.
- **images.** The native digital format for images is the pixel based format. An image is represented as a matrix of pixels, where each pixel has a certain color. Low level image features (*e.g.*, the pixel's color) capture very little of the semantics of the image (*e.g.*, the objects represented in the image). Passing from low-level features to high-level features, while capturing the semantics of the image, is known as the *semantic gap*.
- **video.** The video is digitally represented as a sequence of images, therefore video data can be considered as image data with a temporal component [Zaiane *et al.* 2003]. Consequently, the difficulties of processing video data inherit the those concerning images, to which new ones are added with respect of the temporal evolution (*e.g.*, tracking objects, finding patterns in the sequence of images *etc.*).
- **audio.** Audio data is digitally represented by the frequencies of the sound present in the audio document. Knowing the presence of a certain frequency in an audio file gives little information about the overall genre of music. Other, more high-level tasks (*e.g.*, speech recognition [de Andrade Bresolin *et al.* 2008]), are even more difficult without thorough pre-processing.

The above mentioned types of data can be translated into a numeric description space, on which classical machine learning algorithms can be applied. The general schema of this

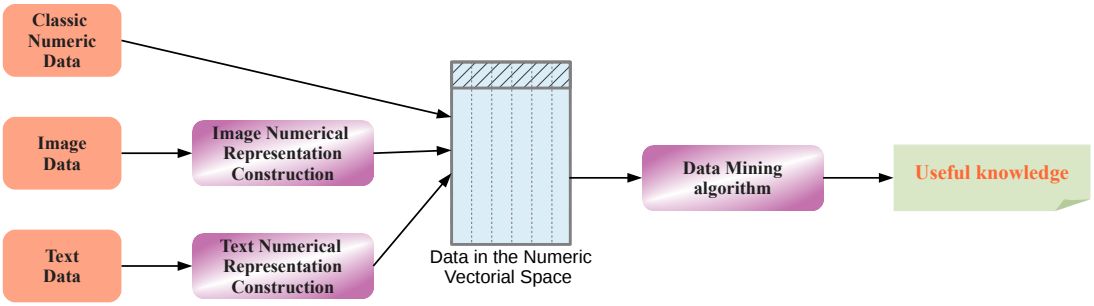


Figure 2.2 – Conceptual schema of how classical numeric data, image data, or text data could be used with a traditional Data Mining algorithm.

process is represented in Figure 2.2. Unlike classic numeric data, data of other types need first to have a numeric representation constructed. The keypoint is to embed enough semantics into the newly created representation so that the results obtained by the machine learning algorithm are "relevant" (as discussed earlier in this section). The schema presented in Figure 2.2 considers the case when knowledge is inferred from only one type of data. Learning simultaneously from data of multiple types is the field of *information fusion* (we discuss using together text and images later in this section).

Most often, **texts** are transformed into a semantically-aware numeric representation by using the "bag-of-features" representation. The underlying assumption is that words that have similar meanings appear often together and that the semantics of a text is captured by the co-occurrence of certain words. This representation is presented in detail in Section 6.2 (p. 128). In a nutshell, after a preprocessing which usually involves removing common words and reducing words to their lemma, texts are represented as an orderless distribution of frequencies over words.

For **images**, a similar representation is used, called the "bag-of-features" representation. This representation is presented in detail in Section 5.1.2 (p. 100). Images are represented similar to texts, with the difference that the place of words is taken by visual words. The visual words are abstractions (normally created though means of clustering) of low-level patch descriptions. They serve a similar purpose as words in the "bag-of-words" representation: they are predictive for the presence of a certain "topic" in an image (*e.g.*, visual words constructed from the patches extracted from an eye are predictors for the presence of an eye in the photo, which itself is a good predictor for the presence of a face and a human).

Using both text and image for learning tasks. Multimedia information is intrinsically multi-modal [Bekkerman & Jeon 2007]. We interpret the word "modality" as the type of input / nature of the data. Learning from only one nature of data at a time (as described earlier and in Chapters 5 and 6) is called an uni-modal approach. The results obtained using uni-modal approaches could be aggregated to infer the overall knowledge, but in practice better results are obtained when using a multi-modal approach [Zaiane *et al.* 2003], in which information from different modalities is simultaneously available to the learning algorithm (*e.g.*, the field of *information fusion*). For example, image captions (*i.e.*, the text associated with an image) and low-level image features are different types of input to an image pro-

cessing system and can, therefore, be considered as two separate modalities. Image captions tend to describe events captured on the image (*i.e.*, they capture semantic information), while image features convey visual information to the system. Consequently, using multiple modalities in the learning process can yield higher performances, as each modality can be used to guide the learning process of another.

We present some of the most common learning tasks that can benefit from using both the text and image natures of data. These tasks share a common trait: low-level visual features (*e.g.*, color, texture, shape, spatial layout, local descriptors *etc.*) used to describe images capture very little of the semantics of images. Their performance can be improved by using text alongside images.

- **Image classification**. Object-based image classification is challenging due to wide variation in object appearance, pose, and illumination effects. Low-level image features are far removed from the semantics of the scene, making it difficult to use them to infer object presence, and it is expensive to obtain enough manually labeled examples from which to learn. To cope with these constraints, text that often accompanies visual data can be leveraged to learn more robust models. Such approaches [Mooney *et al.* 2008, Wang *et al.* 2009] make use of collections of unlabeled images and their textual snippets, usually issued from the Internet.
- **Automatic images categorization**. Unlike the image classification application, in automatic categorization there are no predefined classes. Images are automatically organized based on their similarity through means of clustering. Being fully unsupervised, clustering methods often demonstrate poor performance when performed based only on low-level image features. Clustering results can be improved by using a multi-modal learning paradigm, where image captions and annotations are used alongside image features. Text has been used to help image clustering in number of applications, such as image clustering [Bekkerman & Jeon 2007], in Web image search results clustering [Cai *et al.* 2004] or image sense discrimination [Loeff *et al.* 2006].
- **Improving image numerical description.** Low performances in the tasks described earlier (*i.e.*, image categorization and image clustering) is usually due to the low semantic quality of the image numerical representation: the images are translated into a numerical space which is not easily learnable by machine learning algorithms. Performances can be improved by using the text associated with the images, either in the learning algorithm (as previously seen) or in the creation of the numeric representation [Quattoni *et al.* 2007, Ji *et al.* 2010]. The goal is to embed textual semantics into the image representation and to improve learning in future image-related learning problems. Our work with the images (in Chapter 5), deals with improving the numerical description.
- **Content-based image retrieval.** Content-based image retrieval deals with efficient image searching, browsing and retrieval, with applications in crime prevention, fashion, publishing, medicine, architecture, *etc.* Humans tend to use high-level features (concepts), such as keywords, text descriptors, to search and interpret images and measure their similarity. Features automatically extracted using computer vision techniques are mostly low-level features and, in general, there is no direct link between the high-level concepts and the low-level features [Sethi *et al.* 2001] (also called the "semantic gap"). Some content-based image retrieval systems [Cai *et al.* 2004, Zhuang

et al. 1999] use both the visual content of images and the textual information for bridging the semantic gap. Other techniques exist which do not rely on the usage of associated text [Liu *et al.* 2007] (*i.e.*, (a) using an object ontology, (b) associating low-level features with query concepts, (c) introducing relevance feedback into the retrieval loop or (d) generating semantic template to support high-level image retrieval).

- **Automatic image annotation.** This task deals with automatically annotating images with one or multiple labels, accordingly to their content. One of the difficulties is the huge number of candidate labels and scarce training examples. Automatic image annotation systems [Lu *et al.* 2009, Yang *et al.* 2010] search to find a correspondence between text words and the visual features describing an image. This is most often achieved by using a machine translation approach, where the image-text pairs are seen as bilingual texts and alignment methods are applied [Barnard *et al.* 2003].
- **Human-computer interface systems** use multiple modes of input and output to increase robustness in the presence of noise (*e.g.* by performing audio-visual speech recognition) and to improve the naturalness of the interaction (*e.g.* by allowing gesture input in addition to speech). Such systems often employ classifiers based on supervised learning methods, which require manually labeled data. This usually is costly, especially for systems that must handle multiple users and realistic (noisy) environments. Semi-supervised learning techniques can be leveraged [Christoudias *et al.* 2006] to learn multi-modal (*i.e.*, audio-visual speech and gesture) classifiers, thus eliminating the need of obtaining large amounts of labeled data.

Positioning our work In our work concerning textual data (presented in Chapter 6), we have used a classical "bag-of-words" representation, and concentrated mainly on introducing semantic knowledge into topic evaluation, through a concept-topic mapping. Our work concerning image data (presented in Chapter 5) deals with embedding semantics into the image representation. We show how a semantic-enriched representation can be obtained starting from a baseline representation by employing non-positional labels (*i.e.*, only the presence of objects in the images is known, but not their position).

In our work, we have not dealt with text and image simultaneously. One of the venues in this direction would be to use text instead of non-positional labels. Whatsoever, we use a complete labeling paradigm, in which the absence of a label implies the absence of the object in the image, Passing from the strict labeling to a more relaxed labeling (*i.e.*, authorize missing labels) is one of the perspectives of our work and discussed in Sections 5.3.2 (p. 108). Once this passing done, using the text alongside images is foreseeable. Another research direction would be to embed into images semantic information originating from a concept hierarchy, by passing through text and using the mapping between topics (extracted from text) and concepts.

2.1.3 Temporal/dynamic dimension

Introducing the temporal dimension usually changes the definition of the learning problem: the description of entities in contextualized (*i.e.*, the description is valid for a period of time) and new learning problems emerge, *e.g.*, detecting evolutions and trends, tracking

through time *etc.* Leveraging and interpreting the temporal dimension is one of the central research challenges of this thesis and intrinsically connected to complex data. As discussed earlier, complex data often have a temporal dimension, describing the evolution of a number of entities throughout a period of time. For example, in Chapter 3, we detect typical evolutions of entities and we apply our proposal to a dataset which records the value of a number of socio-economical indicators for a set of countries, over a period of 50 years.

Temporal Data Mining is the sub-domain of Data Mining, closely associated with Complex Data Analysis, which deals with detecting surprising regularities in data with temporal inter-dependencies. Datasets which contain temporal inter-dependencies are called sequential datasets, where sequential data are data ordered by some index. In the case of temporal datasets, the index is the timestamp associated with the observations. Time-series is a popular class of sequential data, which has enjoyed a lot of attention, especially from the statistics community. Time-Series Analysis [Brillinger 2001] has many applications, out of which we mention weather forecast, financial and stock market predictions *etc.* A number of differences exist between Temporal Data Mining and Time-Series Analysis [Laxman & Sastry 2006], the most important being (a) the size and the nature of the studied dataset and (b) the purpose of the study. Temporal Data Analysis deals with prohibitive size datasets, and the data is not always numerical. This are two of the characteristics of complex data. Furthermore, the purpose of Temporal Data Mining goes beyond the forecast of futures values in the series. Some of the learning tasks in Temporal Data Mining can be summarized as follows [Laxman & Sastry 2006]:

- **prediction**. The prediction task has to do with forecasting (typically) future values of the time series based on its past samples. In order to do this, one needs to build a predictive model [Dietterich & Michalski 1985, Hastie *et al.* 2005].
- **classification**. In sequence classification, each sequence is assumed to belong to one of finitely many (predefined) classes or categories and the goal is to automatically determine the corresponding category for the given input sequence. Applications of sequence classification include speech recognition [O'Shaughnessy 2000, Gold *et al.* 2011], gesture recognition [Darrell & Pentland 1993], handwritten word recognition [Plamondon & Srihari 2000].
- **clustering**. Clustering of sequences or time series [Kisilevich *et al.* 2010] is concerned with grouping a collection of time series (or sequences) based on their similarity. Applications are extremely variate and include analyzing patterns in web activity logs, patterns in weather data [Hoffman *et al.* 2008] and trajectories of moving objects [Nanni & Pedreschi 2006], finding similar trends in financial data, regrouping of similar biological sequences like proteins or nucleic acids [Osato *et al.* 2002].
- **search and retrieval**. This task problem is concerned with efficiently locating subsequences in a database of sequences. Query-based searches have been extensively studied in language and automata theory. While the problem of efficiently locating exact matches of substrings is well solved, the situation is quite different when looking for approximate matches [Navarro 2001]. In typical data mining applications like content-based retrieval, it is approximate matching that we are more interested in.
- **pattern discovery**. Unlike the **search and retrieval** task, in pattern discovery there is no specific query with which to search the dataset. The objective is simply to unearth all patterns of interest. Two popular frameworks for frequent pattern dis-

covery are *sequential patterns* [Agrawal & Srikant 1995, Fournier-Viger *et al.* 2011] and *episodes* [Mannila *et al.* 1997]. Sequential pattern mining is essentially an extension of the original association rule mining framework proposed for a database of unordered transaction records [Agrawal *et al.* 1993], which is known as the Apriori algorithm. The extensions deals with incorporating the temporal ordering information into the patterns being discovered. In the frequent episodes framework, we are given a collection of sequences and the task is to discover (ordered) sequences of items (*i.e.* sequential patterns) that occur in sufficiently many of those sequences.

Dynamic data There is fundamental difference between temporal data and dynamic data. Dynamic data are data evolving over time, as new data arrive and old data become obsolete. Furthermore, the updates may change the structures learned so far. Dynamic data is intimately linked with the Internet, where enormous quantities of data are constantly created and need to be dealt with. These data are usually available as a stream, arriving at a rapid rate and cannot be stored for later analyzing.

The temporal-aware algorithms described so far in this section (as well as our proposed TDCK-Means algorithm, which will be described later in Chapter 3) consider the temporal data as recordings of the state of a set of individuals at different moments of time (*e.g.*, macro-economical indicators for countries). These data are stored and studied offline (*a posteriori*), usually for determining causalities, evolutions *etc.* Such "static" algorithms are not appropriate for dynamic data, because (a) huge amounts of data are accumulated over time and cannot be stored, (b) the generative distribution might change over time (which is known in supervised learning as "concept drift"), (c) only one-pass access to the data is available and (d) the data arrives at a rapid rate. Many online algorithms have been proposed, out of which we mention *CluStream* [Aggarwal *et al.* 2003], which performs an online summarization and an offline clustering, and *Single pass K-Means* [Farnstrom *et al.* 2000], which is an extension of K-Means for streams and which performs the clustering in batches that fit into memory.

Positioning our work Our work concerning the temporal dimension does not deal with the dynamic data paradigm, therefore we consider that our dataset is stored in a database and available for querying and transforming to our needs.

The learning task for which our TDCK-Means temporal clustering algorithm was constructed is not a classical clustering task (as defined in Temporal Data Mining), but rather more similar to sequential itemset mining. Among the above defined tasks of Temporal Data Mining, TDCK-Means would be classified as a special case of **pattern discovery**. There is a fundamental difference between the algorithms in the **clustering** category and TDCK-Means: the nature of the individuals that they cluster. For the former, an individual is an entire time-series: they employ similarity measures between time-series and they seek to regroup similar time-series together. For the latter (TDCK-Means), the individuals that it clusters are the observations (*i.e.*, an observation is the description/state of an entity at a given time moment, a tuple $(entity, timestamp, description)$. A time-series is composed of all the observations corresponding to an entity for all the moments of time). TDCK-Means searches to detect groups of similar observations (preferably contiguous, therefore parts of a time-series), in order to detect similar evolutions.

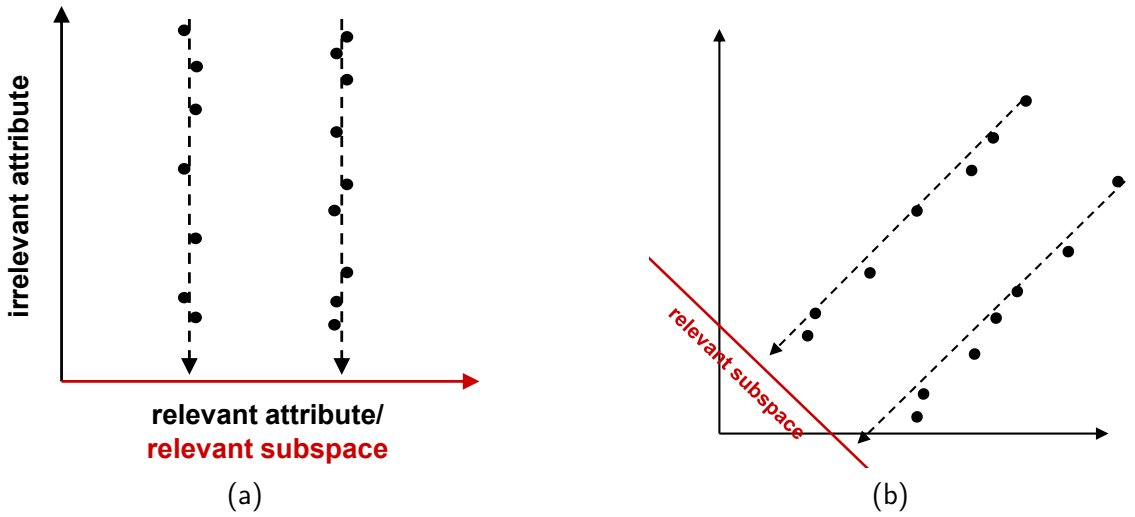


Figure 2.3 – Desired output: (a) the evolution phases and the entity trajectories, (b) the observations of 3 entities contiguously partitioned into 5 clusters.

2.1.4 High data dimensionality

Taking into account data of multiple natures and external knowledge repositories raises dimensionality problems. This dimensionality problems are usually concerning the high dimensionality of the description space. [Kriegel *et al.* 2011] identify 4 problem related with the high dimensionality: (a) the optimization problem, (b) the concentration effect, (c) presence of relevant and irrelevant attributes and (d) the correlation among attributes.

The optimization problem states that the difficulty of any global optimization approach increases exponentially with an increasing number of dimensions [Bellman & Kalaba 1959]. Considering the task of clustering, the fitting of the functions explaining clusters becomes more difficult with more degrees of freedom. **The concentration effect of distances** is identified in [Beyer *et al.* 1999, Hinneburg *et al.* 2000]. As the dimensionality of the description space increases, distances to near and to far neighbors become more and more similar. In other words, the separability of the description space decreases with the increase in the number of dimensions. For a clustering task, this effect has been shown true [Francois *et al.* 2007, Houle *et al.* 2010] only within clusters, but not between different clusters as long as the clusters are well separated. Another problem is the presence of **relevant and irrelevant attributes**. In a large description space, sometimes only a subset of features is relevant for the learning algorithm. For example, in a classification task, the description space depicted in Figure 2.3a is separable only on a subset of features (shown on the horizontal axis). Together with the concentration effect, this problem causes the sharp decrease in the performance of distance function-based learning algorithms that assign equal weights to all dimensions. The forth problems is the **correlation among attributes**. In the context of machine learning (supervised or unsupervised), a useful attribute needs to portray new information. Figure 2.3b shows that case when a subset of attributes are correlated among themselves. Features in the subset depicted on the vertical axis do not bring any new information in the learning process, since their values can be deduced from those depicted on the horizontal axis. Feature correlation only augments the dimension of the

description space, without increasing the relevance of the descriptive space.

Our work presented in Chapter 4 is specifically targeted at the relevance of the descriptive space, with respect to a given dataset. In Section 4.2 (p. 63), we identify three types of solutions for this problem:

- **feature selection.** Feature selection techniques [Lallich & Rakotomalala 2000, Mo & Huang 2011] seek to filter the original feature set in order to remove redundant features.
- **feature extraction.** Feature extraction involves generating a new set of features through means of functional mapping so that the new description space is relevant for the learning task. Examples of such approaches are the SVM's kernel [Cortes & Vapnik 1995] and principal component analysis (PCA) [Dunteman 1989].
- **feature construction.** Feature construction is a process that discovers missing information about the relationships between features. Most constructive induction systems [Pagallo & Haussler 1990, Zheng 1998] construct features as conjunctions or disjunctions of initial attributes.

There are fundamental differences between **feature extraction** and **feature construction** algorithms: (i) the comprehensibility of the new description space, (ii) the underlying purpose of the process and (iii) the dimension of the new space. (i) The description space resulted from feature extraction algorithms is either completely synthetic (for PCA) or hidden/functioning as a black box (for SVM). This renders the interpretation of the results rather difficult. In the case of feature construction, the new attributes are easily comprehensible (*e.g.*, a feature entitled *motorbike and driver* is easier to interpret than the third axis of PCA). (ii) The underlying purpose of feature extraction is just improving the numerical relevance of the description space, whereas the purpose of feature construction is also discovering hidden relations between attributes (*e.g.*, the correlation of *people* and *grass* for a subset of pictures has a certain meaning when portraying a barbecue). (iii) Feature extraction algorithms either output a description space of (a) a lower (*e.g.*, for PCA or Manifold Learning [Huo *et al.* 2006]) or (b) a higher (*e.g.*, the SVM kernel into a very high dimensional space, even an infinitely dimensional space for the RBF kernel [Chang *et al.* 2010]) dimensionality than the original space. Conversely, the feature construction algorithms invariably increase the number of dimensions.

Positioning our work Our work concerning the description space and presented in Chapter 4 proposes a feature construction algorithm for discovering missing semantic links between the attributes describing a dataset. The novelty of the proposed algorithm is that, unlike the rest of the feature construction algorithms present in literature, we construct features in an unsupervised context, based only on the correlations present in the dataset. Our **uFC** algorithm adapts the description space to the semantics of the given dataset. Our experiments in Section 4.6 (p. 75) show that the constructed features are highly comprehensible, while the constructed description space achieves a lesser total correlation between dimensions.

2.2 Semi-Supervised Clustering

In the previous sections, we have shown that complex data often comes bundled with additional information. For example, researchers in human sciences tag documents for easy referencing, medical doctors annotate the files of their patients. The modern Internet is a prolific source of additional information. Many photo sharing online platforms (*e.g.*, Instagram [5], Flickr [6]) allow their users to tag the presence of certain objects or persons in their photos. Furthermore, general purpose knowledge ontologies (*e.g.*, Dbpedia, Freebase) are freely accessible and contain millions of records of general fact, such as information about countries, persons, events, movies *etc.* They are interconnected with more specific ontologies, depicting facts about geo-localization, transport infrastructure *etc.* Together, they form an enormous inter-linked reservoir of knowledge, that can be used as guidance in many learning applications.

Making best use of the knowledge. In traditional machine learning, the learning guidance is performed through supervision, by showing the algorithm a number of correctly classified examples and demanding it to learn to distinguish between classes. This field of Machine Learning is mature enough, and modern classification algorithm (such as SVM [Cortes & Vapnik 1995]) show impressive results. There is a number of applications related to complex data for which the supervised learning paradigm can difficultly be applied because (a) there might not be enough examples for each class, (b) not all classes are known beforehand (sometimes not even the number of classes is known) or (c) class information is not available at all, the only available information is about the relations between a subset of individuals. For example, in an image dataset, the presence of *some* of the objects might be labeled in *some* of the images. For this application, a supervised learning approach cannot be employed to learn how to differentiate between objects, since the quantity of examples for each label is not sufficient and no assumption can be made about the total number of objects.

Leveraging partial expert knowledge is the domain of semi-supervised learning. Partial knowledge is either not complete (*e.g.*, only the relations between certain individuals are known) or simply not enough examples are known for supervised algorithms to function. The domains of semi-supervised learning can be broadly be divided into two categories: semi-supervised classification and semi-supervised clustering. In order to better differentiate the two approaches, we take into account two dimensions: (i) the type of the learning problem to be solved and (ii) the quantity of available supervision. The learning problem can be essentially supervised (*i.e.*, learning to distinguish between predefined classes) or non-supervised (*i.e.*, to discover automatically a typology of the data).

– **semi-supervised classification** [Zhu 2005, Chapelle *et al.* 2006] is an essentially supervised task, and it can be applied to supervised learning problems, in the presence of low quantities of supervision. Such methods can learn from a low number of examples for each category, by enabling the learning from both labeled and unlabeled data. They still inherit from supervised methods a number of restrictions, among which (a)

5. http://instagram.com/
6. http://www.flickr.com/

the number of categories must be fixed and known beforehand and (b) labeled examples must be present for each category. These restrictions can prove to be too severe for complex data, for which the classification typology might not be known beforehand (or not even the number of classes). Furthermore, supervised information might be available only under the form of some pairwise connections (*e.g.*, it can be known the fact that two individuals should be classified together, but no other information is known towards the category under which they should be classified). We, therefore, consider that semi-supervised classification approaches are not completely adapted to complex data.

– **semi-supervised clustering** is an essentially unsupervised task, which is adapted to handle non-supervised learning problems, for which some supervised information is available. These methods deal with using the supervision in order to guide the clustering. Semi-supervised clustering is useful when (i) classes are not known beforehand or not enough examples are available for each class and (ii) the available knowledge is not representative. We consider that this approach is more suitable to take into account the additional information embedded in complex data and, therefore, we further detail it in the rest of this section.

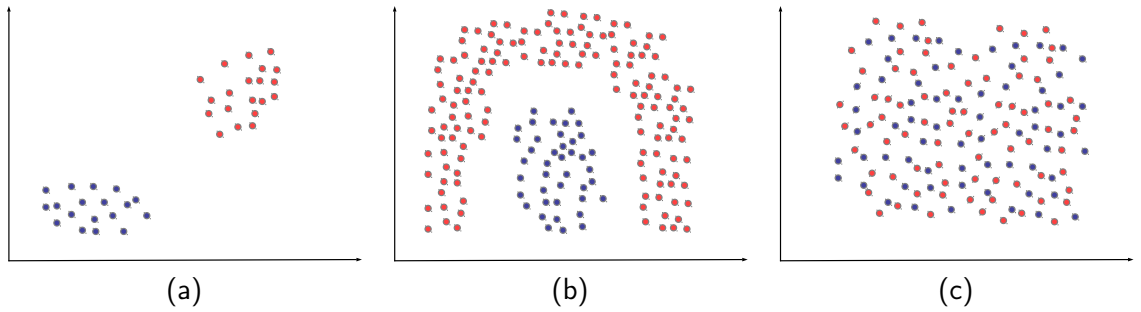


Figure 2.4 – Three clustering problems: (a) an easy problem, (b) a difficult problem and (c) an impossible problem.

Why guide the clustering? Traditional clustering algorithms are adapted to structure data from previously unknown domains (*e.g.*, the Yahoo! problem [Cohn *et al.* 2003]), but it fails to raise to the expectations when some background knowledge exists. For example, if the data naturally form tight clusters that are well-separated (as in Figure 2.4a), there is no need for background knowledge at all, any reasonable clustering algorithm will detect the desired clusters. Likewise, if no distinction can be made between classes in description space (as in Figure 2.4c), then little useful information can be found in the data itself, and supervision will again be of little use. Background knowledge will therefore be most useful when patterns are at least partially present in the data, but a clustering algorithm will not detect them correctly without assistance (as seen in Figure 2.4b). The idea is to use the available background knowledge to guide the clustering algorithm to find the "correct" partition.

How to model the supervision? The expert knowledge can be modeled either by using class labels (*i.e.*, like in supervised learning), or by using constraints. Constraints can be set, for example, on a subset of individuals or on the clusters (*e.g.*, when we want to form clusters respecting a certain condition apart the implicit cohesion). The constraints most used in the literature [Davidson & Basu 2007] are the constraints that model the relations between pairs of individuals, the pairwise constraints.

In [Wagstaff & Cardie 2000], two types of pairwise constraints are introduced. A "must-link" constraints between individuals x and y means that in the created partition, x and y must be placed into the same cluster. Similarly, a "cannot-link" constraint between x and y means that the two individuals cannot be placed into the same cluster. Must-link constraints are transitive (*i.e.*, $(x, y) \in \mathcal{M}$ and $(y, z) \in \mathcal{M} \Rightarrow (x, z) \in \mathcal{M}$, where $\mathcal{M}$ is the set of must-link constraints), which means that the pairwise constraints set can be enriched with new constraints by calculating the transitive closure of the must-link set.

In the next two paragraphs, we show that instance-level constraints are very versatile, being capable to model different types of information (*e.g.*, class labels, cluster conditions), as well as incomplete information.

Modeling different types of information using instance-level constraints Supervision under the form of *class labels* can be considered a special case of supervision using pairwise constraints. Given a subset of labeled individuals, the expert information can be translated into the pairwise constraints form by adding must-link constraints between all individuals sharing a label and cannot-link constraints between any two individuals labeled differently. Furthermore, supervision in the form of constraints is generally more practical than providing class labels in the clustering framework [Basu *et al.* 2003], since true labels may be unknown *a priori* and it is easier for a human expert to specify whether pairs of points belong to the same cluster or different clusters. For example, in [Cohn *et al.* 2003], human interaction is used to iterativelly ameliorate a document partition, by letting the user decide which pairs of documents are wrongly classified.

We have a similar situation for *cluster-level constraints*. For example, in [Davidson & Ravi 2005], two types of cluster level constraints are defined: (a) the ε-constraint enforces each point in a cluster to have a neighbor within a distance of at most ε (*i.e.*, constraint to prevent "rare" clusters, in which individuals are distanced); (b) the δ-constraint enforces that every individual in a cluster to be at a distance of at least δ from every individual in every other cluster (*i.e.*, constraint to enforce the separability of clusters). The two cluster-level constraints can be easily specified using instance-level constraints: the ε-constraint can be represented as a disjunction of must-link constraints (for every individual x, must-link to at least another individual y so that $||x - y|| \leq \varepsilon$) and the δ-constraint translates by a conjunction of must-link constraints (for every individual x, must-link to all individuals y, so that $||x - y|| < \delta$).

In Chapter 3, we use the pairwise constraints to model the *temporal information*, when trying to contiguously segment series of observations. We add must-link constraints between each pair of observations belonging to the same entity and inflict a penalty inversely proportional with their time difference when breaking the constraint.

Partial information vs. complete information Introducing external information into clustering is not a new domain. One of the oldest applications is clustering geographic-related data, using clustering with spatial contiguity constraints. There is a fundamental difference between such approaches and the semi-supervised approaches: the geographic information is available for all individuals, whereas in semi-supervised clustering the supervision is available only for a subset of individuals. Consequently, in clustering with contiguity constraints, a quick solution is to modify the dissimilarity measure to take into account the geographic information. For example, [Webster & Burrough 1972] adds a factor to the dissimilarity measure:

$$d^*_{ij} = d_{ij} \times \left(1 - e^{-\frac{h_{ij}}{w}}\right)$$

where d^*_{ij} is the modified measure between individuals i and j , d_{ij} is the original value of the measure, h_{ij} is the geographic distance between i and j and w is a weighting factor. This type of supervision is a special type of the semi-supervised case. The approaches presented in Section 2.2.1 also modify the similarity measure, but they use a supervised algorithm to leverage the partial information.

Taxonomy Traditional clustering algorithms employ a given *similarity measure* and a given *search strategy* in the solution space, in order to construct a partition of coherent clusters. The available supervised knowledge is leveraged by the semi-supervised clustering algorithms to modify either (or both) the *similarity measure* or the *search strategy*. Therefore, semi-supervised clustering methods can be divided [Basu *et al.* 2003, Grira *et al.* 2005] into two classes: (a) the similarity-based approaches, which seek to learn new similarity measures in order to satisfy the constraints, and (b) the search-based approached in which the clustering algorithm itself is modified. The following two sections (2.2.1 and 2.2.2) present in more detail each type of approach, together with some examples present in the literature.

2.2.1 Similarity-based approaches

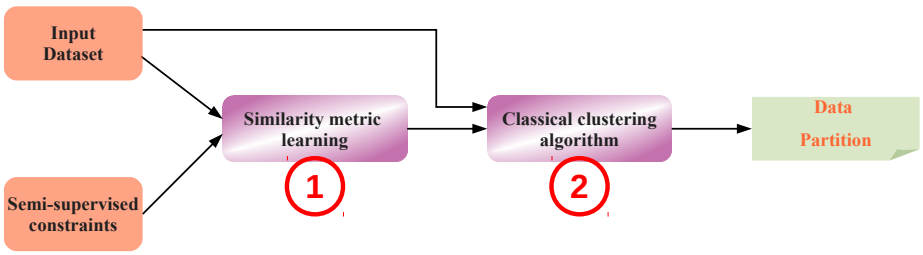


Figure 2.5 – General schema of similarity-based semi-supervised clustering approaches.

In similarity-based semi-supervised clustering approaches, an existing clustering algorithm that uses a similarity metric is employed. However, instead of using one of the existing predefined similarity measures [Lesot *et al.* 2009], the similarity metric used by these approaches is first trained to satisfy the labels or constraints in the supervised data. The general schema of this two phase process is given in Figure 2.5. In *phase 1*, the dataset is used together with the must-link and cannot-link constraints to train the similarity measure.

In *phase 2*, the trained similarity measure is used with the dataset in a classical clustering algorithm.

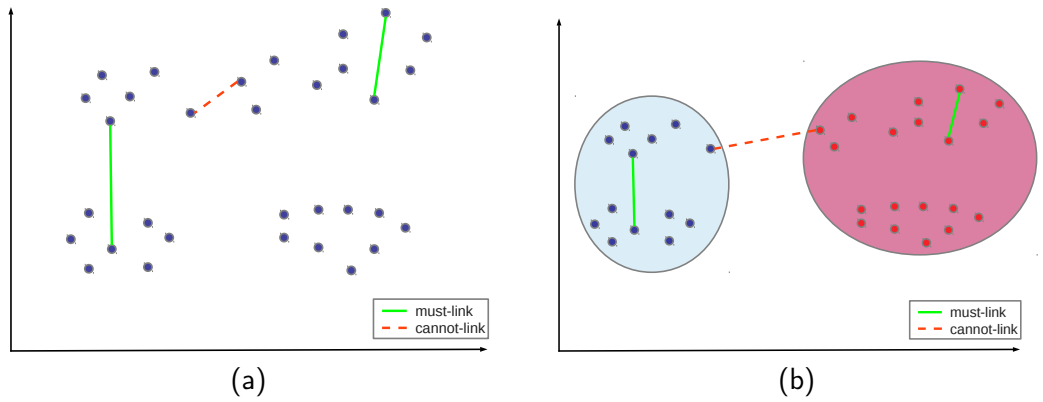


Figure 2.6 – The description space before (a) and after (b) training the similarity distance, in similarity-based approaches.

Training a similarity metric is similar to modifying the description space in order to enlarge the frontiers between groups of clusters. Individuals which are must-linked are pulled closer together, whereas individuals which are cannot-linked are distanced. Figure 2.6b shows a simple example of learning a distance function from the constraints given in Figure 2.6a. Notice that in Figure 2.6b, the input data space has been stretched in the horizontal dimension and compressed in the vertical dimension, to draw the must-linked individuals closer and pull the cannot-linked individuals farther apart.

Several distance measures have been used for distance-based constrained clustering:

- **Mahalanobis distance** trained with **convex optimization** [Xing *et al.* 2002, Bar-Hillel *et al.* 2003]. A parametrized Euclidean distance of the form $||x_1 - x_2||_A = \sqrt{(x_1 - x_2)^T A (x_1 - x_2)}$ is used in [Xing *et al.* 2002]. A two step optimization algorithm is used to fit the matrix A to the must-link and cannot-link constraints. New features are created that are linear combinations of the original features. The Relevant Component Analysis algorithm proposed in [Bar-Hillel *et al.* 2003] is similar, but uses a diagonal matrix instead, which boils down to simply assigning weight on dimensions. In both approaches, the matrix A is responsible for deforming the description space according to the constraints set.
- **Euclidean distance** trained with **shortest path algorithm** [Klein *et al.* 2002]. These methods start from the assumption that constraints suggest space-level generalizations beyond their explicit instance-level rules: not only should points that are must-linked be in the same cluster, but the points that are near these points (in the description space) should probably also be in the same cluster. A distance matrix between individuals is calculated, the distances of must-linked and cannot-linked individuals are modified and, in the end, the whole matrix is re-calculated by assigning as the distance between two individuals the length of the shortest path.
- **Kullback-Leibler divergence** [Kullback & Leibler 1951] trained with **gradient descent** [Cohn *et al.* 2003]. Interaction with the user is used in [Cohn *et al.* 2003] to acquire the semi-supervised constraints. A textual clustering is performed used a

naive Bayes algorithm and using the KL divergence to measure the similarity of two documents. When the user considers that two documents were wrongly grouped in the same cluster, he/she adds a cannot-link between the two. The constraints are taken into account by augmenting the KL divergence with a positive weighting function. The procedure is then reiterated until the user is happy with the result.

One of the strong points of similarity-based approaches is that the additional information is taken into account at the level of the similarity measure. Once the measure trained, any clustering algorithm can be used out of the box, without any modifications. The literature shows examples of several clustering algorithms, which can be used with trained distance measures, including single-link [Bilenko & Mooney 2003] and complete-link [Klein *et al.* 2002] agglomerative clustering, EM [Cohn *et al.* 2003, Bar-Hillel *et al.* 2003], and KMeans [Bar-Hillel *et al.* 2003, Xing *et al.* 2002].

Positioning our work This similarity-based semi-supervised clustering boils down to dividing the learning task into two: a smaller, simpler problem (*i.e.*, a similarity distance) is learned in a supervised fashion, using available supervision. The results of the supervised problem is later used in an unsupervised learning algorithm to improve the results of the bigger learning problem. We adopt a similar approach in Chapter 5, where we construct the "bag-of-features" visual vocabulary based on a small labeled image set. We employ a supervised algorithm to learn the visual words and an unsupervised algorithm to generate the actual numeric representation and the image clustering.

2.2.2 Search-based approaches

In search-based semi-supervised clustering approaches, the clustering algorithm itself is modified so that user-provided labels or constraints are used to bias the search for an appropriate partition. Literature shows several approaches for performing this bias:

- **enforcing constraints** [Wagstaff & Cardie 2000, Wagstaff *et al.* 2001]. Initial semi-supervised algorithms leveraged the additional information by enforcing the pairwise constraints during the clustering process. These are K-Means-like algorithms, which modify the cluster assignment step. Rather than performing a nearest centroid assignment, a nearest *feasible* centroid assignment is performed. This involves that constraints are never broken (which are called **hard pairwise constraints**). While such approaches have shown accuracy improvements, they are particularly sensible to noisy and contradicting constraints [Davidson & Basu 2007] (which are to be expected in real-life labeling). Another problem of such approaches is that they are unstable with regard to the order of presentation of the constraints (i.e., the results greatly vary if the order is changed) [Hong & Kwong 2009].
- **objective function modification** [Demiriz *et al.* 1999, Gao *et al.* 2006, Lin & Hauptmann 2006]. In most relocating clustering algorithm, the search in the solution space is guided by the employed objective function. Most semi-supervised clustering algorithms modify the objective function, in order to take supervision into account. In [Demiriz *et al.* 1999], an extra term is added to quantify the impurity of constructed clusters in terms of known class labels. The proposed clustering algorithm

minimizes the objective function

$$\mathcal{J} = \beta \times Cluster\_Dispersion + \alpha \times Cluster\_Impurity$$

where the parameters α and β control the impact of the supervision.
The problems shown by the previous class of algorithms (*i.e.*, algorithms enforcing constraints) can be solved by allowing constraints to be broken, in which case a penalty is inflicted. Such constraints are called **soft pairwise constraints**. In [Basu *et al.* 2003, Gao *et al.* 2006, Lin & Hauptmann 2006], the must-link and cannot-link pairwise constraints are also leveraged by modifying the objective function. The objective function includes penalization terms for each type of constraints and has the following formula:

$$\mathcal{J} = \sum_{x_i \in \mathcal{X}} ||x_i - \mu_{l_i}||^2 + \sum_{(x_i,x_j)\in\mathcal{M}} w_{ij}\mathbb{1}\left[l_i \neq l_j\right] + \sum_{(x_i,x_j)\in\mathcal{C}} \overline{w}_{ij}\mathbb{1}\left[l_i = l_j\right] \tag{2.1}$$

where:
- $\mathcal{X}$ is the given dataset;
- l_i is the cluster of individual x_i;
- μ_{l_i} is the centroid of cluster l_i;
- $\mathcal{M}$ is the set of must-link constraints;
- $\mathcal{C}$ is the set of cannot-link constraints;
- w_{ij} is the weight of the must-link constraint between x_i and x_j;
- $\overline{w}_{ij}$ is the weight of the cannot-link constraint between x_i and x_j;
- $\mathbb{1}\left[state\right]$ is a function that returns 1 if *state* if true and 0 otherwise.

Using such an objective function, the clustering algorithm converges towards a solution in which the partition breaks as few constraints as possible. In Chapter 3, we propose a temporal-aware constrained clustering algorithm which follows a similar approach.

– **seeding** [Basu *et al.* 2002]. The local optimum reached by a relocating clustering algorithm, such as K-Means, is influenced by the initial choice of centroids. Random initialization is usually used in "unsupervised" clustering, but in semi-supervised clustering, this can be done using the supervised information, in a two-phase process. In the first phase, sets of individuals that should belong to the same cluster are generated through transitive closure of the must-link constraints. In the second phase, centroids are initialized based on each of the computed sets and a classical clustering algorithm is used to further improve them in order to attain a partition with respect to the given constraints.

The different approaches presented earlier are not incompatible one with another. In [Basu *et al.* 2003], for example, a similarity-based approach using a trained Mahalanobis measure is combined with search-based approach, in which the objective function is modified to penalize breaking the constraints. The two are integrated into a K-Means-like approach, in which the centroids are initialized with a seeding strategy, as described here above.

Positioning our work In Chapter 3, we propose a temporal-aware constrained clustering algorithm, which uses a semi-supervised technique to ensure the contiguous segmentation of observations. We follow a search-based approach and we guide the clustering algorithm

by modifying the objective function. In order to ensure the contiguous segmentation of the temporal observations of an entity, we add soft pairwise must-link constraints between all observations belonging to the same entity. We penalize breaking these constraints using a penalization function which is dependent on the time difference between the constraints (the w_{ij} in Equation 2.1, while $\overline{w}_{ij}$ are set to zero).

Chapter 3

Detecting Typical Evolutions

Contents

3.1 Learning task and motivations

We consider that **leveraging the temporal dimension into the learning process** is a crucial point and it is one of the central research challenges of this thesis. The temporal dimension is more than just another descriptive dimension of data, since it profoundly changes the learning problem. The description of data becomes contextualized (*i.e.*, a certain description is true during a given time frame) and new learning problems arise: following the temporal evolution of individuals, detecting trends, topic burstiness, popular events tracking, *etc.* The temporal dimension is intimately related to the interactive Web 2.0, where the time is primordial for many learning tasks. In Section 2.1.3 (p. 16), we have discussed the difference between temporal algorithms and online ("on the fly") algorithms. Online algorithms have seen a lot of attention, especially given the huge amounts of data

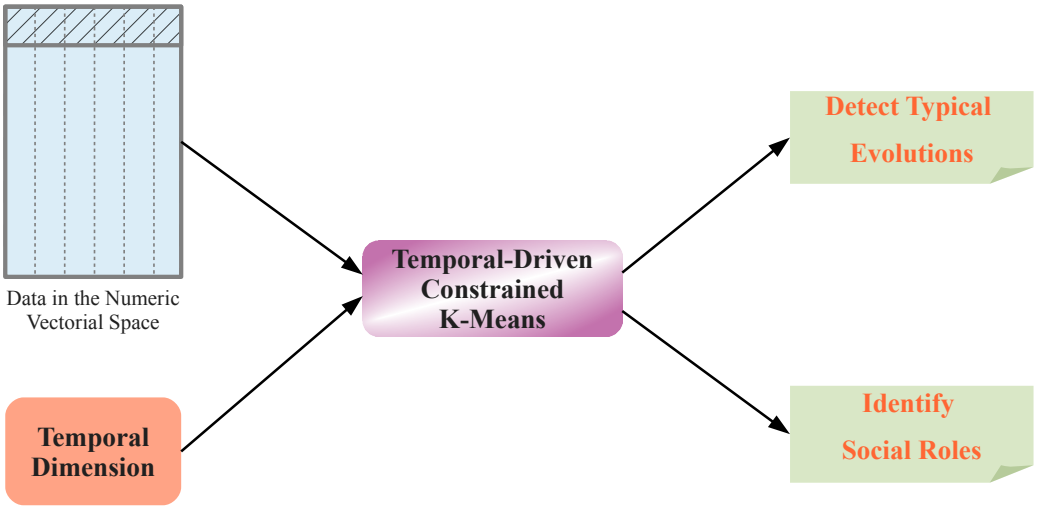


Figure 3.1 – Streamlined schema of how temporal information is used

produced on the Web 2.0. This data cannot be stored and the learning process must be done using single pass algorithms.

The work presented in this chapter tackles with one of our central research challenges, *i.e.*, dealing with the temporal dimension of complex data, by employing semi-supervised clustering techniques. The solutions and the algorithms presented in the following sections were developed as an answer to a specific learning task: *detecting typical evolution patterns*. This specific problem was originally motivated by the research interest of the Political Studies researchers involved in the IMAGIWEB project (see Annex A). We show in Section 3.6, that the application of our proposals is not limited to Political Studies datasets. We also employ our temporal clustering algorithm in a learning task issued from the domain of *Social Network Analysis*: detecting user social roles in a social network inferred based on online discussion forums. Our work concerning the temporal dimension does not concern the "on the fly" aspect. Our data are stored and we study them offline (*a posteriori*), in order to detect evolutions.

Motivation. Researchers in Political Studies have always gathered data and compiled databases of information. This information often has a temporal component: the evolution of a certain number of entities is recorded over a period of time. The idea is to inject the temporal component of the complex data into an automatic learning algorithm and **detect typical evolution patterns**. This is particularly interesting for Political Science researchers, since it would allow detecting hidden connections between the evolution of certain entities and the certain events that took place later in time (*e.g.* the rise of extremist political leaders and later wars). Such an algorithm would also assist the researchers in the process of creating entity typology: the classification of an entity at a certain moment in time is closely related to its previous evolution. This is not limited to Political Sciences, but can be generalized to many other fields of Social Sciences and Humanities. In Psychology, it is well known that the present mental state of a patient is very much influenced by traumatic events in his/her past.

We highlight in Figure 3.1 a schematic representation of the work performed in this chapter. The data is described in a numeric vectorial space and the main idea is to leverage the temporal dimension of the complex data together with its description. The temporal information is used at two levels. It is (a) embedded directly into the distance used to measure the similarity between instances and it is (b) injected into the clustering algorithm using semi-supervised techniques. In the case of the political science dataset, the results of the learning algorithm are typical evolutions. In Section 3.6, we exemplify the use of our proposal to another kind of data: *social network data*. This dataset describes the daily activity of users (the entities) on a web forum. The goal is to identify user roles in the underlying social network. Observations are the working block of our algorithm (*i.e.*, observations are descriptions of entities at a given timestamp). Therefore, our work deals with clustering observations, *i.e.*, tuples $(entity, timestamp, description)$, while taking into account the temporal component and contiguously segmenting the observations belonging to an entity.

The remainder of this chapter is organized as follows. In Section 3.2, we formalize the structure of the dataset and the objectives of our work. We also give an overview of the proposed solution. In Section 3.3 we present some existing previous work related to this specific problem and, in Section 3.4, we present our approach. We introduce the temporal-aware dissimilarity function, the contiguity penalty function and we combine them in the TDCK-Means algorithm. In Section 3.5, we present the dataset that we use, the proposed evaluation measures and the obtained results. In Section 3.6, we apply our algorithms on a *Social Network Analysis* task and we show how the TDCK-Means algorithm can be used to detect behavioral roles, which are in turn used to detect user social roles. Finally, in Section 3.7, we draw the conclusion and plan some future extensions.

3.2 Formalisation

We consider that the data are described in a numeric vectorial space (see Section 2.1, p. 9). It is not uncommon for Social Sciences and Humanities scientists to compile such datasets, which can be converted to a machine readable format with minimal intervention. We give just a few examples of such datasets that are publicly accessible:

- *Compared Political Dataset I* [Armingeon *et al.* 2011]: evolution of 23 democratic countries over a period of 50 years. [Available online] [1];
- *Democracy Time-series Data* [Norris 2008]: contains data on the social, economic and political characteristics of 191 countries with over 600 variables from 1971 to 2007. [Available online] [2];
- *Archigos* [Goemans *et al.* 2009]: is a data set with information on leaders in 188 countries from 1875 to 2004. [Available online] [3].

The dataset we analyze described a set $\phi_j \in \Phi$ of entities. Each entity is described for each considered moment $t_m \in \mathcal{T}$ of time using multiple attributes, which form the multidimensional description space $\mathcal{D}$. Therefore, an entry in such a database would be an observation, a triple $(entity, timestamp, description)$. An observation $x_i = (\phi_l, t_m, x_i^d)$ sig-

1. http://www.ipw.unibe.ch/content/team/klaus_armingeon/comparative_political_data_sets
2. http://www.nsd.uib.no/macrodataguide/set.html?id=56&sub=1
3. http://www.rochester.edu/college/faculty/hgoemans/data.htm

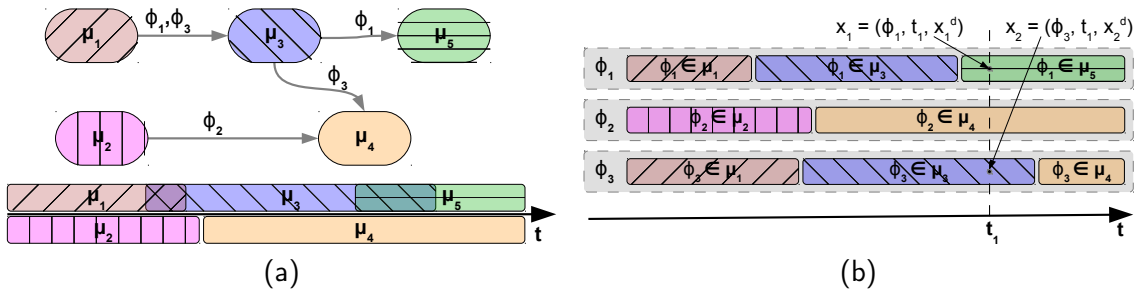


Figure 3.2 – Desired output: (a) the evolution phases and the entity trajectories, (b) the observations of 3 entities contiguously partitioned into 5 clusters.

nifies that the entity ϕ_l is described by the vector x_i^d at the moment of time t_m. We denote by x_i^ϕ the entity to which the observation x_i is associated. Similarly, x_i^t is the timestamp associated with the observation x_i. Each observation belongs to a single entity and, consequently, each entity is associated with multiple observations, for different moments of time. Formally:

$$\forall x_i \in \mathcal{D} : \exists! \, \phi_l \in \Phi \text{ so that } x_i^\phi = \phi_l$$
$$\forall (\phi_l, t_m) \in \Phi \times \mathcal{T} : \exists! \, x_i = (x_i^\phi, x_i^t, x_i^d) \text{ so that } x_i^\phi = \phi_l \text{ and } x_i^t = t_m$$

For example, the database that studies the evolution of democratic states [Armingeon *et al.* 2011] stores, for each country and each year, the value of multiple economical, social, political and financial indicators. The countries are the entities, and the years are the timestamps.

Starting from such a database, one of the interests of Political Studies researchers is to detect typical evolution patterns. There is a double interest: a) obtaining a broader understanding of the phases that the entity collection went through over time (*e.g.* detecting the periods of global political instability, of economic crisis, of wealthiness *etc.*); b) constructing the trajectory of an entity through the different phases (*e.g.* a country may have gone through a period of military dictatorship, followed by a period of wealthy democracy). The criteria describing each phase are not known beforehand (which indicators announce a world economic crisis?) and may differ from one phase to another.

We address these issues by proposing a novel temporal-driven constrained clustering algorithm. The proposed algorithm partitions the observations into clusters $\mu_j \in \mathcal{M}$, that are coherent both in the multidimensional description space and in the temporal space. We consider that the obtained clusters can be used to represent the typical phases of the evolution of the entities through time. Figure 3.2 shows the desired result of our clustering algorithm. Each of the three depicted entities (ϕ_1, ϕ_2 and ϕ_3) is described at 10 moments of time ($t_m, m = 1, 2, ..., 10$). The 30 observations of the dataset are partitioned into 5 clusters ($\mu_j, j = 1, 2, ..., 5$). In Figure 3.2a we observe how clusters μ_j are organized in time. Each of the clusters has a limited extent in time, and the time extents of clusters can overlap. The temporal extent of a cluster is the minimal interval of time that contains all the timestamps of the observations in that cluster. The entities navigate through clusters. When an observation belonging to an entity is assigned to cluster μ_2 and the anterior observation of the same

entity is assigned in cluster μ_1, then we consider that the entity has a transition from phase μ_1 to phase μ_2. Figure 3.2b shows how the series of observations belonging to each entity are assigned to clusters, thus forming continuous segments. This succession of segments is interpreted as the succession of phases through which the entity passes. For this succession to be meaningful, each entity should be assigned to a rather limited number of continuous segments. Passing through too many phases reduces the comprehension. Similarly, evolutions which are alternations between two phases (*e.g.*, $\mu_1 \longrightarrow \mu_2 \longrightarrow \mu_1 \longrightarrow \mu_2$) hinder the comprehension. Figure 3.3 shows an example of such an alternating evolution (the bad segmentation) and a more comprehensible evolution where there is only one alternation (the good segmentation).

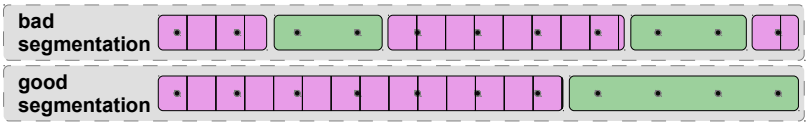


Figure 3.3 – Examples of a good and a bad segmentation into phases.

Based on these observations, we assume that the resulting partition must:

- **regroup observations having similar descriptions into the same cluster** (just as traditional clustering does). The clusters represent a certain type of evolution;
- **create temporally coherent clusters, with limited extent in time.** In order for a cluster to be meaningful, it should regroup observations which are temporally close (be contiguous on the temporal dimension). If there are two different periods with similar evolutions (*e.g.* two economical crises), it is preferable to have them regrouped separately, as they represent two distinct phases. Furthermore, while it is acceptable that some evolutions exist during the entire period, usually the resulted clusters should have a limited temporal extent;
- **segment, as contiguously as possible, the series of observations for each entity.** The sequence of segments will be interpreted as the sequence of phases through which the entity passes.

In order to construct such a partition, we propose a new time-aware dissimilarity measure that takes into account the temporal dimension. Observations that are close in the description space, but distant in time are considered as dissimilar. We also propose a method to enforce the segmentation contiguity, by introducing a penalty term based on the Normal Distribution Function. We combine the two propositions into a novel time-driven constrained clustering algorithm, **TDCK-Means**, which creates a partition of coherent clusters, both in the multidimensional space and in the temporal space. This algorithm uses soft semi-supervised constraints to encourage adjacent observations belonging to the same entity to be assigned to the same cluster. The proposed algorithm constructs the clusters that serve as evolution phases and segments the observations series for each entity. At the moment, our algorithm does not output the graph structure represented in Figure 3.2a. A promising venue for organizing the clusters as a graph, and not just as a post-processing based on the clustering results obtained using TDCK-Means, is going to be addressed in Section 3.7 and, more thoroughly, in Chapter 8.

3.3 Related work

The semi-supervised techniques seen in Section 2.2 are not limited to introducing expert knowledge into the clustering process. They can be used to inject any type of information external to the dataset. Previous works which model temporal information using semi-supervised tools already exist. The literature presents some examples of algorithms used to segment a series of observations into continuous chunks. In [Lin & Hauptmann 2006], the daily tasks of a user are detected by segmenting scenes from the recordings of his activities. Semi-supervised must-link constraints are set between all pairs of observations, and a fixed penalty is inflicted when the following conditions are fulfilled simultaneously: the observations are not assigned to the same cluster and the time difference between their timestamps is less than a certain threshold. A similar technique is used in [De la Torre & Agell 2007], where constraints are used to penalize non-smooth changes (over time) on the assigned clusters. This segmenting technique is used to detect tasks performed during a day, based on video, on sound and on GPS information. In [Sanders & Sukthankar 2001], the objects appearing in an image sequence are detected by using a hierarchical descending clustering, that regroups pixels into large temporally coherent clusters. This method seeks to maximize the cluster size, while guaranteeing intra-cluster temporal consistency. All of these techniques consider only one series of observations (a single entity) and must be adapted for the case of multiple series. The main problem of a threshold based penalty function is to set the value of the threshold, which is usually data-dependent. Optimal matching is used in [Widmer & Ritschard 2009] to discover trajectory models, while studying the de-standardization of typical life courses.

The temporal dimension of the data is also used in some other fields of Information Retrieval. In [Talukdar *et al.* 2012], constrained clustering is used to scope temporal relational facts in the knowledge bases, by exploiting temporal containment, alignment, succession, and mutual exclusion constraints among facts. In [Chen *et al.* 2009], clustering is used to segment temporal observations into continuous chunks, as a preprocessing phase. A graphical model is proposed in [Qamra *et al.* 2006], that uses a probabilistic model in which the timestamp is part of the observed variables, and the story is the hidden variable to be inferred. But still, none of these approaches seek to create temporally coherent partitions of the data, mainly using the temporal dimension as a secondary information.

In the following sections, we propose a dissimilarity measure, a penalty function and a clustering algorithm in which the temporal dimension has a central role, and which address the limitations existing in the above presented work.

3.4 Temporal-Driven Constrained Clustering

The observations $x_i \in \mathcal{X}$ that need to be structured can be written as triples $(entity, time, description)$: $x_i = (x_i^\phi, x_i^t, x_i^d)$. $x_i^d \in \mathcal{D}$ is the vector in the multidimensional description space which describes the entity $x_i^\phi \in \Phi$ at the moment of time $x_i^t \in \mathcal{T}$.

Traditional clustering algorithms input a set of multidimensional vectors, which they regroup in such a way that observations inside a group resemble each other as much as possible, and resemble observations in other groups as little as possible. K-Means [MacQueen 1967] is a clustering algorithm based on iterative relocation, that partitions a dataset into k clusters,

locally minimizing the sum of distances between each data points x_i and its assigned cluster centroids $\mu_j \in \mathcal{M}$. At each iteration, the objective function

$$I = \Sigma_{\mu_j \in \mathcal{M}} \Sigma_{x_i \in \mathcal{C}_j} ||x_i^d - \mu_j^d||^2$$

is minimized until it reaches a local optimum.

Such a system is appropriate for constructing partitions based solely on x_i^d, the description in the multidimensional space. It does not take into account the temporal order of the observations, nor the structure of the dataset, the fact that observations belong to entities. We extend to the temporal case by adding to the centroids a temporal dimension μ_j^t, described in the same temporal space $\mathcal{T}$ as the observations. Just like its multidimensional description vector μ_j^d, the temporal component does not necessary need to exist in the temporal set of the observation. It is an abstraction of the temporal information in the group, serving as a cluster timestamp. Therefore, a centroid μ_j will be the couple (μ_j^t, μ_j^d).

We propose to adapt the K-Means algorithm to the temporal case by adapting the Euclidean distance, normally used to measure the distance between an element and its centroid. This novel temporal-aware dissimilarity measure takes into account both the distance in the multidimensional space and in the temporal space. In order to ensure the temporal contiguity of observations for the entities, we add a penalty whenever two observations that belong to the same entity are assigned to different clusters. The penalty depends on the time difference between the two: the lower the difference, the higher the penalty. We integrate both into the **Temporal-Driven Constrained K-Means** (**TDCK-Means**), which is a temporal extension of K-Means. TDCK-Means searches to minimize the following objective function:

$$\mathcal{J} = \sum_{\mu_j \in \mathcal{M}} \sum_{x_i \in \mathcal{C}_j} \left(||x_i - \mu_j||_{TA} + \sum_{\substack{x_k \notin \mathcal{C}_j \\ x_k^\phi = x_i^\phi}} w(x_i, x_k) \right) \tag{3.1}$$

where $|| \bullet ||_{TA}$ is our temporal-aware (TA) dissimilarity measure (detailed in the next section), $w(x_i, x_j)$ is the cost function that determines the penalty of clustering adjacent observations of the same entity into different clusters, and $\mathcal{C}_j$ is the set of observations in cluster j.

3.4.1 The temporal-aware dissimilarity measure

The proposed temporal-aware dissimilarity measure $||x_i - x_j||_{TA}$ combines the Euclidean distance in the multidimensional space $\mathcal{D}$ and the distance between the timestamps. We propose to use the following formula:

$$||x_i - x_j||_{TA} = 1 - \left(1 - \frac{||x_i^d - x_j^d||^2}{\Delta x_{max}^2}\right) \left(1 - \frac{||x_i^t - x_j^t||^2}{\Delta t_{max}^2}\right) \tag{3.2}$$

where $|| \bullet ||$ is the classical L^2 norm and Δx_{max} and Δt_{max} are the diameters of $\mathcal{D}$, and $\mathcal{T}$ respectively (the largest distance encountered between two observations in the multidimensional description space and, respectively, in the temporal space). The following properties are immediate:

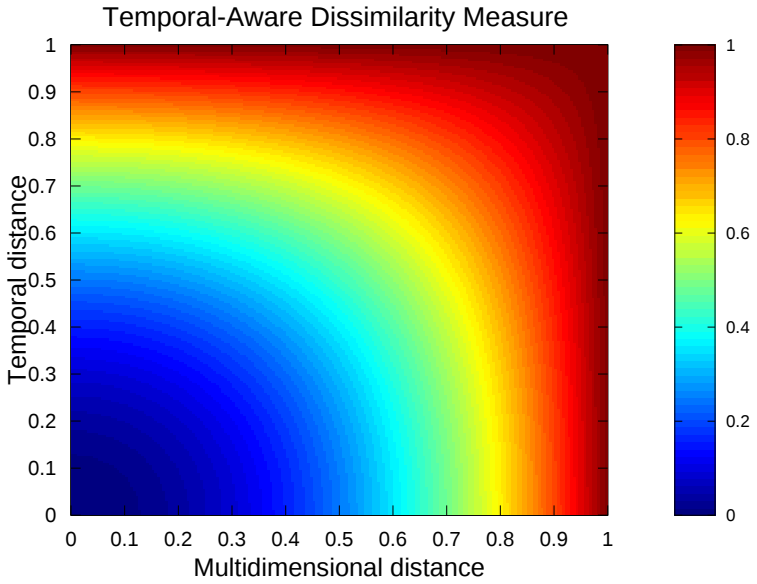


Figure 3.4 – Color map of the temporal-aware dissimilarity measure as a function of the multidimensional component and the temporal component.

- $||x_i - x_j||_{TA} \in [0,1], \forall x_i, x_j \in \mathcal{X}$
- $||x_i - x_j||_{TA} = 0 \Leftrightarrow x_i^d = x_j^d$ and $x_i^t = x_j^t$
- $||x_i - x_j||_{TA} = 1(maximum) \Leftrightarrow ||x_i^d - x_j^d|| = \Delta x_{max}$ or $||x_i^t - x_j^t|| = \Delta t_{max}$

Figure 3.4 plots the temporal-aware dissimilarity measure as a color map, depending on the multidimensional component and the temporal component. The horizontal axis represents the normalized multidimensional distance ($\frac{||x_i^d - x_j^d||^2}{\Delta x_{max}^2}$). The vertical axis represents the normalized temporal distance ($\frac{||x_i^t - x_j^t||^2}{\Delta t_{max}^2}$). The blue color shows a temporal-aware measure close to the minimum and the red color represents the maximum. The dissimilarity measure is zero if and only if the two observations have equal timestamps and equal multidimensional description vectors. Still, it suffices for only one of the components (temporal, multidimensional) to attend the maximum value for the measure to reach its maximum. The measure behaves similar to a MAX operator, always choosing a value closer to the maximum of the two components. The formula for the temporal-aware dissimilarity measure was chosen so that any algorithm that seeks to minimize an objective function based on the this measure, will need to minimize both its components. This makes it suitable for algorithms that search to minimize both the multidimensional and the temporal variance in clusters.

Both components that intervene in the measure follow a function like $1 - \varepsilon^2, \varepsilon \in [0,1]$. This function provides a good compromise: it is tolerant for small values of ε (small time difference, small multidimensional distance), but decreases rapidly when ε augments. The temporal-aware dissimilarity measure is an extension of the Euclidean function. If the timestamps are unknown and set to be all equal, the temporal component is canceled and the temporal-aware dissimilarity measure becomes a normalized Euclidean distance. In Section 3.5.4, we evaluate the behavior of the proposed dissimilarity function. We will call **Temporal-Driven K-Means** the algorithm that is based on the K-Means' iterative structure and uses the temporal-aware dissimilarity measure to asses similarity between ob-

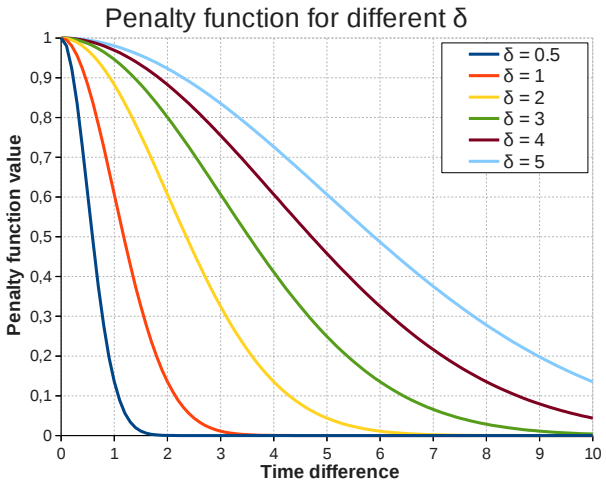


Figure 3.5 – Penalty function vs. time difference for multiple δ. ($\beta = 1$)

servations. Notice that **Temporal-Driven K-Means**, relative to TDCK-Means, has no contiguous segmentation penalty function (the contiguous segmentation penalty function is detailed in the next section).

3.4.2 The contiguity penalty function

The penalty function encourages temporally adjacent observations of the same entity to be assigned to the same cluster. We use the notion of *soft pair-wise constraints* from semi-supervised clustering. A "must-link" soft constraint is added between all pairs of observations belonging to the same entity. The clustering is allowed to break the constraints, while inflicting a penalty for each of these violations. The penalty is more severe if the observations are closer in time. The function is defined as:

$$w(x_i, x_j) = \beta * e^{-\frac{1}{2}\left(\frac{||x_i^t - x_j^t||}{\delta}\right)^2} \mathbb{1}\left[x_i^\phi = x_j^\phi\right] \tag{3.3}$$

where β is a scaling factor and, at the same time, the maximum value taken by the penalty function; δ is a parameter which controls the width of the function. β is dataset dependent and can be set as a percentage of the average distance between observations. $\mathbb{1}[statement]$ is a function that returns 1 if *statement* is true and 0 otherwise.

The function resembles to the positive side of the Normal Distribution function, centered in zero. The function has a particular shape, as represented in Figure 3.5. For small time differences, it descends very slowly, thus inflicting a high penalty for breaking a constraint. As the time difference increases, the penalty decreases rapidly, converging towards zero. When δ is small, the functions value descends very quickly with the time difference. The function produces penalties only if the constraint is broken for adjacent observation. For high values of δ, breaking constraints for distant observations cause high penalties, therefore creating segmentations with large segments. Figure 3.5 shows the evolution of the penalty function with the time difference between two observations, for multiple values of δ and for $\beta = 1$.

An advantage of the proposed function is that it requires no time discretization or setting a fixed window width, as proposed in [Lin & Hauptmann 2006]. The δ parameter permits the fine tuning of the penalty function. In Section 3.5.4, we evaluate **Constrained K-Means**, which is an extension of K-Means, to which we add the proposed contiguity penalty function (but which does not take into account the temporal dimension when measuring the distance between observations). The influence of both β and δ will be studied in Section 3.5.5.

3.4.3 The TDCK-Means algorithm

The time dependent distance $||x_i - \mu_j||_{TA}$ encourages the decrease of both the temporal and multidimensional variance of clusters; meanwhile the penalty function $w(x_i, x_j)$ favors the adjacent observations belonging to the same entity to be assigned to the same cluster. The rest of the TDCK-Means algorithm is similar to the K-Means algorithm. It seeks to minimize $\mathcal{J}$ by iterating an assignment phase and a centroid update phase until the partition does not change between two iterations. The outline of the algorithm is given in Algorithm 1.

The **choose_random** function chooses randomly, for each centroid μ_j, an observation x_i and sets $\mu_j = (x_i^t, x_i^d)$. Furthermore, in the initialization phase we perform a K-Means iteration in order to calculate an initial affectation of observations to clusters. For each individual the **best_initial_cluster** function solves the following equation:

$$\textbf{best\_initial\_cluster}(i) = \underset{j=1,2,\ldots,k}{argmin} \left(||x_i - \mu_j^{(iter-1)}||_{TA}^2 \right) \tag{3.4}$$

therefore affecting each observation to the closest centroid, in terms of temporal-aware dissimilarity measure. In the assignment phase, for every observation x_i, the **best_cluster** function chooses a cluster $\mathcal{C}_j$ so that the temporal-aware dissimilarity measure from x_i to the cluster's centroid μ_j, added to the cost of penalties possibly incurred by this cluster assignment, is minimized. This function is similar to the **best_initial_cluster** function in Equation 3.4, to which the penalty term is added. It resumes to solving the following equation:

$$\textbf{best\_cluster}(i) = \underset{j=1,2,\ldots,k}{argmin} \left(||x_i - \mu_j^{(iter-1)}||_{TA}^2 + \sum_{\substack{x_k^{\phi}=x_i^{\phi} \\ x_k \notin \mathcal{C}_j^{(iter-1)}}} w(x_i, x_k) \right) \tag{3.5}$$

This guaranties that the contribution of x_i to the value of $\mathcal{J}$ diminishes or stays constant. Overall, this assures that $\mathcal{J}$ diminishes in the assignment phase (or stays constant). It is noteworthy that, in Equation 3.5, the affectation of an observation in the current iteration is dependent on the affectation of the other observations in the previous iteration. This is also the reason why an extra K-Means iteration was necessary in the initialization of the algorithm and why the **best_initial_cluster** function was defined.

In the centroid update phase, the **update_centroid** function recalculates the cluster centroids using the observations in $\mathcal{X}$ and the assignment at the previous iteration. Therefore the contribution of each cluster to the $\mathcal{J}$ function is minimized. Each of the temporal and the multidimensional components is calculated individually. In order to find the values that minimize the objective function, we need to solve the equations:

$$\frac{\partial \mathcal{J}}{\partial \mu_j^d} = 0 \; ; \quad \frac{\partial \mathcal{J}}{\partial \mu_j^t} = 0 \tag{3.6}$$

Algorithm 1 Outline of the TDCK-Means algorithm.

Input: $x_i \in \mathcal{X}$ - observations to cluster;
Input: k - number of requested clusters;
Output: $\mathcal{C}_j, j = 1, 2, ..., k$ - k clusters;
Output: $\mu_j, j = 1, 2, ..., k$ - centroids for each cluster;

```
for j = 1, 2, .., k do
    μ_j ← choose_random(X)
    C_j^0 ← ∅
for x_i ∈ X do
    C_j^0 = C_j^0 ∪ x_i| where j = best_initial_cluster(X, M^0)
iter ← 0
M^(iter) ← ∅      //set of centroids
P^(iter) ← ∅      //set of clusters
repeat
    iter ← iter + 1
    for j = 1, 2, ..., k do
        C_j^(iter) ← ∅
    // assignment phase
    for x_i ∈ X do
        C_j^(iter) = C_j^(iter) ∪ x_i| where j = best_cluster(X, M^(iter−1), P^(iter−1))
    // centroids update phase
    for j = 1, 2, ..., k do
        (μ_j^(φ,(iter)), μ_j^(t,(iter))) ← update_centroid(j, X, M^(iter−1), P^(iter−1))
    M^(iter) ← {μ_j^(iter)|j = 1, 2, ..., k}
    P^(iter) ← {C_j^(iter)|j = 1, 2, ..., k}
until C_j^(iter) = C_j^(iter−1), ∀j ∈ [1, k]
```

By replacing equations (3.2) and (3.3) in (3.1), we obtain the following formula for the objective function:

$$\mathcal{J} = |\mathcal{X}| - \sum_{j=1}^{k} \sum_{x_i \in \mathcal{C}_j} \left[\left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2} \right) \left(1 - \frac{||x_i^t - \mu_j^t||^2}{\Delta t_{max}^2} \right) \right] + \sum_{x_i \in \mathcal{X}} \sum_{x_k \notin \mathcal{C}_j} \beta * e^{-\frac{1}{2}\left(\frac{||x_i^t - x_j^t||}{\delta}\right)^2} \mathbb{1}\left[x_i^\phi = x_j^\phi \right] \tag{3.7}$$

We exemplify the calculation for the update formula of μ_j^t.

$$\frac{\partial \mathcal{J}}{\partial \mu_j^t} = 0 \Leftrightarrow \frac{\partial}{\partial \mu_j^t} \left(\sum_{l=1}^{k} \sum_{x_i \in \mathcal{C}_l} \left[\left(1 - \frac{||x_i^d - \mu_l^d||^2}{\Delta x_{max}^2} \right) \left(1 - \frac{(x_i^t - \mu_l^t)^2}{\Delta t_{max}^2} \right) \right] \right) = 0$$

$$\Leftrightarrow \frac{2}{\Delta t_{max}^2} \times \sum_{x_i \in \mathcal{C}_j} \left[\left(x_i^t - \mu_j^t \right) \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2} \right) \right] = 0$$

$$\Leftrightarrow \sum_{x_i \in \mathcal{C}_j} \left[\left(x_i^t - \mu_j^t\right) \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2}\right)\right] = 0$$

$$\Leftrightarrow \sum_{x_i \in \mathcal{C}_j} x_i^t \times \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2}\right) = \sum_{x_i \in \mathcal{C}_j} \mu_j^t \times \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2}\right)$$

$$\Leftrightarrow \mu_j^t = \frac{\sum_{x_i \in \mathcal{C}_j} x_i^t \times \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2}\right)}{\sum_{x_i \in \mathcal{C}_j} \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2}\right)}$$

Therefore, from equations (3.6) and (3.7), we obtain the centroid update formulas:

$$\begin{aligned} \mu_j^d &= \frac{\sum_{x_i \in \mathcal{C}_j} x_i^d \times \left(1 - \frac{||x_i^t - \mu_j^t||^2}{\Delta t_{max}^2}\right)}{\sum_{x_i \in \mathcal{C}_j} \left(1 - \frac{||x_i^t - \mu_j^t||^2}{\Delta t_{max}^2}\right)} \\ \mu_j^t &= \frac{\sum_{x_i \in \mathcal{C}_j} x_i^t \times \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2}\right)}{\sum_{x_i \in \mathcal{C}_j} \left(1 - \frac{||x_i^d - \mu_j^d||^2}{\Delta x_{max}^2}\right)} \end{aligned} \tag{3.8}$$

Just like the centroid update phase in K-Means, the new centroids in TDCK-Means are also averages over the observations. Unlike K-Means, the averages are weighted for each component, using the distance from the other. For example, each observation contributes to the multidimensional description of the new centroid, proportional with its temporal centrality in the cluster. Observations that are more distant in time (from the centroid) contribute less to the multidimensional description than the ones being closer in time. A similar logic applies to the temporal component. The consequence is that the new clusters are coherent both in the multidimensional space and in the temporal one.

Algorithm's complexity Equation (3.7) shows that TDCK-Means' complexity is $\mathcal{O}(n^2k)$, due to the penalty term. Still, the equation can be rewritten, so that only observations belonging to the same entity are tested. If p is the number of entities and q is the maximum number of observations associated with each entity, then $n = p \times q$. The complexity of TDCK-Means is $\mathcal{O}(pq^2k)$, which is well adapted to Social Science and Humanities datasets, where often a large number of individuals is studied over a relatively short period of time ($p > q$).

3.4.4 Fine-tuning the ratio between components

The temporal-aware dissimilarity measure, as presented in equation (3.2), gives equal importance to both the multidimensional component and the temporal component. This might pose problems when the data are not uniformly distributed both in the multidimensional descriptive space and in the temporal space. If the medium standard deviation reported to the medium distance between pairs of observations is greater in one space than in the other, giving equal weight to the components can lead to important bias in the clustering process.

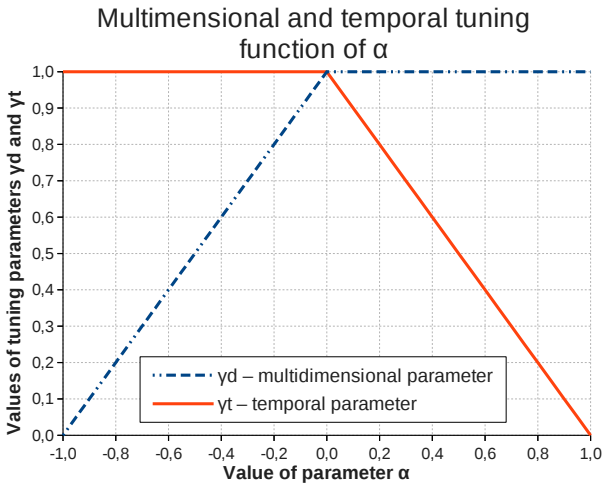


Figure 3.6 – Multidimensional component, temporal component and temporal-aware dissimilarity measure function of α

E.g. observations that are very uniformly distributed in the temporal space (same number of observations for each timestamp) and, at the same time, rather compactly distributed in the description space. In this case, in average, the temporal component weight more in the dissimilarity measure than the multidimensional component. Consequently, the clustering is biased towards the temporal cohesion of clusters. Similarly, in some applications, it is desirable to privilege one component over the other. *E.g.* on a large enough scale, user roles in social networks have a temporal component (new types of roles might appear over the years). But in a limited time span, it is perfectly acceptable that the roles can coexist simultaneously. Therefore, the temporal component should have only a mild impact on the overall measure.

We adjust the ratio between the two components by using two tuning factors γ_d and γ_t. γ_d weights the multidimensional component of the temporal-aware dissimilarity measure, whereas γ_t weights the temporal component. Equation (3.2) can be rewritten as:

$$||x_i - x_j||_{TA} = 1 - \left(1 - \gamma_d \frac{||x_i^d - x_j^d||^2}{\Delta x_{max}^2}\right)\left(1 - \gamma_t \frac{||x_i^t - x_j^t||^2}{\Delta t_{max}^2}\right) \tag{3.9}$$

When the tuning factor for a certain component is set at zero, the respective component does not contribute to the temporal-aware measure. When the tuning factor is set to one, no penalty is inflicted to the contribution of the respective component to the measure. It is immediate that equation (3.2) is a special case of equation (3.9), with $\gamma_d = 1$ and $\gamma_t = 1$ (no weights).

Setting the weights γ_d and γ_t γ_d and γ_t are not independent one from another, their values are set using a unique parameter α.

$$\gamma_d = \begin{cases} 1+\alpha, & \text{if } \alpha \leq 0 \\ 1, & \text{if } \alpha > 0 \end{cases} \quad ; \quad \gamma_t = \begin{cases} 1, & \text{if } \alpha \leq 0 \\ 1-\alpha, & \text{if } \alpha > 0 \end{cases} \tag{3.10}$$

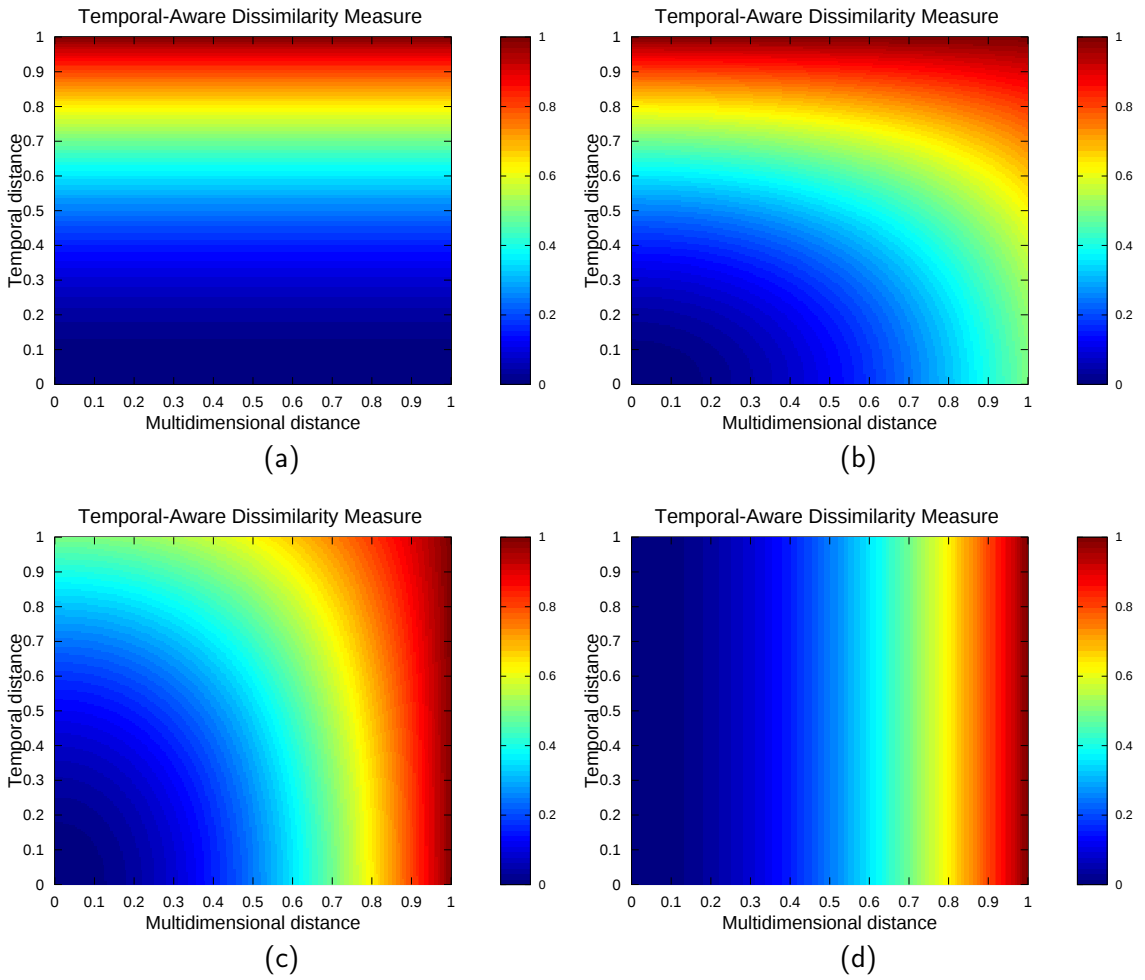


Figure 3.7 – Color map of the temporal-aware dissimilarity measure for $\alpha = -1$ (a), $\alpha = -0.5$ (b), $\alpha = 0.5$ (c) and $\alpha = 1$ (d) .

α acts as a slider, taking values from -1 to 1. Figure 3.6 shows the evolution γ_d and γ_t with α. Also, Figure 3.7 shows the color map of the temporal-aware dissimilarity measure for multiple values of α.

When $\alpha = -1$, then $\gamma_d = 0$ and $\gamma_t = 1$. The multidimensional component is eliminated and only the time difference between the two observations is considered. The temporal-aware measure becomes a normalized time difference ($||x_i - x_j||_{TA} = \frac{||x_i^t - x_j^t||^2}{\Delta t_{max}^2}$). The color map in Figure 3.7a ($\alpha = -1$) shows that the values of the dissimilarity measure is independent of the multidimensional component.

As the value of α increases, the weight of the descriptive component increases also. In Figure 3.7b ($\alpha = -0.5$), the multidimensional component has a limited impact on the overall measure. When $\alpha = 0$, then $\gamma_d = 1$ and $\gamma_t = 1$, both components have equal importance, as proposed initially in equation (3.2). In Figure 3.7c ($\alpha = 0.5$), the color map shows that the multidimensional component has a larger impact then the temporal component. Large values of the temporal component have only moderate influence over the measure. When $\alpha = 1$

(color map in Figure 3.7d), then $\gamma_d = 1$ and $\gamma_t = 0$, the temporal dimension is eliminated and the measure becomes a normalized Euclidean distance ($||x_i - x_j||_{TA} = \frac{||x_i^d - x_j^d||^2}{\Delta x^2_{max}}$).

Since the temporal-aware dissimilarity measure is used into the objective function in equation (3.7), the latter changes accordingly to integrate the tuning factors. γ_d and γ_t behave as constants in the derivation formulas in equation (3.6). As a result, the centroid update formulas in equation (3.8) are rewritten as:

$$\mu_j^d = \frac{\sum_{x_i \in \mathcal{C}_j} x_i^d \times \left(1 - \gamma_t \frac{||x_i^t - \mu_j^t||^2}{\Delta t^2_{max}}\right)}{\sum_{x_i \in \mathcal{C}_j} \left(1 - \gamma_t \frac{||x_i^t - \mu_j^t||^2}{\Delta t^2_{max}}\right)} \quad ; \quad \mu_j^t = \frac{\sum_{x_i \in \mathcal{C}_j} x_i^t \times \left(1 - \gamma_d \frac{||x_i^d - \mu_j^d||^2}{\Delta x^2_{max}}\right)}{\sum_{x_i \in \mathcal{C}_j} \left(1 - \gamma_d \frac{||x_i^d - \mu_j^d||^2}{\Delta x^2_{max}}\right)}$$

The tuning between the multidimensional and temporal component in the temporal-aware dissimilarity measure propagates into the centroid update formula of TDCK-Means. We study, in Section 3.5.6, the influence of the tuning parameter and we propose an heuristic to set its value.

3.5 Experiments

3.5.1 Dataset

Experimentations with Time-Driven Constrained K-Means are performed on a dataset issued from political sciences: *Comparative Political Data Set I* [Armingeon *et al.* 2011]. It is a collection of political and institutional data, which consists of annual data for 23 democratic countries for the period from 1960 to 2009. The dataset contains 207 political, demographic, social and economic variables.

The dataset was cleaned by removing redundant variables (*e.g.* country identifier and postal code) and the corpus was preprocessed by removing entity bias from the data. For example, it is difficult to compare, on the raw data, the evolution of population between populous country and one with fewer inhabitants, since any evolution in the 50 years timespan of the dataset will be rendered meaningless by the initial difference. Inspired from panel data econometrics [Dormont 1989], we remove the entity-specific, time-invariant effects, since we assume them to be fixed over time. We subtract from each value the average over each attribute and over each entity. We retain the time-variant component, which is in turn normalized, in order to avoid giving too much importance to certain variables. The obtained dataset is under the form of triples $(country, year, description)$.

3.5.2 Qualitative evaluation

When studying the evolution of countries over the years, it is quite obvious for the human reader why the evolutions of the eastern European countries resemble each other for most of the second half of the twentieth century. The reader would create a group entitled "Communism", extending from right after the Second World War until roughly 1990, for defining the typical evolution of communist countries. One would expect that, based on a political dataset, the algorithms would succeed in identifying such typical evolutions and segment the time series of each of these countries accordingly. Figure 3.8 shows the typical

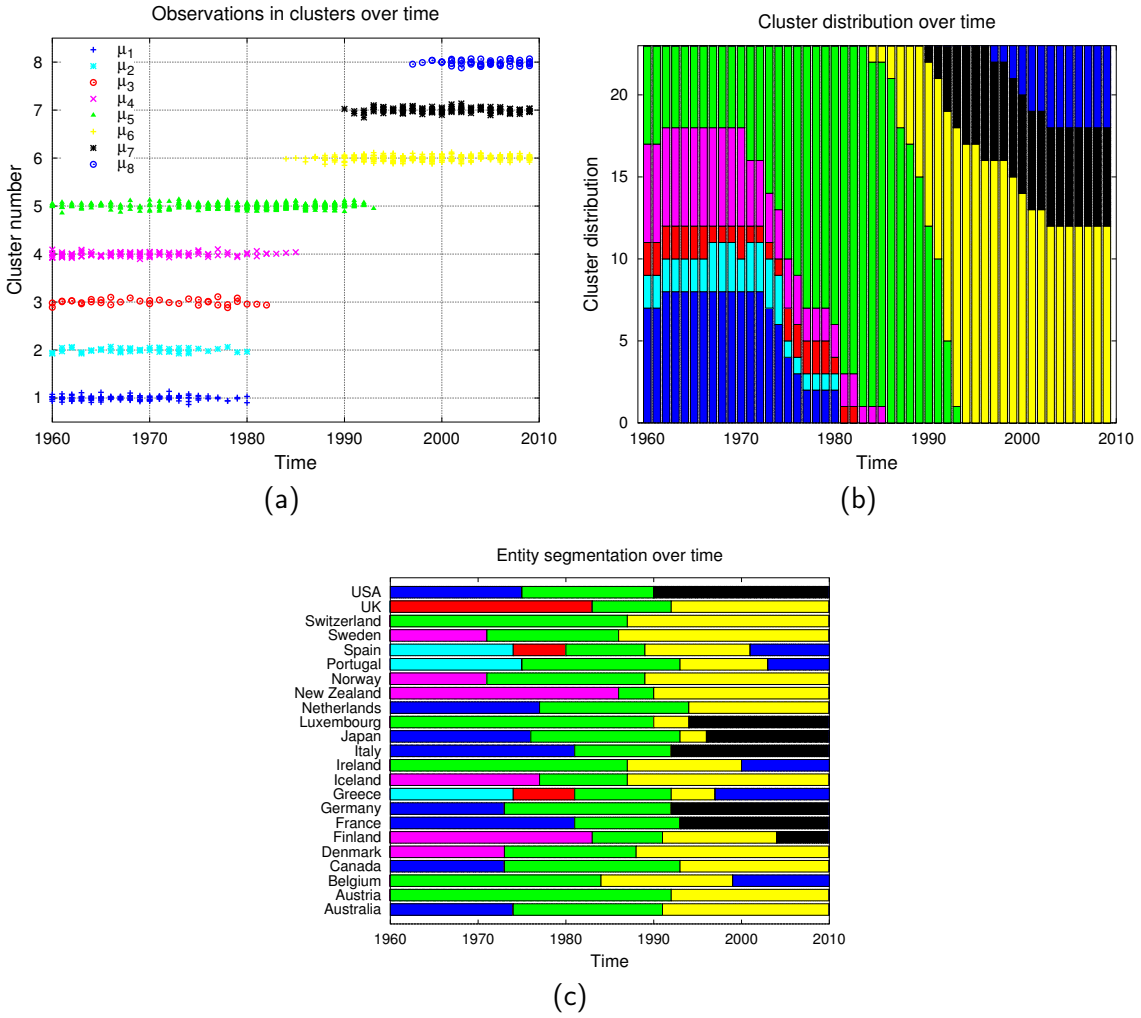


Figure 3.8 – Typical evolution patterns constructed by TDCK-Means on *Comparative Political Data Set I* with 8 clusters. The distribution over time of observations in each cluster (a), how many entities belong in a certain clusters for each year (b) and the segmentation of entities over clusters (c).

evolution patterns constructed by TDCK-Means (with $\beta = 0.003$ and $\delta = 3$, obtained as shows in Section 3.5.5), when asked for 8 clusters. The distribution over time of observations in each cluster is given in Figure 3.8a. All constructed clusters are fairly compact in time and have limited temporal extents. They can be divided into two temporal groups. In the first one, clusters μ_1 to μ_5 consistently overlap. Same for clusters μ_6 to μ_8, in the second group. This indicates that the evolution of each country passes by at least one cluster from each group. The turning point between the two groups is around 1990. Figure 3.8b shows how many countries belong in a certain cluster for each year. Clusters μ_5 and μ_6 contain most of the observations, suggesting the general typical evolution.

The meaning of each constructed cluster starts to unravel only when studying the segmentation of countries over clusters, in Figure 3.8c. For example, cluster μ_2 regroups the observations belonging to Spain, Portugal and Greece from 1960 up until around 1975.

Historically, this coincides with the non-democratic regimes in those countries (Franco's dictatorship in Spain, the "Regime of the Colonels" in Greece). Likewise, cluster μ_4 contains observations of countries like Denmark, Finland, Iceland, Norway, Sweden and New Zealand. This cluster can be interpreted as the "Swedish Social and Economical Model" of the Nordic countries, to which the algorithm added, interestingly enough, New Zealand. In the second period, cluster μ_8 regroups observations of Greece, Ireland, Spain, Portugal and Belgium, the countries which seemed the most fragile in the aftermaths of the economical crises of 2008.

3.5.3 Evaluation measures

Since the dataset contains no labels to report to as ground truth, we use the classical Information Theory measures in order to numerically evaluate the proposed algorithms. We evaluate separately each of the three goals that we propose in Section 3.2.

Create clusters that are coherent in the multidimensional description space. It is desirable that observations that have similar multidimensional descriptions to be partitioned under the same cluster. The similarity in the description space is measured by the multidimensional component of the temporal-aware dissimilarity measure. This goal is pursued by all classical clustering algorithms (like K-Means) and any traditional clustering evaluation measure [Halkidi *et al.* 2001] can be used to asses it. We choose the mean cluster variance, which is traditionally used in clustering to quantify the dispersion of observations in clusters. The *MDvar* measure is defined as:

$$MDvar = \frac{1}{|\mathcal{X}|} \times \sum_{j=1}^{k} \sum_{x_i \in \mathcal{C}_j} ||x_i^d - \mu_j^d||^2$$

Create temporally coherent clusters, with limited extend in time. This goal is very similar to the previous one, translated in the temporal space. It is desirable that observations that are assigned to the same cluster to be similar in the temporal space (*i.e.* to be close in time). The similarity in the temporal space is measured by the temporal component in the temporal-aware dissimilarity measure. The limited time extent of a centroid implies small temporal distances between observations timestamp and the centroid timestamp. As a result, the variance can also be used to measure the dispersion of clusters in the temporal space. Similarly to *MDvar*, the *Tvar* measure is defined as:

$$Tvar = \frac{1}{|\mathcal{X}|} \times \sum_{j=1}^{k} \sum_{x_i \in \mathcal{C}_j} ||x_i^t - \mu_j^t||^2$$

Segment the temporal series of observations of each entity into a relatively small number of contiguous segments. The goal is to have successive observations belonging to an entity grouped together, rather that scattered in different clusters. The Shannon entropy can quantify the number of clusters which regroup the observations of an entity, but it is insensible to alternations between two classes (evolutions like $\mu_1 \longrightarrow \mu_2 \longrightarrow \mu_1 \longrightarrow \mu_2$).

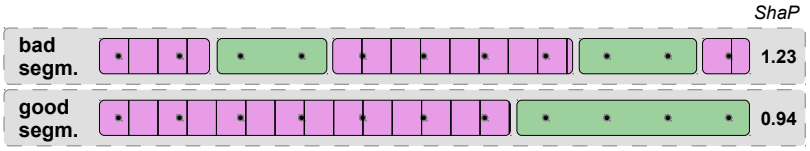


Figure 3.9 – Examples of a good and a bad segmentation in contiguous chunks and their related *ShaP* score.

We evaluate using an adapted mean Shannon entropy of clusters over entities, which weights the entropy by a penalty factor depending on the number of continuous segments in the series of each entity. The *ShaP* measure is calculated as:

$$ShaP = \frac{1}{|\mathcal{X}|} \times \sum_{x_i \in \mathcal{X}} \sum_{j=1}^{k} \left(-p(\mu_j) \times \log_2(p(\mu_j)) \times \left(1 + \frac{n_{ch} - n_{min}}{n_{obs} - 1} \right) \right)$$

where n_{ch} is the number of changes in the cluster assignment series of an entity, n_{min} is the minimal required number of changes and n_{obs} is the number of observations for an entity. For example, in Figure 3.9, if the series of 11 observations of an entity is assigned to two clusters, but it presents 4 changes, the entropy penalty factor will be $1 + \frac{4-1}{11-1} = 1.33$. The *ShaP* score for this segmentation will be 1.23, compared to a score of 0.94 of the "ideal" segmentation (only two contiguous chunks).

The "ideal" values for *MDvar*, *Tvar* and *ShaP* is zero and, in all of the experiments presented in the following sections, we search to minimize the values of the three measures.

3.5.4 Quantitative evaluation

For each combination of algorithms and parameters, we execute 10 times and compute only the average and the standard deviation. We vary k, the number of clusters, from 2 to 36. The performances of five algorithms are compared from a quantitative point of view:

- **Simple K-Means** - clusters the observations based solely on their resemblance in the multidimensional space;
- **Temporal-Driven K-Means** - optimizes only the temporal and multidimensional components, without any contiguity constraints; combines K-Means with the temporal-aware dissimilarity measure defined in Section 3.4.1. Parameters: $\alpha = 0$ (α defined in Equation 3.8) and $\beta = 0$ (β defined in Equation 3.3);
- **Constrained K-Means** - uses only the multidimensional space (and not the temporal component) together with the penalty component, as proposed in Section 3.4.2. Parameters: $\alpha = 1$, $\beta = 0.003$ and $\delta = 3$ (δ defined in Equation 3.3);
- **TDCK-Means** - the Temporal-Driven Constrained Clustering algorithm proposed in Section 3.4.3. $\alpha = 0$, $\beta = 0.003$ and $\delta = 3$;
- **tcK-Means** - the temporal constrained clustering algorithm proposed in [Lin & Hauptmann 2006]. It uses a threshold penalty function $w(x_i^{t_i}, x_j^{t_i}) = \alpha^* \mathbb{1}(|x_i^t - x_j^t| < d^*)$ when observations x_i and x_j are not assigned to the same cluster. It was adapted to the multi-entity case by applying it only to observations belonging to the same entity. Parameters: $\alpha^* = 2, d^* = 4$.

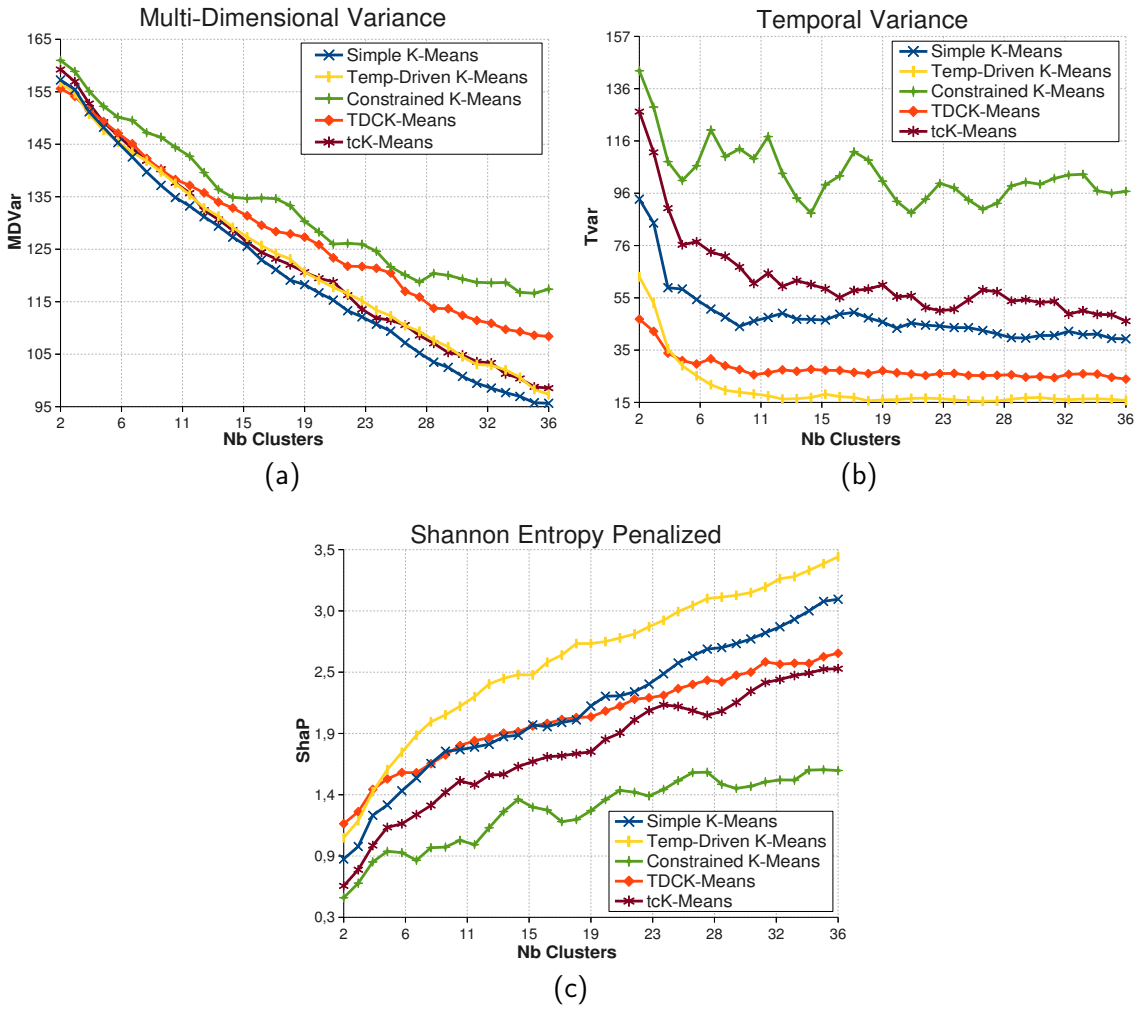


Figure 3.10 – *MDvar* (a), *Tvar* (b) and *ShaP* (c) values of the considered algorithms when varying the number of clusters.

The α^* parameter in **tcK-Means** should not be mistaken with the α parameter in **TDCK-Means**, as they do not have the same meaning. In **tcK-Means**, α^* controls the weight of the penalty function, whereas in **TDCK-Means** α is the fine-tuning parameter.

Obtained results. All the parameters are determined as shown in Section 3.5.5. Table 3.1 shows the average values for the indicators, as well as the average standard deviation (in italic) obtained by each algorithm over all values of k. The average standard deviation is only used to give an idea of the order of magnitude of the stability of each algorithm. Since Simple K-Means, Temporal-Driven K-Means and Constrained K-Means are designed to optimize mainly one component, it is not surprising that they show the best scores for, respectively, the multidimensional variance, the temporal variance and the entropy (best results in boldface). TDCK-Means seeks to provide a compromise, obtaining in two out of three cases the second best score. It is noteworthy that the proposed temporal-aware dissimilarity measure used in Temporal-Driven K-Means provides the highest stability (the lowest

Table 3.1 – Mean values for indicators and standard deviations

	Algorithm	*MDvar*		*Tvar*		*ShaP*	
Scores	**Simple K-Means**	**120.59**	*2.97*	48.01	*8.87*	2.15	*0.23*
	Temp-Driven K-Means	122.98	*2.85*	**19.97**	*5.39*	2.58	*0.18*
	Constrained K-Means	132.69	*8.07*	103.15	*42.98*	**1.24**	*0.5*
	TDCK-Means	127.81	*3.96*	27.54	*5.81*	2.06	*0.2*
	tcK-Means	123,04	*3.8*	62.44	*24.16*	1.79	*0.32*
% Gain	**Temp-Driven K-Means**	-1.99%		**58.40%**		-19.63%	
	Constrained K-Means	-10.04%		-114.84%		**42.21%**	
	TDCK-Means	-5.99%		42.64%		4.19%	
	tcK-Means	-2.03%		-30.05%		16.99%	

average standard deviation) for all indicators. Meanwhile, the constrained algorithms (Constrained K-Means and tcK-Means) show high instability, especially on *Tvar*. TDCK-Means shows a very good stability. The second part of Table 3.1 gives the relative gain of performance of each of the proposed algorithms over Simple K-Means. It is noteworthy the effectiveness of the temporal-aware dissimilarity measure proposed in Section 3.4.1, with a 58% gain of Temporal Variance and less than 2% loss of multidimensional variance. The proposed dissimilarity measure greatly enhances the temporal cohesion of the resulted clusters, without a significant scattering of observations in the multidimensional space. Similarly, the Constrained KM shows an improvement in the contiguity measure *ShaP* of 42%, while losing 10% multidimensional variance. By comparison, tcK-Means shows modest results, improving *ShaP* by only 17% and still showing important losses on both *Tvar* (-30%) and *MDvar* (-2%). This proves that the threshold penalty function proposed in literature has lower performances than our newly proposed contiguity penalty function. TDCK-Means combines the advantages of the other two algorithms, providing an important gain of 43% of temporal variance and increasing the *ShaP* measure by more than 4%. Nonetheless, it shows a 6% loss of *MDvar*.

Varying the number of clusters Similar conclusions can be drawn when varying the number of clusters. *MDvar* (Figure 3.10a) decreases, for all algorithms, as the number of cluster increases. It is well known in clustering literature that the intra-cluster variance decreases steadily with the increase of number of clusters. As the number of clusters augments, so does the differences of TDCK-Means and Constrained K-Means, when compared to the Simple K-Means algorithm. This is due to the fact that the constraints do not let too many clusters to be assigned to the same entity, resulting in the convergence towards a local optimum, with a higher value of *MDvar*. An opposite behavior is shown by the *ShaP* measure in Figure 3.10c, which increases with the number of clusters. It is interesting to observe how the *MDvar* and the *ShaP* measures have almost opposite behaviors. An algorithm that shows the best performances on one of the measures, also shows the worst on the other. The temporal divergence in Figure 3.10b shows a very sharp decrease for a low number of clusters, and afterwards remains relatively constant.

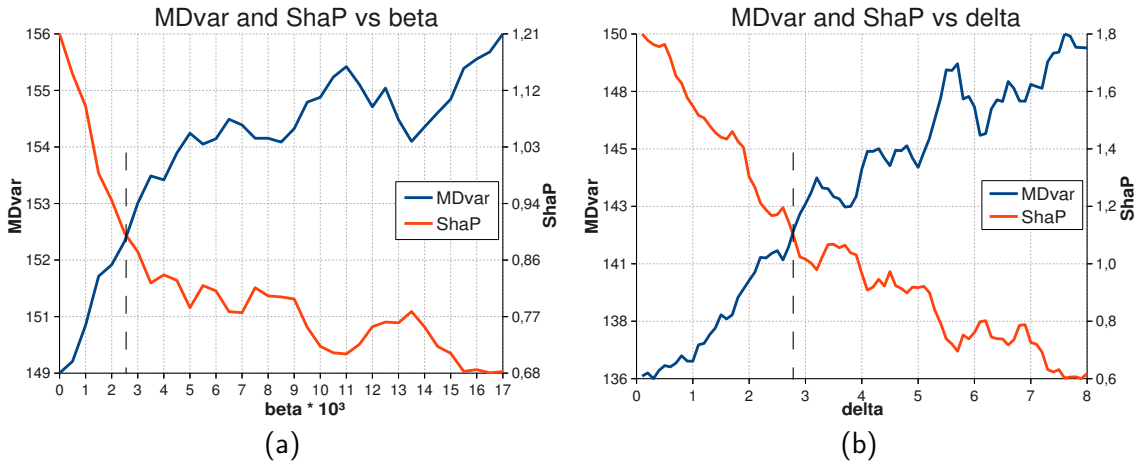


Figure 3.11 – *MDvar* and *ShaP* function of β (a) and of δ (b)

3.5.5 Impact of parameters β and δ

The β parameter controls the impact of the contiguity constraints in equation (3.3). When set to zero, no constraints are imposed, and the algorithm behaves just like the Simple K-Means. The higher the values of β, the higher the penalty inflicted when breaking a constraint. When β is set to large values, the penalty factor will take precedence over the similarity measure in the objective function. Observations that belong to a certain entity will be assigned to the same cluster, regardless of their resemblance in the description space. When this happens, the algorithm cannot create partitions with higher number of clusters than the number of entities. In order to evaluate the influence of parameter β, we execute the Constrained K-Means algorithm with β varying from 0 to 0.017 with a step of 0.0005. The value of δ is set at 3, and 5 clusters are constructed. For each value of β, we executed 10 times the algorithm and we plot the average obtained values. Figure 3.11a shows the evolution of measures *MDvar* and *ShaP* with β. When $\beta = 0$ both *MDvar* and *ShaP* have the same values as for Simple K-Means. As β increases, so does the penalty for non-contiguous segmentation of entities. *MDvar* starts to increase rapidly, while *ShaP* decreases rapidly. Once β reaches higher values, the measures continue their evolution, but with a leaner slope. In the extreme case, in which all observations are assigned to the same cluster regardless of their similarity, the *ShaP* measure will reach zero.

The δ parameter controls the width of the penalty function in equation (3.3). As Figure 3.5 shows, when δ has a low value, a penalty is inflicted only if the time difference of a pair of observations is small. As the time difference increases, the function quickly converges to zero. As δ increases, the function decreases with a leaner slope, thus also taking into account observations which are farther away in time. In order to analyze the behavior of the penalty function when varying δ, we have executed the Constrained K-Means, with δ ranging from 0.1 to 8, using a step of 0.1. β was set at 0.003 and 10 clusters were constructed. Figure 3.11b plots the evolution of measures *MDvar* and *ShaP* with δ. The contiguity measure *ShaP* decreases almost linearly as δ increases, as the series of observations belonging to each entity gets segmented in larger chunks. At the same time, the multidimensional vari-

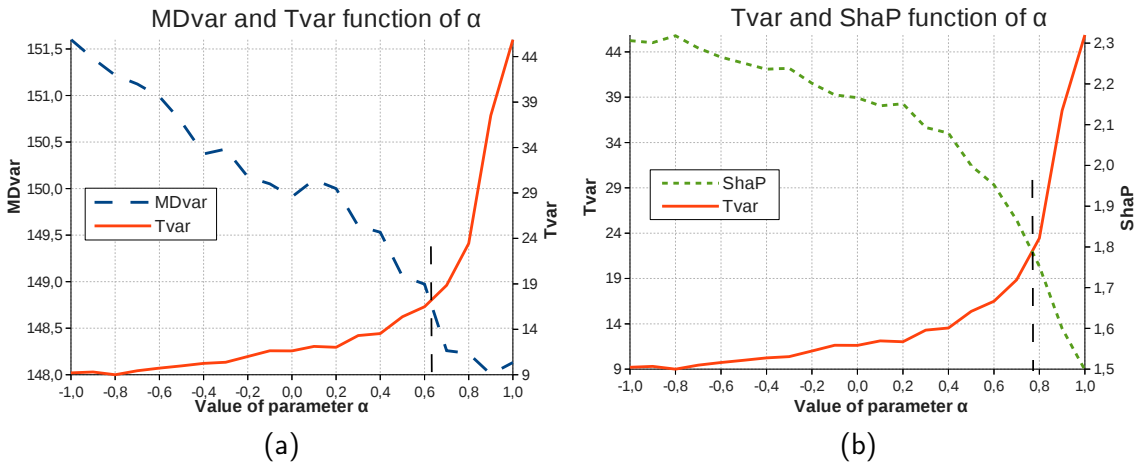


Figure 3.12 – Influence of tuning parameter α on *MDvar* and *Tvar* (a) and *Tvar* and *ShaP* (b)

ance *MDvar* increases linearly with δ. Clusters become more heterogeneous and variance increases, as observations get assigned to clusters based on their membership to an entity, rather than their descriptive similarities.

Varying α^* and d^* for the tcK-Means proposed in [Lin & Hauptmann 2006] yields similar results, with the *MDvar* augmenting and the *ShaP* descending, when α^* and d^* increase. For the tcK-Means, these evolutions are linear, whereas for the Constrained K-Means they are exponential, following a trend line of function $e^{-\frac{const}{x}}$. Plotting the evolution of the *MDvar* and the *ShaP* indicators on the same graphic, provides a heuristic for choosing the optimum values for the (β, δ) parameters of the Constrained K-Means and the TDCK-Means, respectively the (α^*, d^*) parameters of the tcK-Means. Both curves are plotted with the vertical axis scaled to the interval $[min_{value}, max_{value}]$. Their point of intersection determines the values of the parameters (as shown in Figure 3.11a and 3.11b). The disadvantage of such a heuristic would be that a large number of executions must be performed with multiple values for the parameters before the "optimum" can be found.

3.5.6 The tuning parameter α

The parameter α, proposed in Section 3.4.4, allows the fine tuning of the ratio between the multidimensional component and the temporal component in the temporal-aware dissimilarity measure. When α is close to -1, the temporal component is predominant. Conversely, when α is close to 1, the multidimensional component takes precedence. The two components have equal weights when $\alpha = 0$. To evaluate the influence of parameter α, we executed Temporal-Driven K-Means with α varying from -1 to 1 with a step of 0.1. In order not to bias the results and to evaluate only the impact of the tuning parameter, we remove the contiguity constraints from the objective function $\mathcal{J}$, by setting $\beta = 0$. For each value of α, we executed 10 times and we present the average values.

Figure 3.12a shows the evolution of measures *MDvar* and *Tvar* with α. For low values of α, the value of the temporal-aware dissimilarity measure is given mainly by the temporal

component, so *Tvar* shows its lowest value, while *MDvar* presents its maximum. As α increases, *MDvar* decreases as more importance is given to the multidimensional component. For $\alpha \in (-1, 0]$, the importance of the temporal component remains intact, the increase of *Tvar* is solely the result of the algorithm converging to a local optimum which also takes into account the multidimensional component. For $\alpha \in [0, 1)$, the impact of the multidimensional component stays constant, whereas the importance of the temporal components diminishes. As a result, *MDvar* continues its decrease and *Tvar* increases sharply. For $\alpha = 1$ the temporal component is basically ignored from the measure. The Temporal-Driven K-Means behaves just like Simple K-Means. Figure 3.12b shows the evolution of *ShaP* alongside *MDvar*. Even if the contiguity penalty component was neutralized by setting $\beta = 0$, the value of *ShaP* is not constant, but it descends with α. For low values of α, the temporal component is predominant in the similarity measure. This generates partitions where every cluster regroups all observations from a specific period, regardless of their multidimensional description. This means that all entities have segments in all the clusters, which leads to a high value of *ShaP*.

It is noteworthy that the evolution of the indicators is not linear with α. As α increases, *Tvar* augments only very slowly and picks up the pace only for large values of α. This indicates that the temporal component has an inherent advantage over the multidimensional one. As we presumed in Section 3.4.4, this is due to the intrinsic nature of the dataset and the main reason why the tuning parameter α was introduced. The distributions of observations in the multidimensional and temporal spaces is different: in the temporal space, the observations tend to be evenly distributed, whereas in the multidimensional description space, they cluster together. To quantify this, we calculate the ratio between average standard deviation and average distances between pairs of observations:

$$r_{dim} = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} \frac{stdev\left(\left\{||x_i^{dim} - x_j^{dim}||^2 \middle| x_j \in \mathcal{X},\ i \neq j\right\}\right)}{\frac{1}{|\mathcal{X}|} \sum_{\substack{j=1 \\ j \neq i}}^{|\mathcal{X}|} ||x_i^{dim} - x_j^{dim}||^2}$$

where *dim* is replaced with d or t (the descriptive or the temporal dimension). On *Comparative Political Data Set I*, $r_d = 29.5\%$ and $r_t = 65.3\%$. This shows that observations are a lot more dispersed in the temporal space than in the multidimensional description space. This explains why *Tvar* augments very slowly with α and starts to increase more rapidly only starting from $\alpha = 0.4$.

Following the heuristic proposed in Section 3.5.5, we can determine a "compromise" value for α. As shown in Figure 3.12, all vertical axes are magnified between their functions' minimum and maximum values. The "compromise" value for α is found at the intersection point of *MDvar* and *Tvar* (and *MDvar* and *Shap*). This value is set around 0.7, showing the dataset's bias towards the temporal component. This technique for setting the value of the tunning parameter is just a heuristic, the actual value of α is dependent on the dataset. This is why we are currently working on a method, inspired from multi-objective optimization using evolutionary algorithms [Zhang & Li 2007] to automatically determine the values of α, as well as the other parameters of TDCK-Means (β and δ).

3.6 Current work: Role identification in social networks

We are at present working on multiple extensions and applications of our work. In this section, we present the most advanced of them: the application of TDCK-Means to user social role identification in social networks. This is an ongoing work, the result of the research collaboration with the *Technicolor's Research & Innovation Laboratories*[4] in Rennes, France. Technicolor's research interests include those parts of the Web related to Cinema and Television, *e.g.* Web forums, social media, and professional sites. The end purpose is to enrich the content by linking meta-data, extracted by analyzing the usage patterns and semantic information. This collaboration allows us to test and further analyze the behavior of our proposed TDCK-Means algorithm to social network data. This work is in an advanced state: the submission of an article is already planned.

In the remainder of this section, we present the general context of this application of TDCK-Means and the used dataset (in Section 3.6.1), followed by the description of the user role identification framework (in Section 3.6.2) and some preliminary results that we obtained (in Section 3.6.3).

3.6.1 Context

The base hypothesis of this work is that, when interacting in an online community, a user plays multiple roles, during the given period of time. These roles are temporally coherent (when in a role, the user's activity is uniformly similar) and he/she can change between roles. We denote these roles as *behavioral roles*. The global *social role* is constructed as a mixture of different behavioral roles, which incorporates the dynamics of behavioral transitions. Therefore, we define the user social role as a succession of behavioral roles.

We construct the user social roles, based on a social network, inferred from online discussion forums. The *social roles* are identified in a three phase framework: (a) behavioral characteristics are identified, based on the structure of the inferred social network, (b) *behavioral roles* are created using TDCK-Means and (c) the user *social roles* are determined based in the transitions between behavioral roles.

Dataset and social network creation In this application of the TDCK-Means, we used the `TWOP` [Anokhin *et al.* 2012] dataset, which is an online forum discussion dataset. We discuss in more details the forum discussion online environment in Chapter 7, where we introduce `CommentWatcher`, an online forum analysis platform. In online forums, users can start new discussion threads or they reply to other users' messages, through a quote citing mechanism. Discussions are formed by multiple users simultaneously answering one to another. This structure can be used to infer an implicit online social network of users: two users are considered to have a relation when they reply one to another. This implicit social network is modeled as an oriented weighted graph. The nodes of the graph are the users posting in the forums, whereas the arcs signal their relations. A directed arc is added from user A to user B when A replies to B. The strength of the relation between A and B is directly proportional with the number of replies of A to B.

4. https://research.technicolor.com/rennes/

The TWOP dataset is constructed based on the *Television Without Pity*[5] forum website. It contains 58994 post of 7066 authors. It was constructed by parsing messages posted in 6 forums, corresponding to 6 TV series, during the year of 2007.

3.6.2 The framework for identifying social roles

The *social roles* are identified by passing through *behavioral roles*. The underlying assumption is that the user's behavior might change during the observed period of time. For example, he/she might be very active during a certain period, follow by a period of being absent from the online community or simply exchanging less with other users. We consider this temporally coherent periods as being part of the same *behavioral role*. We, therefore, define the general *social role* as a succession of behavioral roles. We determine the *social roles* using a three-phase framework:

(a) we calculate the temporal behavioral characteristics, based on the structure of the inferred social network;

(b) the *behavioral roles* are extracted using TDCK-Means, by constructing temporally coherent clusters and contiguously segmenting the temporal measurement vectors associated to each users;

(c) the user *social roles* are determined based on the transitions between the behavioral roles.

In the next paragraphs of this section, we detail each of these phases.

Determining behavioral characteristics The users interactions are quantified in *phase (a)* as shown in [Anokhin *et al.* 2012], by analyzing the directed graph of user interactions. We use measures adapted from the citation analysis literature to provide a measure of interaction and importance to our forum users. In particular, two well-known citation metrics are adapted: the *h-index* [Hirsch 2005] and its successor, the *g-index* [Egghe 2006]. The following characteristics are computed and used to describe the activity of a user:

- the node's *in/out g-Index*. Based on the *g-index*, it measures how active is the neighborhood of the node.
- *catalitic power*. This indicator can differentiate between people who constantly receive replies and those who start just one or two debates.
- *weighted in/out-degree*. It measures the "quantity" of communication in the neighborhood of the node.
- *activity* is the number of posts of the user.
- *cross-thread entropy* measures the user's focus on a thread.

These measures are rendered temporal by calculating them on an adaptive-length time window, which ensures that periods with low activity do not bias the temporal clustering (*i.e.*, in the summer there is no broadcast of TV series and, consequently, there is almost no activity on the chosen television forum site in the summer).

Creating user behavioral roles In *phase (b)*, the proposed temporal-aware constrained clustering algorithm (TDCK-Means) is used to detect the behavioral roles. In Section 3.4,

5. http://www.televisionwithoutpity.com/

the clustered observations are triples $(entity, time, description)$. In the case of user role detection application, the *entities* are the users and the *description vector* is defined in the multi-dimensional description space, in which each behavioral measure is a dimension. As already discussed in Section 3.4.4, the importance of the temporal component is dependent on the application. There is a difference between the application of TDCK-Means to the detection of country evolutions and the construction of behavioral roles. Evolution phases of countries are inherently temporal, while the behavioral roles can appear during the entire observed period. Consequently, we reduce the importance of the temporal component by setting the α parameter (see Section 3.5.6) to values close to 1. The result of executing TDCK-Means are k temporal coherent clusters $(\mu_l, l = 1, 2, ..., k)$, which are interpreted as behavioral roles. This already allows, for a given user, to evaluate the stability of his/her behavior.

Creating user social roles In *phase (c)*, for each user u_i, we estimate the transition matrix $\Psi_i = \{\psi^i_{s,t}\}$, where $\psi^i_{s,t}$ is the probability of user u_i to have a transition from the behavioral state μ_s to the behavioral state μ_t. The matrices Ψ_i have the tendency of being rare and have high values on the main diagonal, since TDCK-Means privileges (a) consecutive observation to belong to the same behavioral role and (b) that a user passes through rather few behavioral roles (see the discussion in Section 3.2).

Each social role is a mixture of different behavioral roles and it incorporates the dynamics of behavioral transitions. In other words, we interpret as a social role a succession of transition through behavioral roles (similar to our previous application, the typical evolutions of countries). We use a simple K-Means to regroup the users' transition probabilities. Notice that, in this second clustering, the individuals being clustered are the entities (*i.e.*, the users) and not the observations, as it was in the case of TDCK-Means. The users' transitions are represented by the matrices Ψ_i and the obtained cluster centroids are (a) described in the same numeric space as the clustered instances and (b) interpreted as the social roles. Each social role is, therefore, a matrix which gives two types of information: (a) the probability of transitions from one role to the other and (b) the mixture of behavioral roles, denoted by the values on the main diagonal ($\psi^i_{s,s}$ is the probability that the user u_i passes from the behavioral role μ_s to μ_s, which can be interpreted as the proportion of the time in which u_i stays in the behavioral role μ_s).

Such a representation has the inconvenient of being time orderless: it gives the proportion in which behavioral roles are present in each social role, but it does not give their temporal order of succession. One of the current work we are undergoing at the present deals with inferring a graph structure for the constructed clusters (more details in Section 3.7 and in Chapter 8). A graph approach would solve the order problem and detecting a social role becomes detecting frequent paths in the generated graph.

3.6.3 Preliminary experiments

This section presents the experiments and the preliminary results that were performed and obtained with the framework presented in Section 3.6.2. There is still work to be done, mainly in parameter choice, interpretation and qualitative evaluation. A thorough parameter sweep needs to be performed to motivate the choice of parameters and qualitative evaluation

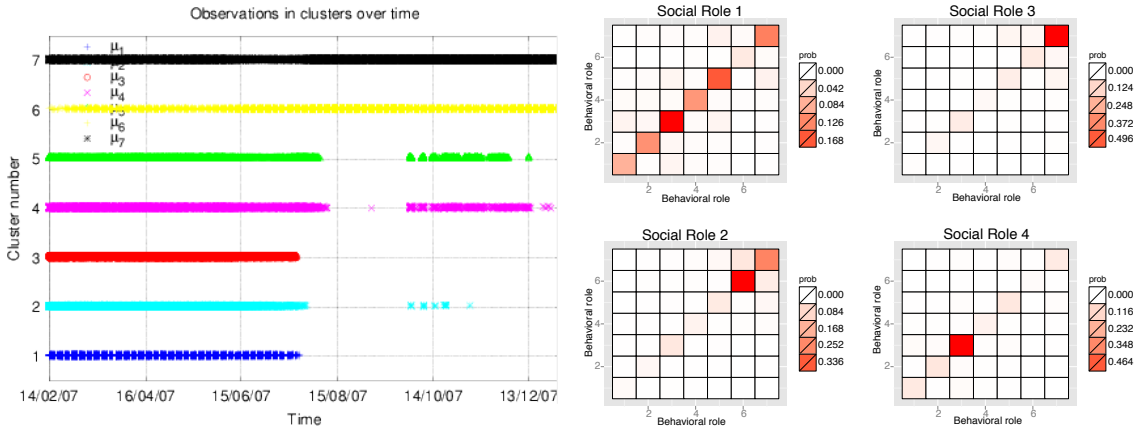


Figure 3.13 – The presence of the behavioral roles during the considered period of time (a) heat maps showing the transition probabilities between behavioral roles for each extracted social role (b).

measures are still lacking. Whatsoever, the first qualitative results are encouraging and we choose to present them here.

Table 3.2 – The behavioral roles represented by the centroids constructed by TDCK-Means

Role	Date	*in g-Index*	*out g-Index*	*cat. power*	*in-deg.*	*out-deg.*	*entropy*	*activ.*
μ_1	20/04/07	-0.898	-0.632	-0.856	-0.863	-0.540	1.333	-0.223
μ_2	22/04/07	-0.075	-0.886	-0.038	0.013	-0.892	-0.579	-0.861
μ_3	24/04/07	-0.977	-0.844	-0.964	-1.007	-0.843	-0.531	-0.923
μ_4	09/05/07	0.113	-0.210	0.068	0.249	-0.102	1.494	0.296
μ_5	07/06/07	-0.152	0.179	-0.202	-0.152	0.189	-0.351	-0.064
μ_6	30/08/07	1.765	1.771	1.730	1.771	1.759	1.126	1.88
μ_7	16/09/07	0.928	0.908	0.955	0.856	0.819	-0.431	0.736

In *phase (a)*, the behavioral characteristics are calculated, normalized and the values are transformed into a fixed range by first performing min-max normalization and a log transformation, afterwards. In *phase (b)* of the social role extraction framework, we execute the TDCK-Means algorithm with the parameters $k = 7, \alpha = 0.95, \beta = 0.005, \delta = 1.0$ and we construct 7 temporal clusters. These clusters regroup similar user activities and are interpreted as behavioral roles. Figure 3.13a shows the presence of each of the behavioral roles over the course of the year 2007. We can see that some behavioral roles are only present in the first half of the year. This is due to the particularity of the `TWOP` dataset that there is a very low activity during the summer (when there is no broadcast of TV series). Therefore, there is a temporal gap in the user activities and this causes TDCK-Means to create temporal clusters with a limited time-span.

Table 3.2 shows the centroids of each of the temporal clusters. Since a centroid sum-

marizes the main features of the user activities in its cluster, we interpret the behavioral roles based on the associated centroids μ_l (in the following discussion we also denote a behavioral role with the name of the associated centroid). We can see, for example, that user activities that fall into the behavioral role μ_1 describe users who post across multiple forums (hence the high *entropy*), but do not seem to engage other community members (low *in/out g-Index*, low *in/out-degree*, low *catalytic power*). By contrast, behavioral role μ_4 also exhibits a high *entropy*, but there is evidence that users in this role are receiving replies (high *in-degree*). However, their replies are not from highly connected users, evidenced by the slightly positive *in g-Index* value.

In *phase (c)*, a K-Means clustering is performed on the behavioral transition probability vectors, as described in Section 3.6.2 and 4 clusters are constructed, which are interpreted as social roles. The *social role* is constructed as a mixture of different behavioral roles. Figure 3.13b shows, for each constructed social role, the obtained transition probabilities, depicted as heat maps. Most of the non-null values are on the main diagonal (lines are numbered from bottom to top, columns from left to right). The values on the main diagonal give the behavioral composition within each social role. Role 1 represents users who are active members, able to generate and participate in active conversations within the forum. Sometimes, these conversations are with highly connected users, though mostly at random. Role 2 represents highly influential central figures, who take part and form the basis of conversation in multiple sub-forums. Role 3 represents the slightly less active and more focused users. Role 4 regroups users who are present throughout the peak of conversations, and reduce their activity afterwards.

Interestingly, although users present in Role 1 move between nearly all the behavioral roles, they are very rarely present in μ_6. Table 3.2 shows that in this behavioral role, users are highly central to conversation on more than one forum. This shows that most of the users on the `TWOP` forum are only interested in a single show, and subsequently will always have low entropy. Another distinct role played within the forum is the behavioral role μ_4. Users presenting this role, although they are unable to generate (or contribute to) any conversation, remain on the forum. Despite being ignored, users presenting the behavioral role μ_4 continue to participate in conversations.

In conclusion This section presented some of the work we are currently performing with the TDCK-Means algorithm in particular, and with the temporal dimension of data in general. We have shown that our proposed temporal-aware clustering algorithm has new promising investigation fields and can be applied to other type of data than the one presented in Section 3.5. The interpretation of the constructed social roles is still difficult, given that they are expressed as an orderless mixture of behavioral roles. Our plans are to extend TDCK-Means to include the inference of a graph structure of the constructed clusters. Constructing the social roles as a path in the behavioral role graph would render the interpretation easier.

3.7 Conclusion and future work

In this chapter we have studied the detection of typical evolutions from a collection of observations. The presented work tackles with one of our central research challenges, *i.e.*, dealing with the temporal dimension of complex data. As we discussed at the beginning of the chapter, we consider that the temporal dimension is more than just another descriptive dimension of data, since it changes the definition of the learning problem. Therefore, one of the original contributions presented in this chapter is a novel method to introduce temporal information directly into the dissimilarity measure, weighting the descriptive component by the temporal component. This new measure allows us to weight the importance of the temporal component relative to the descriptive component and fine-tune its impact on the learning process.

We have also proposed TDCK-Means, an extension of K-Means, which uses the temporal-aware dissimilarity measure and a new objective function which takes into consideration the temporal dimension. We use a penalty factor to make sure that the observation series related to an entity get segmented into continuous chunks. We infer a new centroid update formula, where elements distant in time contribute less to the centroid than the temporally close ones. We have shown that our proposition consistently improves temporal variance, without any significant losses in the multidimensional variance. The algorithm can be used in other applications where the detection of typical evolutions is required, *e.g.* career evolution of politicians or abnormal disease evolution. We have shown how our proposal can be adapted to another specific problem, *i.e.* role identification in social networks, and another dataset.

Perspectives of our work In our current work, we have only detected the centroids that serve as the evolution phases. We are currently working on an extension of TDCK-Means, which has embedded in the algorithm the construction of the evolution graph (as shown in Figure 3.2a, p. 32). The objective is to construct, starting from the available complex data, a graph structure of the clusters. The idea is to estimate, in addition to the centroids and the temporal membership of observation to clusters, an adjacency matrix that defines the graph. This will allow an succinct description of the evolution of an entity as a path thought the constructed graph. Another direction of research will be describing the clusters in a human readable form. We work on means to provide them with an easily comprehensible description by introducing temporal information into the unsupervised feature construction algorithm (we give more details about this current work in Chapters 4 and 8). We are also experimenting a method for setting automatically the values of TDCK-Means's parameters (α, β and δ), by using an approach inspired from multi-objective optimization using evolutionary algorithms [Zhang & Li 2007].

The work presented in this chapter was published in the 24th IEEE International Conference on Tools with Artificial Intelligence, receiving the **Best Student Paper Award** [Rizoiu *et al.* 2012].

Chapter 4

Using Data Semantics to Improve Data Representation

Contents

4.1 Learning task and motivations

Leveraging semantics when dealing with complex data is one of the core research challenges of the work in this thesis. This chapter tackles the crucial learning task of **constructing a semantic-improved representation space** for describing the data. While

in the rest of our work, we either use external available information or the temporal aspect of data in order to infer a more complete knowledge, in the work presented in this chapter we concentrate on the semantics already available in the data itself.

In the context of automatic classification, a useful feature needs to express new information as compared to other features. Correlated features do not bring any new information (this problem was already presented in Section 2.1.4, p. 18). Whatsoever, when two features co-occur, it is usually the result of a semantic connection between the two. Therefore, in this chapter we have two missions: (a) improve the representation space of data by removing correlations between features and (b) discover semantic links between features by analyzing their co-occurrences in the data. To address these tasks, we propose a novel unsupervised algorithm, **uFC**, that improves the representation space by reducing the overall correlation between features, while discovering semantics links between features by performing feature construction: pairs of highly co-occurring features are replaced with Boolean conjunctions. The total correlation of the new feature set is reduced and the semantically-induced co-occurrences in the initial set are emphasized.

Motivations One of the limitations of representing data in the *feature-value vector* format is that the supplied features are sometimes not adequate to describe, in terms of classification, the data. This happens, for example, when general-purpose features are used to describe a collection that contains certain relations between individuals. *E.g.*, a user of an online photo sharing service might find that the proposed generic labels are not adapted for tagging his/her photo collection. When labeling an image of a cascade, what should he/she use? *water*, *cascade* or both (since a cascade is made out of water). Given this problem, our departing premise is that it exists an underlying semantic in the feature set, which is dependent on the dataset: the way features are organized over the observations is not random. On the contrary it gives precious information about existing relations between features. If features co-occur in the description of individuals, we consider that it is due to a semantic connection between them. In the above example of the user labeling images, the tags *people* and *trees* co-occur in pictures depicting a barbecue because it exists a link between the two in the given context, and a new feature *people and trees* should better describe this context. Furthermore, considering the case of complete labeling (*i.e.*, a labeling in which no labels are missing, we further discuss it in Section 4.10), given the co-occurrence pattern between *water* and *cascade* (*water* always appears together with *cascade*), we can deduce that it exists a special type of relation between the two (*e.g.*, a "type-of" relation).

In our work presented in this chapter, we are interested in how to augment the expressive power of the employed features set, by taking into account the underlying semantics present in the dataset. Figure 4.1 presents a streamlined schema of this treatment. No external information intervenes, the feature set is rewritten based only on the information contained in the dataset. This means that the instances are translated into another space, defined by the new feature set. The work presented in this chapter originated in the need to reorganize a set of labels used for tagging images. We employ these labels later, in Chapter 5, in order to construct a semantic-enriched numerical representation for images.

The remainder of this chapter is structured as follows. The rest of this section further discusses the need to re-organize a feature set and presents an overview of the proposed solutions. In Section 4.2, we briefly present related works that deal with rewriting a feature

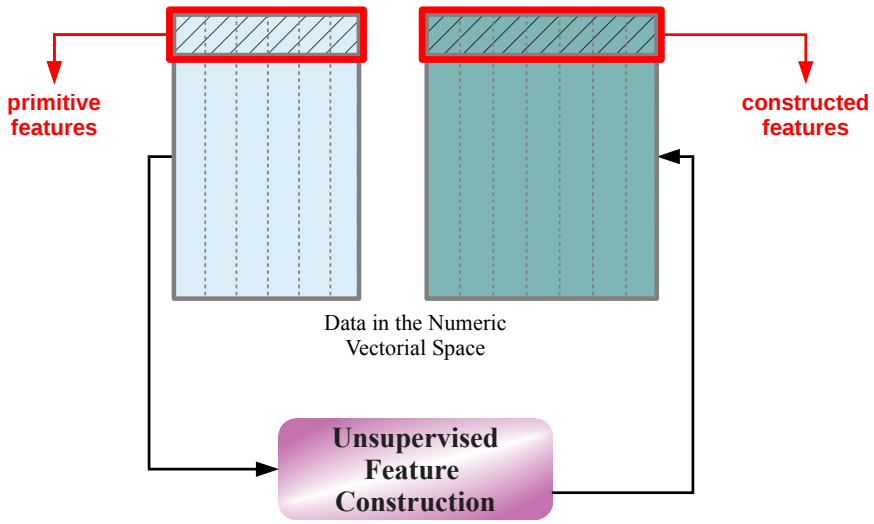


Figure 4.1 – Streamlined schema of improving data representation.

set. In Sections 4.3 and 4.4, we present our proposed algorithms and in Section 4.5 we describe the evaluation metrics and the complexity measures. In Section 4.6, we perform a set of initial experiments and outline some of the inconveniences of the algorithms. In Section 4.7, by use of statistical hypothesis testing, we address these weak points, notably the choice of the threshold parameter. In Section 4.8, a second set of experiments validates the proposed improvements. The evaluation metrics we employ are opposing criteria and, in order to optimize them, we make use of multi-objective optimization techniques that we summarize in Section 4.9. Finally, Section 4.10 draws the conclusion and outlines some current and future works.

4.1.1 Why construct a new feature set?

In the context of automatic classification (supervised or unsupervised), a useful feature needs to express new information as compared to other features. A feature p_j, that is highly correlated with another feature p_i, does not bring any new information, since the value of p_j can be deduced from that of p_i. Subsequently, one could filter out "irrelevant" features before applying the classification algorithm. But by simply removing certain features, one runs the risk of losing important information of the **semantic connection between the features**, and this is the reason why we choose to perform **feature construction** instead (we have discussed the differences between feature selection and feature construction in Section 2.1.4, p. 18). Feature construction attempts to increase the expressive power of the original features by discovering missing information about relationships between features.

We deal primarily with datasets described with **Boolean** features. In real-life datasets, most binary features have specific meanings. For example, a collection of images is tagged using a set of labels (the Boolean features), where each label marks the presence (**true**) or the absence (**false**) of a certain object in the image or give information about the context of the image. The objects could include: *water*, *cascade*, *manifestation*, *urban*, *groups* or

Figure 4.2 – Example of images tagged with {*groups*, *road*, *building*, *interior*}.

interior, whereas the contexts can be *holiday* or *evening*. In the given example, part of the semantic structure of the feature set can be guessed quite easily. Relations like "is-a" and "part-of" are fairly intuitive: *cascade* is a sort of *water*, *paw* is part of *animal etc.* But other relations might be induced by the semantics of the dataset (*i.e.*, the images in our example). $manifestation$ will co-occur with *urban*, for they usually take place in the city. Figure 4.2 depicts a simple image dataset described using the feature set {*groups*, *road*, *building*, *interior*}. The feature set is quite redundant and some of the features are non-informative (*e.g.*, the feature *groups* is present for all individuals). Considering co-occurrences between features, we could create the more eloquent features *people at the interior and not on the road* ($groups \wedge \neg road \wedge interior$, describing the top row) and *people on the road with buildings on the background* ($groups \wedge road \wedge building$, describing the bottom row).

The idea is to create a data-dependent feature set, so that the new features are as independent as possible, limiting co-occurrences between the new features. At the same time, given that one of our directive guidelines is to create human comprehensible outputs, the newly created features should be easily comprehensible. The advantage is that, the newly constructed features express the semantic connections between the primitive features. For example, the fact that a newly constructed feature $holiday \wedge water$ is set for a great number of images is a good indicator of a vacation on the seaside. This already gives information about the dataset, without even looking at the images.

4.1.2 A brief overview of our proposals

In order to obtain good results in classification tasks, many algorithms and preprocessing techniques (*e.g.*, SVM [Cortes & Vapnik 1995], PCA [Dunteman 1989] *etc.*) deal with non-adequate variables by changing the description space (internally for the SVM). The main drawback of these approaches is that they function as a black box, where the new representation space is either hidden (for SVM) or completely synthetic and incompre-

hensible to human readers (PCA). The literature also proposes algorithms that construct features based on the original user-supplied features. However, to our knowledge, all of these algorithms construct the feature set in a supervised way, based on the class information, supplied *a priori* with the data.

The novelty of our proposals Relative to existing solutions in the literature, our novel solutions have two advantages. In addition to constructing a representation space in which features co-occur less, they (a) produce humanly comprehensible features and (b) they function in the absence of pre-classified examples, in an unsupervised manner. The first algorithm we propose is an adaptation of an established supervised algorithm, making it unsupervised. For the second algorithm, we have developed a completely new heuristic that selects, at each iteration, pairs of highly correlated features and replaces them with conjunctions of literals. Therefore, the overall redundancy of the feature set is reduced. Later iterations create more complex Boolean formulas, which can contain negations (meaning absence of features). We use statistical considerations (hypothesis testing) to automatically determine the value of parameters depending on the dataset, and a *Pareto front* [Sawaragi *et al.* 1985]-inspired method for the evaluation. The main advantage of the proposed methods over PCA or the kernel of the SVM is that the newly-created features are comprehensible to human readers (features like $people \wedge manifestation \wedge urban$ and $people \wedge \neg urban \wedge forest$ are easily interpretable). As mentioned earlier, our algorithms function in a complete labeling paradigm. We discuss in Section 4.10 how our approaches can be adapted to function with missing label.

Using Boolean features Just like many other algorithms in the feature construction literature, our algorithms are limited to binary features. Any dataset described using the *feature-value vector* format can be converted to a binary format using discretization and binarization. Data Mining presents extensive work [Fayyad & Irani 1993] on the discretization of continuous features. While it is true that such a process has its drawbacks (*e.g.*, loss of details and order), there are advantages to discretization: it renders the learning algorithms less sensible to outliers and noise, it deals better with missing values, by creating a special feature which regroups them and it avoids the problems of asymmetry of the distribution of observations corresponding to a variable. Some researchers [Kotsiantis & Kanellopoulos 2006, Elomaa & Rousu 2004] even conclude that "most machine learning algorithms produces better models when performing discretization on continuous variables".

Some of the shortcomings of discretization can be overcomed by using cut-points. Instead of discretizing by creating the feature "$a \in [v_1, v_2]$", we can create two features "$a < v_1$" and "$a < v_2$". A value lower than v_1 would have both "$a < v_1$" and "$a < v_2$" set to true. In this way the order relation between the binary features is preserved. This type of discretization is inspired from the proportional-odds cumulative model, used with ordinal data in the field of logistic regression [Agresti 2002].

4.2 Related work

The literature proposes methods for augmenting the descriptive power of features. [Liu & Motoda 1998] collects some of them and divides them into three categories: feature selection, feature extraction and feature construction.

Feature selection [Lallich & Rakotomalala 2000, Mo & Huang 2011] seeks to filter the original feature set in order to remove redundant features. This results in a representation space of lower dimensionality. These approaches address directly the problem of high data dimensionality (described in Section 2.1.4, p. 18)) and created a space which is more adapted for machine learning. Whatsoever, removing features runs the risk of loosing potentially interesting information. In the example of the co-occurrence between *cascade* and *water*, a feature selection approach might remove the *cascade*, since *water* is more general. But the "is-a" induced by the dataset would be lost. Therefore, we do not consider any further this kind of approaches.

Feature extraction is a process that builds a set of new features from the original features through functional mapping [Motoda & Liu 2002]. For example, while the **SVM algorithm** [Cortes & Vapnik 1995] does not properly build a new description space (the kernel function only maps the description space into a predefined larger space, which is separable in the context of supervised learning), we can assimilate this approach to a feature extraction since the purpose is to better describe the data and the new space is difficult to comprehend from a semantic point of view. Furthermore, supervised and non-supervised algorithms can be boosted by pre-processing with **principal component analysis** (PCA) [Dunteman 1989]. PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables, called *principal components*.

Another technique which can be associate with feature extraction is **Manifold learning** [Huo *et al.* 2006], which pursuits the goal to embed data that originally lies in a non-linear manifold into a lower dimensional space, while preserving characteristic properties. It can be assimilated to feature extraction methods, since the purpose is to build a new lower dimensional description space for the data.

Feature extraction mainly seeks to reduce the description space and redundancy between features (exception the SVM, which builds a higher dimensional space than the original data). The problem with these approaches is that newly created features are rarely human comprehensible and difficult to interpret, from a semantic point of view. Therefore, we consider feature extraction methods inadequate for detecting relations between the original features.

Feature Construction Feature Construction is a process that discovers missing information about the relationships between features and augments the space of features by inferring or creating additional features [Motoda & Liu 2002]. The accent in feature construction, unlike feature extraction presented earlier, is on the comprehensibility of the newly created features. These methods usually construct a representation space with a larger dimension than the original space. Constructive induction [Michalski 1983] is a process of constructing new features using two intertwined searches [Bloedorn & Michalski 1998]: one in the representation space (modifying the feature set) and another in the hypothesis space (using

Algorithm 2 General feature construction schema.

Input: P – set of primitive user-given features
Input: I – the data expressed using P which will be used to construct features
  Inner parameters: Op – set of operators for constructing features, M – machine learning algorithm to be employed
Output: F – set of new (constructed and/or primitives) features.
  $F \leftarrow P$
  $iter \leftarrow 0$
  repeat
    $iter \leftarrow iter + 1$
    $I_{iter} \leftarrow$ **construct**(I_{iter-1}, F)
    $output \leftarrow$ Run $\mathbf{M}(I_{iter}, F)$
    $F \leftarrow F \bigcup$ new feat. constructed with $\mathbf{Op}(F, output)$
    prune useless features in F
  until stopping criteria are met.

classical learning methods). The actual feature construction is done using a set of constructing operators and the resulted features are often conjunctions of primitives, therefore easily comprehensible to a human reader. Feature construction has mainly been used with decision tree learning. New features served as hypotheses and were used as discriminators in decision trees. Supervised feature construction can also be applied in other domains, like decision rule learning [Zheng 1995].

Algorithm 2, presented in [Gomez & Morales 2002, Yang *et al.* 1991], represents the general schema followed by most feature construction algorithms. The general idea is to start from I, the dataset described with the set of primitive features. Using a set of constructors and the results of a machine learning algorithm **M**, the algorithm constructs (in the **construct** step) new features that are added to the feature set. In the end, useless features are pruned. These steps are iterated until some stopping criterion is met (*e.g.*, a maximum number of iterations performed or a maximum number of created features).

Most constructive induction systems construct features as conjunctions or disjunctions of literals. Literals are the features or their negations. E.g., for the feature set $\{a, b\}$ the literal set is $\{a, \neg a, b, \neg b\}$. Operator sets $\{AND, Negation\}$ and $\{OR, Negation\}$ are both complete sets for the Boolean space. Any Boolean function can be created using only operators from one set. **FRINGE** [Pagallo & Haussler 1990] creates new features using a decision tree that it builds at each iteration. New features are conjunctions of the last two nodes in each positive path (a positive path connects the root with a leaf having the class label **true**). The newly-created features are added to the feature set and then used in the next iteration to construct the decision tree. This first algorithm of feature construction was initially designed to solve replication problems in decision trees. The replication problem [Pagallo & Haussler 1990] states that in a decision tree representing a Boolean function in its Disjunctive Normal Form, the same sequence of decision tests leading to a positive leaf is replicated in the tree.

Other algorithms have further improved this approach. **CITRE** [Matheus 1990] adds other search strategies like *root* (selects first two nodes in a positive path) or *root-fringe*

(selects the first and last node in the path). It also introduces domain knowledge by applying filters to prune the constructed features. **CAT** [Zheng 1998] is another example of a hypothesis-driven constructive algorithm similar to **FRINGE**. It also constructs conjunctive features based on the output of decision trees. It uses a dynamic-path based approach (the conditions used to generate new features are chosen dynamically) and it includes a pruning technique.

Alternative representations There are alternative representations, other than conjunctive and disjunctive. The $M-of-N$ and $X-of-N$ representations use feature-value pairs. A feature-value pair $AV_k(A_i = V_{ij})$ is **true** for an instance if and only if the feature A_i has the value V_{ij} for that instance. The difference between $M-of-N$ and $X-of-N$ is that, while the second one counts the number of true feature-value pairs, the first one uses a threshold parameter to assign a value of truth for the entire representation. The algorithm **ID2-of-3** [Murphy & Pazzani 1991] uses $M-of-N$ representations for the newly-created features. It has a specialization and a generalization construction operator and it does not need to construct a new decision tree at each step, but instead integrates the feature construction into the decision tree construction. The **XofN** algorithm [Zheng 1995] functions similarly, except that it uses the $X-of-N$ representation. It also takes into account the complexity of the features generated.

Comparative studies like [Zheng 1996] show that conjunctive and disjunctive representations have very similar performances in terms of prediction accuracy and theoretical complexity. $M-of-N$, while more complex, has a stronger representation power than the two before. The $X-of-N$ representation has the strongest representation power, but the same studies show that it suffers from data fragmenting more than the other three.

The problem with all of these algorithms is that they all work in a supervised environment and they cannot function without a class label. In the following sections, we will propose two approaches towards unsupervised feature construction.

4.3 uFRINGE - adapting FRINGE for unsupervised learning

We propose **uFRINGE**, an unsupervised version of FRINGE, one of the first feature construction algorithms. FRINGE [Pagallo & Haussler 1990] is a framework algorithm (see Section 4.2), following the same general schema shown in Algorithm 2. It creates new features using a logical decision tree, created using a traditional algorithm like ID3 [Quinlan 1986] or C4.5 [Quinlan 1993]. Taking a closer look at FRINGE, one would observe that its only component that is supervised is the decision tree construction. The actual construction of features is independent of the existence of a class attribute. Hence, using an "unsupervised decision tree" construction algorithm renders FRINGE unsupervised.

Clustering trees Clustering trees [Blockeel *et al.* 1998] were introduced as generalized logical decision trees. They are constructed using a top-down strategy. At each step, the cluster under a node is split into two subclusters, seeking to maximize the intra-cluster variance. The authors argue that supervised indicators, used in traditional decision trees algorithms, are special cases of intra-cluster variance, as they measure intra-cluster **class**

diversity. Following this interpretation, clustering trees can be considered as generalizations of decision trees and are suitable candidates for replacing ID3 in **uFRINGE**.

Adapting FRINGE to use clustering trees is straightforward: it is enough to replace **M** in Algorithm 2 with the clustering trees algorithm. At each step, **uFRINGE** constructs a clustering tree using the dataset and the current feature set. Just like in FRINGE, new features are created using the conditions under the last two nodes in each path connecting the root to a leaf. FRINGE constructs new features starting only from positive leaves (leaves labelled true). But unlike decision trees, in clustering trees the leaves are not labelled using class features. Therefore, in **uFRINGE**, we choose to construct new features based on all paths from root to a leaf.

Newly-constructed features are added to the feature set and used in the next classification tree construction. The algorithm stops when either no more features can be constructed from the clustering tree or when a maximum allowed number of features have already been constructed.

Limitations **uFRINGE** is capable of constructing new features in an unsupervised context. It is also relatively simple to understand and implement, as it is based on the same framework as FRINGE. However, it suffers from a couple of drawbacks. Constructed features tend to be redundant and contain doubles. Newly-constructed features are added to the feature set and are used, alongside old features, in later iterations. Older features are never removed from the feature set and they can be combined multiple times, thus resulting in doubles in the constructed feature set. What is more, old features can be combined with new features in which they already participated, therefore constructing redundant features (*e.g.*, f_2 and $f_1 \wedge f_2 \wedge f_3$ resulting in $f_2 \wedge f_1 \wedge f_2 \wedge f_3$). Another limitation is controlling the number of constructed features. The algorithm stops when a maximum number of features is constructed. This is very inconvenient, as the dimension of the new feature set cannot be known in advance and is highly dependent on the dataset. Furthermore, constructing too many features leads to overfitting and an overly complex feature set.

These shortcomings could be corrected by refining the constructing operator and by introducing a filter operator.

4.4 uFC - a greedy heuristic

We address the limitations of **uFRINGE** by proposing a second, innovative algorithm, called **uFC** (Unsupervised Feature Construction). We propose an iterative approach that reduces the overall correlation of features of a dataset by iteratively replacing pairs of highly correlated features with conjunctions of literals. We use a greedy search strategy to identify the features that are highly correlated, then we use a construction operator to create new features. From two correlated features f_i and f_j we create three new features: $f_i \wedge f_j$, $f_i \wedge \overline{f_j}$ and $\overline{f_i} \wedge f_j$. In the end, both f_i and f_j are removed from the feature set. The algorithm stops when no more new features are created or when it has performed a maximum number of iterations. The formalization and the different key parts of the algorithm (*e.g.*, the search strategy, construction operators or feature pruning) are presented in the next sections.

Figure 4.3 illustrates visually, using Venn diagrams, how the algorithm replaces the old

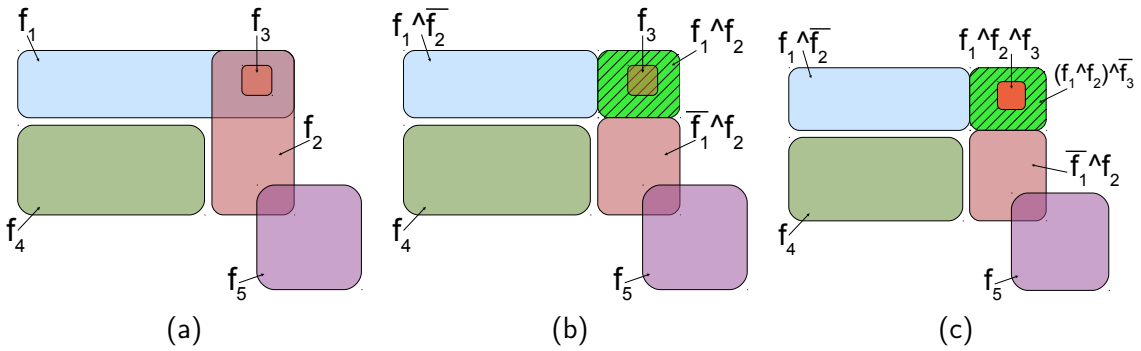


Figure 4.3 – Graphical representation of how new features are constructed - Venn diagrams. (a) Iter. 0: Initial features (Primitives), (b) Iter. 1: Combining f_1 and f_2 and (c) Iter. 2: Combining $f_1 \wedge f_2$ and f_3.

features with new ones. Features are represented as rectangles, where the rectangle for each feature contains the individuals having that feature set to **true**. Naturally, the individuals in the intersection of two rectangles have both features set to **true**. Figure 4.3a, shows the configuration of the original feature set. f_1 and f_2 have a big intersection, showing that they co-occur frequently. On the contrary, f_2 and f_5 have a small intersection, suggesting that their co-occurrence is less than that of the hazard (negatively correlated). f_3 is included in the intersection of f_1 and f_2, while f_4 has no common elements with any other. f_4 is incompatible with all of the others. The purpose of the algorithm is to construct a new feature set, in which there are no intersections between the corresponding Venn diagrams.

In the first iteration (Figure 4.3b), f_1 and f_2 are combined and 3 features are created: $f_1 \wedge f_2$, $f_1 \wedge \overline{f_2}$ and $\overline{f_1} \wedge f_2$. These new features will replace f_1 and f_2, the original ones. At the second iteration (Figure 4.3c), $f_1 \wedge f_2$ is combined with f_3. As f_3 is contained in $f_1 \wedge f_2$, the feature $\overline{f_1 \wedge f_2} \wedge f_3$ will have a support equal to zero and will be removed. Note that f_2 and f_5 are never combined, as they are considered uncorrelated. The final feature set will be $\{f_1 \wedge \overline{f_2}, f_1 \wedge f_2 \wedge f_3, f_1 \wedge f_2 \wedge \overline{f_3}, \overline{f_1} \wedge f_2, f_4, f_5\}$

4.4.1 uFC - the proposed algorithm

We define the set $P = \{p_1, p_2, ..., p_k\}$ of k user-supplied initial boolean features and $I = \{i_1, i_2, ..., i_n\}$ the dataset described using P. We start from the hypothesis that even if the primitive set P cannot adequately describe the dataset I (because of the correlations in the feature set), there exists a data-specific feature set $F = \{f_1, f_2, ..., f_m\}$ that can be created in order to represent the data better (meaning that the total correlation between features is reduced). New features are created iteratively, using conjunctions of primitive features or their negations (as seen in Figure 4.3). Our algorithm does not use the output of a learning algorithm in order to create the new features. Instead we use a greedy search strategy and a feature set evaluation function to determine if a newly-obtained feature set is more appropriate than the former one.

The schema of our proposal is presented in Algorithm 3. The feature construction is performed starting from the dataset I and the primitives P. The algorithm follows the

Algorithm 3 uFC - Unsupervised feature construction.

Input: P – set of primitive user-given features
Input: I – the data expressed using P which will be used to construct features
Inner parameters: λ – correlation threshold for searching, $limit\_iter$ – max no of iterations.
Output: F – set of newly-constructed features.

$F_0 \leftarrow P$
$iter \leftarrow 0$
repeat
  $iter \leftarrow iter + 1$
  $O \leftarrow$ **search_correlated_pairs**$(I_{iter}, F_{iter-1}, \lambda)$
  $F_{iter} \leftarrow F_{iter-1}$
  while $O \neq \emptyset$ **do**
    $pair \leftarrow$ **highest_scoring_pair**(O)
    $F_{iter} \leftarrow F_{iter} \bigcup$ **construct_new_feat**$(pair)$
    remove_candidate$(O, pair)$
  prune_obsolete_features(F_{iter}, I_{iter})
  $I_{iter+1} \leftarrow$ **convert**(I_{iter}, F_{iter})
until $F_{iter} = F_{iter-1}$ **OR** $iter = limit\_iter$
$F \leftarrow F_{iter}$

general inductive schema presented in Algorithm 2. At each iteration, **uFC** searches for frequently co-occurring pairs in the feature set created at the previous iteration (F_{iter-1}). It determines the candidate set O and then creates new features as conjunctions of the highest scoring pairs. The new features are added to the current set (F_{iter}), after which the set is filtered in order to remove obsolete features. At the end of each iteration, the dataset I is translated to reflect the feature set F_{iter}. A new iteration is performed as long as new features were generated in the current iteration and a maximum number of iterations have not yet been reached ($limit\_iter$ is a parameter for the algorithm).

Temporal complexity In order to calculate the complexity of the algorithm presented in Algorithm 3, we consider that vector operations are indivisible and executed in $O(1)$, which is the case in modern vectorial mathematical environments (*e.g.*, Octave). Vector operations (*e.g.*, sum of two vectors, element-wise multiplication) are performed in almost constant time, due to the optimization in memory access and the parallelization of computing. Consequently, we consider that calculating the correlation between a pair of features has a complexity of $O(1)$. Therefore, the search of correlated pairs has a complexity of $|O| * O(1) = O(|O|)$. The maximum size of the set O is

$$|O| \leq \frac{|F_{iter-1}| \times (|F_{iter-1}| - 1)}{2} \leq |F_{iter-1}|^2 \tag{4.1}$$

where $|F_{iter-1}|$ is the size of the created feature set at the previous iteration. In order to calculate the maximum value of $|F_{iter-1}|$, we consider that, at maximum, 3 features are

constructed based on a single pair. Consequently, the following is true:

$$|F_{iter}| \leq \frac{3}{2} \times |F_{iter-1}| \leq \left(\frac{3}{2}\right)^2 \times |F_{iter-2}| \leq ... \leq \left(\frac{3}{2}\right)^{iter} \times |F_0| = \left(\frac{3}{2}\right)^{iter} \times k \quad (4.2)$$

Knowing that $iter \leq max_{iter}$, we deduce from Equations 4.1 and 4.2 that the maximum size of the set O is:

$$|O| \leq |F_{iter-1}|^2 \leq \left(\frac{3}{2}\right)^{2\times iter} \times k^2 \leq \left(\frac{3}{2}\right)^{2\times limit_{iter}} \times k^2 \quad (4.3)$$

As the rest of the operation in the Algorithm 3 are executed in $O(1)$ or $O(k)$, and based on the size of the set O calculated in Equation 4.3, the temporal complexity of **uFC** is

$$limit_{iter} \times O\left(\left(\frac{3}{2}\right)^{2\times limit_{iter}} \times k^2\right)$$

Considering that $limit_{iter}$ is a constant, we obtain the final complexity of **uFC**, which is $O(k^2)$, therefore, quadratic with the initial size of the feature set.

In practice, if a pair (f_i, f_j) is chosen, the **prune_obsolete_features** function removes any other pairs which contain f_i or f_j, which greatly reduces the number of considered pairs and the execution time of the algorithm.

4.4.2 Searching co-occurring pairs

The **search_correlated_pairs** function searches for frequently co-occurring pairs of features in a feature set F. We start with an empty set $O \leftarrow \emptyset$ and we investigate all possible pairs of features $(f_i, f_j) \in F \times F$. We use a function (r) to measure the co-occurrence of a pair of features (f_i, f_j) and compare it to a threshold λ. If the value of the function is above the threshold, then their co-occurrence is considered as significant and the pair is added to O. Therefore, the set O will be

$$O = \{(f_i, f_j) \in F \times F,\, i \neq j \,|\, \mathbf{r}((f_i, f_j)) > \lambda\} \quad (4.4)$$

For the r correlation function, we choose to use the empirical **Pearson correlation coefficient**, which is a classical measure of the strength of the linear dependency between two variables. $r \in [-1, 1]$ and it is defined as the covariance of the two variables divided by the product of their standard deviations. The sign of the r function gives the direction of the correlation (inverse correlation for $r < 0$ and positive correlation for $r > 0$), while the absolute value or the square gives the strength of the correlation. A value of 0 implies that there is no linear correlation between the variables. When applied to Boolean variables, having the contingency table as shown in Table 4.1, the r function has the following formulation:

$$r((f_i, f_j)) = \frac{a \times d - b \times c}{\sqrt{(a+b) \times (a+c) \times (b+d) \times (c+d)}} \quad (4.5)$$

The λ threshold parameter will serve to fine-tune the number of selected pairs. Its impact on the behaviour of the algorithm will be studied in Section 4.6.3. A method of automatic choice of λ using statistical hypothesis testing is presented in Section 4.7.1.

Table 4.1 – Contingency table for two Boolean features.

	f_j	$\neg f_j$
f_i	a	b
$\neg f_i$	c	d

4.4.3 Constructing and pruning features

Once O is constructed, **uFC** performs a greedy search. The function **highest_scoring_pair** is iteratively used to extract from O the pair (f_i, f_j) that has the highest co-occurrence score.

The function **construct_new_feat** constructs three new features: $f_i \wedge f_j$, $\overline{f_i} \wedge f_j$ and $f_i \wedge \overline{f_j}$. f_i and f_j can be either primitives or features constructed in previous iterations. They represent, respectively, the intersection of the initial two features and the relative complements of one feature in the other. The new features are guaranteed by construction to be negatively correlated. If one of them is set to true for an individual, the other two will surely be false. At each iteration, very simple features are constructed: conjunctions of two literals. The creation of more complex and semantically rich features appears through the iterative process. In the example showed in Figure 4.3 (p.68), the feature $f_1 \wedge f_2 \wedge \overline{f_3}$ is obtained by combining f_1 and f_2 in a first iterations and then combining $f_1 \wedge f_2$ and f_3 in a second iteration.

After the construction of features, the **remove_candidate** function removes from O the pair (f_i, f_j), as well as any other pair that contains f_i or f_j. This is because each features is authorized to participate in only one combination in each iteration. The newly-generated features replace the old features. When there are no more pairs in O, **prune_obsolete_features** is used to remove from the feature set two types of features:

- **features that are false for all individuals**. These usually appear in the case of hierarchical relations. We consider that f_1 and f_2 have a hierarchical relation if all individuals that have feature f_1 true, automatically have feature f_2 true (*e.g.*, f_1 "is a type of" f_2 or f_1 "is a part of" f_2). We denote this relation with the notation $f_i \supseteq f_j$. One of the generated features (in the example $\overline{f_1} \wedge f_2$) is false for all individuals and, therefore, eliminated. In the example of *water* and *cascade*, we create only $water \wedge cascade$ and $water \wedge \neg cascade$, since there cannot exist a cascade without water. This is possible in the context of a complete labeling, in which a value of **false** means the absence of a feature and not missing data.
- **features that participated in the creation of a new feature**. Effectively, all $\{f_i | (f_i, f_j) \in O, f_j \in F\}$ are replaced by the newly-constructed features.

$$\{f_i, f_j \in F | (f_i, f_j) \in O\} \xrightarrow{\text{replaced by}} \{f_i \wedge f_j, \overline{f_i} \wedge f_j, f_i \wedge \overline{f_j}\}$$

4.5 Evaluation of a feature set

To our knowledge, there are no widely accepted measures to evaluate the overall correlation between the features of a feature set. We propose a measure inspired from the "inclusion-exclusion" principle [Feller 1950]. In set theory, this principle permits to express

the cardinality of the finite reunion of finite ensembles by considering the cardinality of those ensembles and their intersections. In the Boolean form, it is used to calculate the probability of a *clause* (disjunction of literals) as a function of its composing *terms* (conjunctions of literals). For example, given a feature set with 3 features $F_3 = \{f_1, f_2, f_3\}$, then the probability of the clause $f_1 \vee f2 \vee f3$ is calculated as

$$\begin{aligned} p(f_1 \vee f_2 \vee f_3) = & p(v_1) + p(v_2) + p(v_3) \\ & - p(v_1 \wedge v_2) - p(v_1 \wedge v_3) - p(v_2 \wedge v_3) \\ & + p(v_1 \wedge v_2 \wedge v_3) \end{aligned}$$

Generalizing, given the feature set $F = \{f_1, f_2, ..., f_m\}$, we have:

$$p(f_1 \vee f_2 \vee ... \vee f_m) = \sum_{k=1}^{m} \left((-1)^{k-1} \sum_{1 \leq i_1 < ... < i_k \leq m} p(f_{i_1} \wedge f_{i_2} \wedge ... \wedge f_{i_k}) \right)$$

which, by putting apart the first term, is equivalent to:

$$p(f_1 \vee f_2 \vee ... \vee f_m) = \sum_{i=1}^{m} p(f_i) + \sum_{k=2}^{m} \left((-1)^{k-1} \sum_{1 \leq i_1 < ... < i_k \leq m} p(f_{i_1} \wedge f_{i_2} \wedge ... \wedge f_{i_k}) \right)$$

Without loss of generality, we can consider that each individual has at least one feature set to **true**. Otherwise, we can create an artificial feature "null" that is set to **true** when all the others are **false**. Consequently, the left side of the equation is equal to 1. On the right side, the second term is the probability of intersections of the features. Knowing that $1 \leq \sum_{i=1}^{m} p(f_i) \leq m$, this probability of intersection has a value of zero when all features are incompatible (no overlapping). It has a "worst case scenario" value of $m - 1$, when all individuals have all the features set to **true**.

Based on these observations, we propose the **Overlapping Index** evaluation measure:

$$OI(F) = \frac{\sum_{i=1}^{m} p(f_i) - 1}{m - 1}$$

where $OI(F) \in [0, 1]$ and needs to be minimized. Hence, a feature set F_1 describes a dataset better than another feature set F_2 when $OI(F_1) < OI(F_2)$.

4.5.1 Complexity of the feature set

Number of features Considering the case of the majority of machine learning datasets, where the number of primitives is inferior to the number of individuals in the dataset, reducing correlations between features comes at the expense of increasing the number of features. Consider the pair of features (f_i, f_j) judged correlated. Unless $f_i \supseteq f_j$ or $f_i \subseteq f_j$, the algorithm will replace $\{f_i, f_j\}$ by $\{f_i \wedge f_j, \overline{f_i} \wedge f_j, f_i \wedge \overline{f_j}\}$, thus increasing the total number of features. A feature set that contains too many features is no longer informative, nor comprehensible.

The function $unique(I)$ counts how many unique individuals exist in the dataset (I is the dataset). Two individuals are considered different if and only if their description vectors

are not equal. Therefore, $unique(I) \leq |I|$, as some individuals in the dataset might not be unique. Considering F as the constructed feature set and that **uFC** searches for correlated pairs, when $|F| = unique(I)$ there is a constructed features for each unique individual in the dataset. Consequently, the maximum number of features that can be constructed by **uFC** is mechanically limited by the number of unique individuals.

$$|F| \leq unique(I) \leq |I|$$

To measure the complexity in terms of number of features, we use:

$$C_0(F) = \frac{|F| - |P|}{unique(I) - |P|}$$

where P is the primitive feature set. C_0 measures the ratio between how many extra features are constructed and the maximum number of features that can be constructed. $0 \leq C_0 \leq 1$ and needs to be minimized (a value closer to 0 means a feature set with a lower complexity).

C_0 is guarantied not to be negative only for the **uFC** algorithm. **uFRINGE** does not have any filtering mechanism and, as the results in Section 4.6.1 show, the same pairs get combined and the total number of constructed features explodes. Furthermore, C_0 can be used only for datasets for which the number of primitive attributes is lower than the number of individuals. In certain applications, this assumption is not true, *e.g.*, in text mining, a typical dataset might have several hundreds documents (individuals) and several thousands words in the dictionary serving as attributes (we present the textual numeric representation in Section 6.2.2, p. 129). For these cases, we propose in the next paragraph another indicator, based on the length of constructed features.

The average length of features At each iteration, simple conjunctions of two literals are constructed. Complex Boolean formulas are created by combining features constructed in previous iterations. Long and complicated expressions generate incomprehensible features, which are more likely a random side-effect rather than a product of underlying semantics.

We define C_1 as the average number of literals (a primitive or its negation) that appear in a Boolean formula representing a new feature.

$$\overline{P} = \{\overline{p_i} | p_i \in P\};\ \mathcal{L} = P \bigcup \overline{P}$$

$$C_1(F) = \frac{\sum_{f_i \in F} |\{l_j \in \mathcal{L} \,|\, l_j \text{ is a literal in } f_i\}|}{|F|}$$

where P is the primitive set and $1 \leq C_1 < \infty$, it does not have a superior bound and needs to be minimized.

As more iterations are performed, the feature set contains more features (C_0 grows) which are increasingly more complex (C_1 grows). This suggests a correlation between the two. What is more, since C_1 can potentially double at each iteration and C_0 can have at most a linear increase, the correlation is exponential. This correlation is further studies in Section 4.6.4. For this reason and the fact that C_1 does not have a superior bound, in the following sections we choose to use only C_0 as the complexity measure.

Overfitting All the machine learning algorithms risk to overfit the solution to the learning set. There are two ways in which **uFC** can overfit the resulted feature set, corresponding to the two complexity measures above: (a) constructing too many features (measure C_0) and (b) constructing features that are too long (measure C_1). The worst overfitting of type (a) is when the algorithm constructs as many features as the maximum theoretical number (one for each individual in the dataset). The worst overfitting of type (b) appears in the same conditions, where each constructed feature is a conjunction of all the primitives appearing for the corresponding individual. The two complexity measures can be used to quantify the two types of overfitting. Since C_0 and C_1 are correlated, both types of overfitting appear simultaneously and can be considered as two sides of a single phenomenon.

4.5.2 The trade-off between two opposing criteria

C_0 is a measure of how overfitted is a feature set. In order to avoid overfitting, the feature set complexity should be kept at low values, while the algorithm optimizes the co-occurrence score of the feature set (measure using the OI measure). Optimizing both the correlation score and the complexity at the same time is not possible, as they are opposing criteria. A compromise between the two must be achieved. This is equivalent to the optimization of two contrary criteria, which is a very well-known problem in multi-objective optimization. To acquire a trade-off between the two mutually contradicting objectives, we use the concept of **Pareto optimality** [Sawaragi *et al.* 1985], originally developed in economics. Given multiple feature sets, a set is considered to be Pareto optimal if there is no other set that has both a better correlation score and a better complexity for a given dataset. Pareto optimal feature sets will form the Pareto front. This means that no single optimum can be constructed, but rather a class of optima, depending on the ratio between the two criteria.

We plot the solutions in the plane defined by the co-occurrence score, as the horizontal axis and the complexity, as vertical axis, as shown in Figure 4.4. Constructing the Pareto front in this plane makes a visual evaluation of several characteristics of the **uFC** algorithm possible, based on the deviation of solutions compared to the front. The distance between the different solutions and the constructed Pareto front visually shows how stable the algorithm is. The convergence of the algorithm can be visually evaluated by how fast (in number of performed iterations) the algorithm transits the plane from the region of solutions with low complexity and high co-occurrence score to solutions with high complexity and low co-occurrence. We can visually evaluate overfitting, which corresponds to the region of the plane with high complexity and low co-occurrence score. Solutions found in this region are considered to be overfitted.

In order to avoid overfitting, we propose the **"closest-point"** heuristic for finding a trade-off between OI and C_0. We choose to give the two criteria equal importance. We consider as a good compromise, the solution in which the gain in co-occurrence score and the loss in complexity are fairly equal. If one of the indicators has a value considerably larger than the other, the solution is considered to be unsuitable. Such solutions would have either a high correlation between features or a high complexity. Therefore, we perform a battery of tests and we search *a posteriori* the Pareto front for solutions for which the two indicators have essentially equal values. In the space of solutions, this translates into a minimal Euclidian distance between the solution and the ideal point (the point $(0; 0)$).

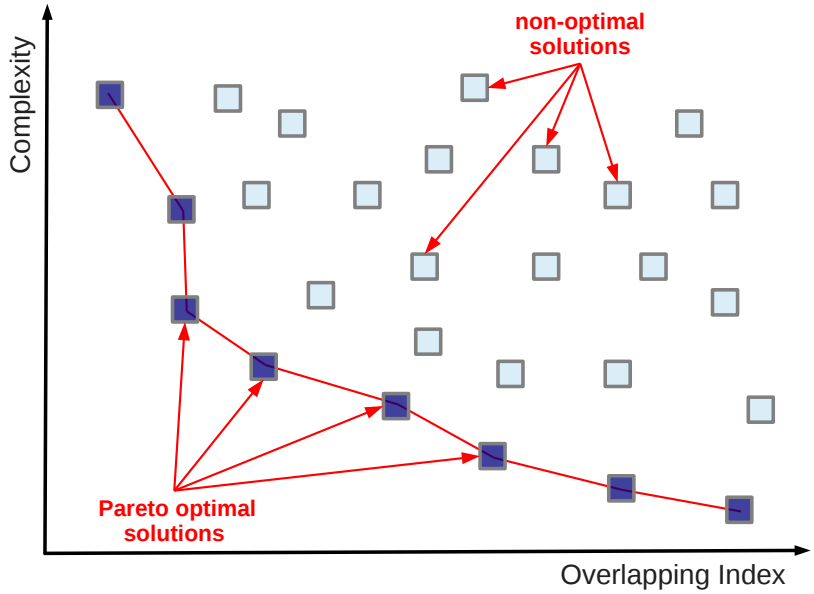
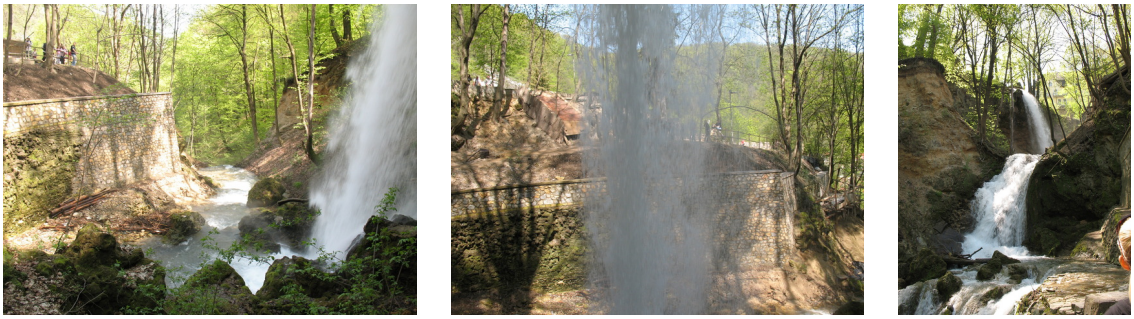


Figure 4.4 – Example of the distribution of solutions in the (Co-occurrence, Complexity) space and the Pareto optimal solutions.

Figure 4.5 – Images related to the newly-constructed feature $sky \wedge \overline{building} \wedge panorama$ on `hungarian`: (a) Hungarian puszta, (b)(c) Hungarian Mátra mountains.

4.6 Initial Experiments

Throughout the experiments, **uFC** was executed by varying only the two parameters: λ (defined in Equation 4.4) and $limit_{iter}$ (defined in Algorithm 3). We denote an execution with specific values for parameters as **uFC**(λ, $limit_{iter}$), whereas the execution where the parameters were determined *a posteriori* using the "closest-point" strategy will be noted **uFC***(λ, $limit_{iter}$). For **uFRINGE**, the maximum number of features was set at 300. We perform a comparative evaluation of the two algorithms seen from a qualitative and quantitative point of view, together with examples of typical executions. Finally, we study the impact of the two parameters of **uFC**.



(a) (b) (c)

Figure 4.6 – Images related to the newly-constructed feature $water \wedge cascade \wedge tree \wedge forest$ on `hungarian`.

Datasets Experiments were performed on three Boolean datasets. The `hungarian` dataset [1] is a real-life collection of images, depicting Hungarian urban and countryside settings. Images were manually tagged using one or more of the 13 tags. Each tag represents an object that appears in the image (eg. tree, cascade etc.). The tags serve as features and a feature takes the value **true** if the corresponding object is present in the image or **false** otherwise. The resulted dataset contains 264 individuals, described by 13 Boolean features. Notice that the work presented in this chapter deals only with reconstructing the description space and it does not deal with images (which is the object of Chapter 5). Therefore, the `hungarian` dataset contains only the labels associated with the images. The images themselves are used only for illustrative purposes in the qualitative evaluation. The `street` dataset [2] was constructed in a similar way, starting from images taken from the LabelMe dataset [Russell *et al.* 2008]. 608 urban images from Barcelona, Madrid and Boston were selected. Image labels were transformed into tags depicting objects by using the uniformization list provided with the toolbox. The dataset contains 608 individuals, described by 66 Boolean features.

The third dataset is "Spect Heart" [3] from the UCI. The dataset describes cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal (the "class" attribute). The database of 267 SPECT image sets (patients) was processed to extract 22 binary feature patterns. The original corpus is divided into a learning corpus and a testing one. We eliminated the class attribute and concatenated the learning and testing corpus into a single dataset. Unlike the first two datasets, the features of `spect` have no specific meaning, being called "F1", "F2", ... , "F22".

4.6.1 uFC and uFRINGE: Qualitative evaluation

For the human reader, it is quite obvious why *water* and *cascade* have the tendency to appear together or why *road* and *interior* have the tendency to appear separately. One

1. http://eric.univ-lyon2.fr/~arizoiu/files/hungarian.txt
2. http://eric.univ-lyon2.fr/~arizoiu/files/street.txt
3. http://archive.ics.uci.edu/ml/datasets/SPECT+Heart

(a) (b) (c)

Figure 4.7 – Images related to the newly-constructed feature $headlight \wedge windshield \wedge \overline{arm} \wedge head$ on `street`.

would expect, that based on a given dataset, the algorithms would succeed in making these associations and catching the underlying semantics. Table 4.2 shows the features constructed with **uFRINGE** and **uFC***$(0.194, 2)$ on `hungarian`. A quick overview shows that constructed features manage to make associations that seem "logical" to a human reader. For example, one would expect the feature $sky \wedge \overline{building} \wedge panorama$ to denote images where there is a panoramic view and the sky, but no buildings, therefore suggesting images outside the city. Figure 4.5 supports this expectation. Similarly, the feature $\overline{sky} \wedge building \wedge groups \wedge road$ covers urban images, where groups of people are present and $water \wedge cascade \wedge tree \wedge forest$ denotes a cascade in the forest (Figure 4.6).

Comprehension quickly deteriorates when the constructed feature set is overfitted, when the constructed features are too complex. The execution of **uFC**$(0.184, 5)$ reveals features like:

$$\overline{sky \wedge building} \wedge tree \wedge building \wedge \overline{forest} \wedge \overline{\overline{sky} \wedge building \wedge groups \wedge road} \wedge$$
$$\overline{sky \wedge \overline{building}} \wedge \overline{panorama} \wedge \overline{\overline{groups} \wedge road \wedge person} \wedge \overline{sky} \wedge \overline{groups \wedge road}$$

Even if the formula is not in the Disjunctive Normal Form (DNF), it is obvious that it is too complex to make any sense. If **uFC** tends to construct overly complex features, **uFRINGE** suffers from another type of dimensionality curse. Even if the complexity of features does not impede comprehension, the fact that there are over 300 features constructed from 13 primitives makes the newly-constructed feature set unusable. The number of features is actually greater than the number of individuals in the dataset, which proves that some of the features are redundant. The actual correlation score of the newly-created feature set is even greater than the initial primitive set. What is more, new features present redundancy, just as predicted in Section 4.3. For example, the feature $water \wedge forest \wedge grass \wedge water \wedge person$ which contains two times the primitive *water*.

The same conclusions are drawn from execution on the `street` dataset. **uFC***$(0.322, 2)$ creates comprehensible features. For example $headlight \wedge windshield \wedge \overline{arm} \wedge head$ (Figure 4.7) suggests images in which the front part of cars appears. It is especially interesting how the algorithm specifies $\overline{arm}$ in conjunction with *head* in order to differenciate between people $(head \wedge arm)$ and objects that have heads (but no arms).

Table 4.2 – Feature sets constructed by **uFC** and **uFRINGE**.

primitives	**uFRINGE**	**uFC(0.194, 2)**
person	$water \wedge forest \wedge grass \wedge water \wedge person$	$groups \wedge \overline{road} \wedge interior$
groups	$\overline{panorama} \wedge \overline{building} \wedge forest \wedge grass$	$groups \wedge \overline{road} \wedge \overline{interior}$
water	$tree \wedge person \wedge grass$	$\overline{groups \wedge \overline{road}} \wedge interior$
cascade	$tree \wedge person \wedge \overline{grass}$	$water \wedge cascade \wedge tree \wedge forest$
sky	$\overline{groups} \wedge tree \wedge person$	$\overline{water \wedge cascade} \wedge \overline{tree \wedge forest}$
tree	$\overline{person} \wedge interior$	$\overline{water \wedge cascade} \wedge tree \wedge forest$
grass	$\overline{person} \wedge \overline{interior}$	$sky \wedge building \wedge tree \wedge \overline{forest}$
forest	$water \wedge \overline{panorama} \wedge grass \wedge \overline{groups} \wedge tree$	$sky \wedge building \wedge \overline{tree \wedge \overline{forest}}$
statue	$water \wedge \overline{panorama} \wedge \overline{grass \wedge \overline{groups}} \wedge tree$	$\overline{sky \wedge building} \wedge tree \wedge forest$
building	$\overline{\overline{statue} \wedge \overline{groups}} \wedge groups$	$sky \wedge \overline{building} \wedge panorama$
road	$\overline{\overline{statue} \wedge \overline{groups}} \wedge \overline{groups}$	$sky \wedge \overline{building} \wedge \overline{panorama}$
interior	$\overline{panorama} \wedge \overline{statue} \wedge \overline{groups}$	$\overline{sky \wedge \overline{building}} \wedge panorama$
panorama	$\overline{grass} \wedge \overline{water} \wedge \overline{forest} \wedge sky$	$\overline{groups} \wedge road \wedge person$
	$\overline{grass} \wedge \overline{water} \wedge \overline{forest} \wedge \overline{sky}$	$\overline{groups} \wedge road \wedge \overline{person}$
	$person \wedge \overline{grass} \wedge \overline{water} \wedge \overline{forest}$	$\overline{groups} \wedge \overline{road} \wedge person$
	$groups \wedge \overline{sky} \wedge \overline{grass} \wedge \overline{building}$	$\overline{sky} \wedge building \wedge groups \wedge road$
	$groups \wedge \overline{sky} \wedge \overline{\overline{grass} \wedge \overline{building}}$	$\overline{sky} \wedge building \wedge \overline{groups \wedge road}$
	$\overline{person} \wedge \overline{water} \wedge \overline{forest} \wedge \overline{statue} \wedge \overline{groups}$	$\overline{\overline{sky} \wedge building} \wedge groups \wedge road$
	$\overline{person} \wedge \overline{water} \wedge \overline{forest} \wedge \overline{statue} \wedge \overline{groups}$	$water \wedge \overline{cascade}$
	$\overline{grass} \wedge \overline{person} \wedge \overline{statue}$	$\overline{tree} \wedge forest$
	$\overline{grass} \wedge \overline{\overline{person} \wedge \overline{statue}}$	*grass*
	`... and 284 others`	*statue*

Table 4.3 – Values of indicators for multiple runs on each dataset.

	Strategy	$\#feat$	*length*	*OI*	C_0
hungar.	**Primitives**	13	1.00	0.24	0.00
	uFC*(0.194, 2)	21	2.95	0.08	0.07
	uFC(0.184, 5)	36	11.19	0.03	0.20
	uFRINGE	306	3.10	0.24	2.53
street	**Primitives**	66	1.00	0.12	0.00
	uFC*(0.446, 3)	81	2.14	0.06	0.04
	uFC(0.180, 5)	205	18.05	0.02	0.35
	uFRINGE	233	2.08	0.20	0.42
spect	**Primitives**	22	1.00	0.28	0.00
	uFC*(0.432, 3)	36	2.83	0.09	0.07
	uFC(0.218, 4)	62	8.81	0.03	0.20
	uFRINGE	307	2.90	0.25	1.45

4.6.2 uFC and uFRINGE: Quantitative evaluation

Table 4.3 shows, for the three datasets, the values of certain indicators, like the size of the feature set, the average length of a feature C_1, the OI and C_0 indicators. For each dataset, we compare four feature sets: the initial feature set (primitives), the execution of **uFC*** (parameters determined by the "closest-point" heuristic), **uFC** with a set of parameters that generate an overfitted solution and **uFRINGE**. For the `hungarian` and `street` datasets, the same parameter combinations are used as in the qualitative evaluation.

On all three datasets, **uFC*** creates feature sets that are less correlated than the primitive sets, while the increase in complexity is only marginal. Very few (2-3) iterations are needed, as **uFC** converges very fast. Increasing the number of iterations has very little impact on OI, but results in very complex vocabularies (large C_0 and C_1). In the feature set created by **uFC**$(0.180, 5)$ on `street`, on average, each feature contains more than 18 literals. This is obviously too much for human comprehension.

For **uFRINGE**, the OI indicator shows very marginal or no improvement on `spect` and `hungarian` datasets, and even a degradation on `street` (compared to the primitive set). Features constructed using this approach have an average length between 2.08 and 3.1 literals, just as much as the selected **uFC*** configuration. But, it constructs between 2.6 and 13.9 times more features than **uFC***. We consider this to be due to the lack of filtering in **uFRINGE**, which would also explain the low OI score. Old features remain in the feature set and amplify the total correlation by adding the correlation between old and new features.

Some conclusions Similar conclusions can be drawn from the quantitative evaluations as from the qualitative evaluation. Both **uFC** and **uFRINGE** succeed in capturing the semantic links between the primitive features, by constructing comprehensible boolean formulas. Whatsoever, each suffers from a dimensionality problem: **uFRINGE** constructs too many new features, while **uFC** constructs features that are too complex. Furthermore, **uFRINGE** fails in reducing the total correlation between the newly constructed features. For **uFC**, it is crucial to control the complexity of the created features and the proposed heuristic achieves this task. Only marginal complexity increases yield high correlation decreases.

4.6.3 Impact of parameters λ and $limit_{iter}$

In order to understand the impact of parameters, we executed **uFC** with a wide range of values for λ and $limit_{iter}$ and studied the evolution of the indicators OI and C_0. For each dataset, we varied λ between 0.002 and 0.5 with a step of 0.002. For each value of λ, we executed **uFC** by varying $limit_{iter}$ between 1 and 30 for the `hungarian` dataset, and between 1 and 20 for `street` and `spect` (for execution time considerations, given that `street` and `spect` are larger datasets than `hungarian`). We study the evolution of the indicators as a function of $limit_{iter}$, respectively λ, we plot the solution in the (OI, C_0) space and construct the Pareto front.

For the study of $limit_{iter}$, we hold λ fixed at various values and we make vary only $limit_{iter}$. The evolution of the OI correlation indicator is given in Figure 4.8a. As expected,

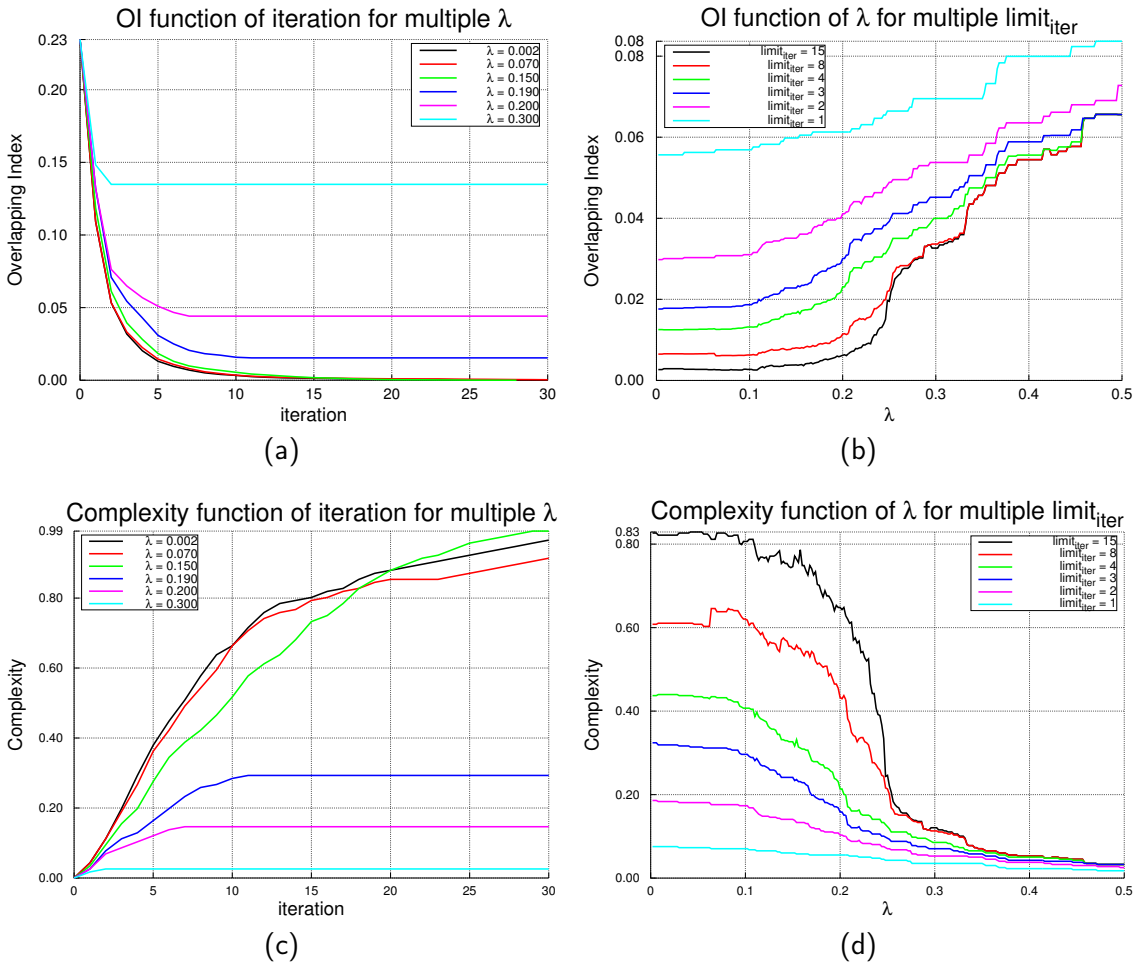


Figure 4.8 – Variation of indicators OI (top row) and C_0 (bottom row) with $limit_{iter}$ on `hungarian` (left column) and with λ on `street` (right column).

the measure ameliorates with the number of iterations. OI has a very rapid descent and needs less than 10 iterations to converge on all datasets towards a value dependent on λ. The higher the value of λ, the higher the value of convergence. The complexity has a very similar evolution, shown in Figure 4.8c), but in the inverse direction: it increases with the number of iterations performed. It also converges towards a value that is dependent on λ: the higher the value of λ, the lower the complexity of the resulting feature set.

Similarly, we study λ by fixing $limit_{iter}$. Figure 4.8b shows how OI evolves when varying λ. As foreseen, for all values of $limit_{iter}$, the OI indicator increases with λ, while C_0 decreases with λ. OI shows an abrupt increase between 0.2 and 0.3, for all datasets. For lower values of λ, many pairs get combined as their correlation score is bigger than the threshold. As λ increases, only highly correlated pairs get selected and this usually happens in the first iterations. Performing more iterations does not bring any change and indicators are less dependent on $limit_{iter}$. For `hungarian`, no pair has a correlation score higher than 0.4. Setting λ higher than this value causes **uFC** to output the primitive set (no features are created). In Figure 4.8d, the evolution of the complexity is, as in the previous case of

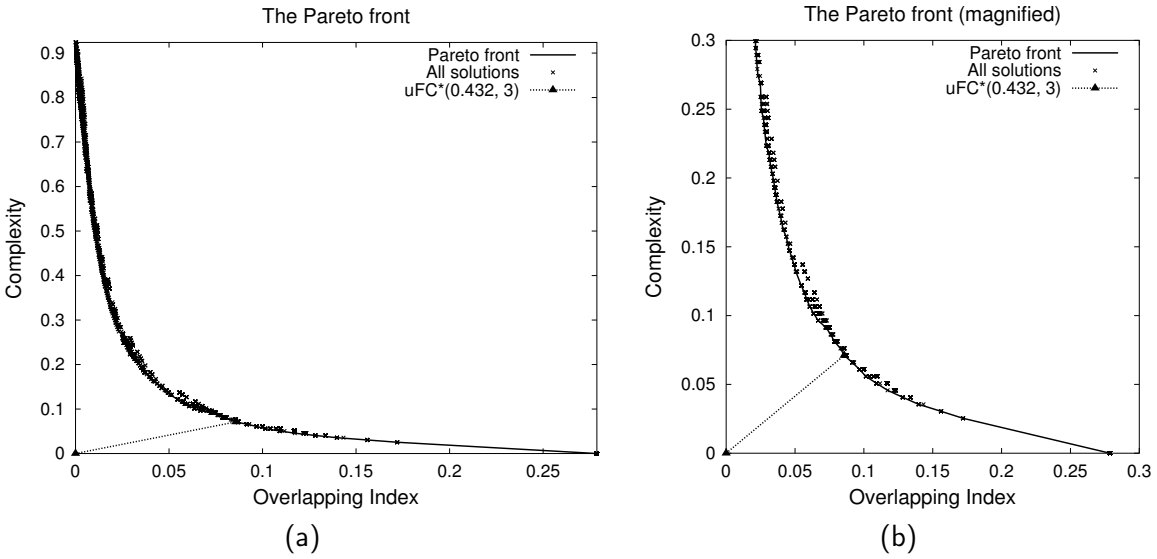


Figure 4.9 – The distribution of solutions, the Pareto front and the closest-point on `spect` dataset.

$limit_{iter}$ very similar to the one of OI, but in the inverse direction. The complexity of the constructed feature set descends with the increase of λ.

Pareto optimality To study Pareto optimality, we plot the generated solutions in the (OI, C_0) space. Figure 4.9a presents the distribution of solutions, the Pareto front and the solution chosen by the "closest-point" heuristic. The solutions generated by **uFC** with a wide range of parameter values are not dispersed in the solution space, but their distribution is rather close together. This shows good algorithm stability. Even if not all the solutions are Pareto optimal, none of them are too distant from the front and there are no outliers.

Most of the solutions densely populate the part of the curve corresponding to low OI and high C_0. As pointed out in the Section 4.5.2, the area of the front corresponding to high feature set complexity (high C_0) represents the overfitting area. This confirms that the algorithm converges fast, then enters overfitting. Most of the improvement in quality is done in the first 2-3 iterations, while further iterating improves quality only marginally with the cost of an explosion of complexity. The "closest-point" heuristic keeps the constructing out of overfitting, by stopping the algorithm at the point where the gain of co-occurence score and the loss in complexity are fairly equal. Figure 4.9b zooms to the region of the solution space corresponding for low numbers of iterations and both axis have equal scales.

4.6.4 Relation between number of features and feature length

Both the average length of a feature (C_1) and the number of features (C_0) increase with the number of iterations. In Section 4.5.1 we have speculated that the two are correlated: $C_1 = f(C_0)$. For each λ in the batch of tests, we create the C_0 and C_1 series depending on the $limit_{iter}$. Figure 4.10 shows the series for $\lambda \in [0.002, 0.07, 0.15, 0.19, 0.2, 0.3]$ (C_1 is plotted on a logarithmic scale). We perform a statistical hypothesis test, using the Kendall

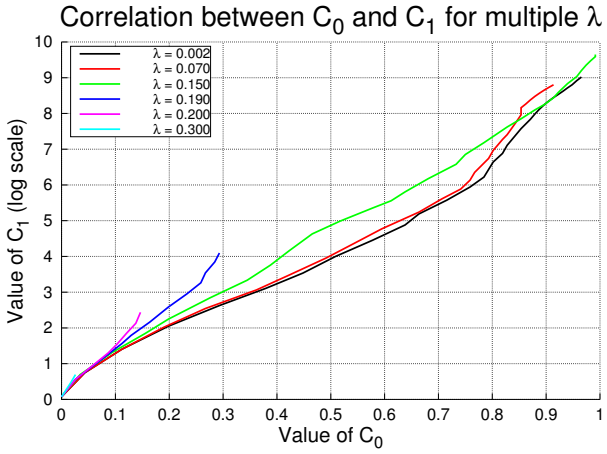


Figure 4.10 – The variation of complexities C_0 and C_1 (log scale) for multiple values of λ on `hungarian` dataset.

rank coefficient as the test statistic. The Kendall rank coefficient is particularly useful as it makes no assumptions about the distributions of C_0 and C_1. For all values of λ, for all datasets, the statistical test revealed a p-value of the order of 10^{-9}. This is consistently lower than habitually used significance levels and makes us reject the null independence hypothesis and statistically prove that C_0 and C_1 are correlated. Furthermore, the graphics in Figure 4.10 make us conclude empirically that there is an exponential correlation between C_0 and C_1: $C_1 = f(e^{C_0})$.

4.7 Improving the uFC algorithm

The major difficulty of **uFC**, shown by the initial experiments, is setting the values of parameters. An unfortunate choice would result in either an overly complex feature set or a feature set where features are still correlated. But both parameters λ and $limit_{iter}$ are dependent on the dataset and finding the suitable values would prove to be a process of trial and error for each new corpus. The "closest-point" heuristic achieves acceptable equilibrium between complexity and performance, but requires multiple executions with large choices of values for parameters and the construction of the Pareto front, which might be very costly, especially for large datasets.

We propose, in Section 4.7.1, a new method for choosing λ based on statistical hypothesis testing and, in Section 4.7.2, a new stopping criterion inspired from the "closest-point" heuristic. These are integrated into a new "risk-based" heuristic, which approximates the best solution while avoiding the time consuming construction of multiple solutions and the Pareto front. The only parameter is the significance level α, which is independent of the dataset, and which simplifies the task of running **uFC** on new, unseen datasets. A pruning technique is also proposed.

4.7.1 Automatic choice of λ

The co-occurrence threshold λ is highly dependent on the dataset (*e.g.*, on small datasets it should be set higher, while on large datasets only a small value is required to consider two features as correlated). Therefore, we propose to replace this user-supplied threshold with a technique that selects only pairs of features for whom the positive linear correlation is statistically significant. These pairs are added to the set O of co-occurring pairs (defined in Section 4.4.2) and, starting from O, new features are constructed. We use a statistical method: the *hypothesis testing*. For each pair of candidate features, we test the independence hypothesis H_0 against the positive correlation hypothesis H_1.

We use as a test statistic the Pearson correlation coefficient (calculated as defined in Section 4.4.2) and test the following formally defined hypothesis: $H_0 : \rho = 0$ and $H_1 : \rho > 0$, where ρ is the theoretical correlation coefficient between two candidate features. We can show that in the case of Boolean variables, having the contingency table shown in Table 4.1, the observed value of the χ^2 of independence is $\chi^2_{obs} = nr^2$ (n is the size of the dataset). Consequently, considering true the hypothesis H_0, nr^2 is approximately following a χ^2 distribution with one degree of freedom ($nr^2 \sim \chi^2_1$), resulting in $r\sqrt{n}$ following a standard normal distribution ($r\sqrt{n} \sim N(0,1)$), given that n is large enough.

We reject the H_0 hypothesis in favour of H_1 if and only if $r\sqrt{n} \geq u_{1-\alpha}$, where $u_{1-\alpha}$ is the right critical value for the standard normal distribution. Two features will be considered significantly correlated when $r((f_i, f_j)) \geq \frac{u_{1-\alpha}}{\sqrt{n}}$. The significance level α represents the risk of rejecting the independence hypothesis when it was in fact true. It can be interpreted as the *false discovery risk* in data mining. In our context of feature construction, α represents the *false construction risk*, since this is the risk of constructing new features based on a pair of features that are not really correlated. Statistical literature usually sets α at 0.05 or 0.01, but levels of 0.001 or even 0.0001 are often used.

The proposed method repeats the independence test a great number of times, which inflates the number of *type I errors* (a type I error is the incorrect rejection of a true null hypothesis, a false positive). [Ge *et al.* 2003] presents several methods for controlling the false discoveries. Setting aside the Bonferroni correction, often considered too simplistic and too drastic, one has the option of using sequential rejection methods [Benjamini & Liu 1999, Holm 1979], the q-value method of Storey [Storey 2002] or making use of bootstrap [Lallich *et al.* 2006]. In our case, applying these methods is not clear-cut, as tests performed at each iteration depend on the results of the tests performed at previous iterations. It is noteworthy that a trade-off must be acquired between the inflation of false discoveries and the inflation of missed discoveries. This makes us choose a risk between 5% and $\frac{5\%}{m}$, where m is the theoretical number of tests to be performed.

4.7.2 Stopping criterion. Candidate pruning technique.

In this section, we deal with the second data-dependent parameter of **uFC**: $limit_{iter}$. $limit_{iter}$ controls the number of iterations the algorithm performs and, implicitly, the complexity of the constructed features. We replace the $limit_{iter}$ parameter with a stopping criterion and propose a new data-independent heuristic. We also introduce a pruning technique, issued from the theoretical conditions necessary for the χ^2 statistical testing.

Risk-based heuristic We have introduced in Section 4.5.2 the "closest-point" for choosing the values for parameters λ and $limit_{iter}$. It searches the solution on the Pareto front for which the indicators are sensibly equal. We transform the heuristic into a stopping criterion: OI and C_0 are combined into a single formula, the **root mean square** (RMS) (also known as the quadratic mean). The algorithm iterates while the value of RMS descends and it stops iterating when RMS has reached a minimum. The RMS function has the following formula $RMS(OI, C_0) = \sqrt{\frac{OI^2+C_0^2}{2}}$ and has the tendency of having a value which is closer to the maximum between OI and C_0. This means that when OI is very high or C_0 is very high, the RMS function has a high value. Its value decreases as the difference between the two indicators decreases (as shown in the experiments, in Figure 4.11). In our case, the RMS reaches its minimum value when OI and C_0 are having equal values.

The $limit_{iter}$ parameter, which is data-dependent, is replaced by the automatic RMS stopping criterion. This stopping criterion together with the automatic λ choice strategy, presented in Section 4.7.1, form a data-independent heuristic for choosing parameters. We will call the new heuristic **risk-based heuristic**. This new heuristic make possible to approximate the data-dependent parameters (λ is approximated by $\frac{u_{1-\alpha}}{\sqrt{n}}$ and $limit_{iter}$ using the RMS function) and to avoid the time consuming task of computing a batch of solutions and constructing the Pareto front.

Pruning The theoretical condition necessary in order to apply the χ^2 independence test is that the expected (theoretical) frequencies, considering true the H_0 hypothesis, are greater or equal than 5. We add this constraint to the new feature search strategy (defined in Section 4.4.2). Pairs for whom the values of $\frac{(a+b)(a+c)}{n}$, $\frac{(a+b)(b+c)}{n}$, $\frac{(a+c)(c+d)}{n}$ and $\frac{(b+d)(c+d)}{n}$ are not greater than 5, will be filtered from the set of candidate pairs O. This stops the algorithm from constructing features that are present for very few individuals in the dataset.

While this pruning technique is inherently related to the χ^2 independence test, it can also be applied with the initial form of the **uFC** algorithm, proposed in Section 4.4. As the experiments presented in the next section show, applying the theoretical pruning successfully tackles the problem of overfitting.

4.8 Further Experiments

We test the proposed improvements, similarly to the methods used in Section 4.6, on the same three datasets: `hungarian`, `spect` and `street`. We execute **uFC** in two ways: the classical **uFC** (Section 4.4) and the improved **uFC** (Section 4.7). The classical **uFC** needs to have parameters λ and $limit_{iter}$ set (noted **uFC**(λ, $limit_{iter}$)). **uFC***(λ, $limit_{iter}$) denotes the execution with parameters which were determined *a posteriori* using the "closest-point" heuristic. The improved **uFC** will be denoted as **uFC**$_\alpha(risk)$. The "risk-based" heuristic will be used to determine the parameters and control the execution.

4.8.1 Risk-based heuristic for choosing parameters

Root Mean Square In the first batch of experiments, we study the variation of the Root Means Square aggregation function for a series of selected values of λ. We make vary

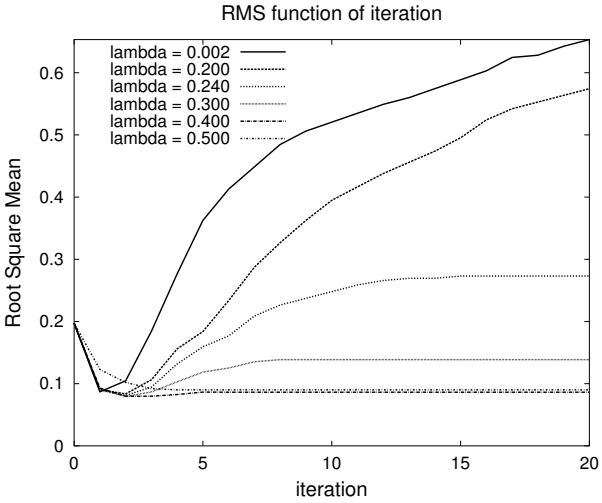


Figure 4.11 – RMS vs. $limit_{iter}$ on `spect`.

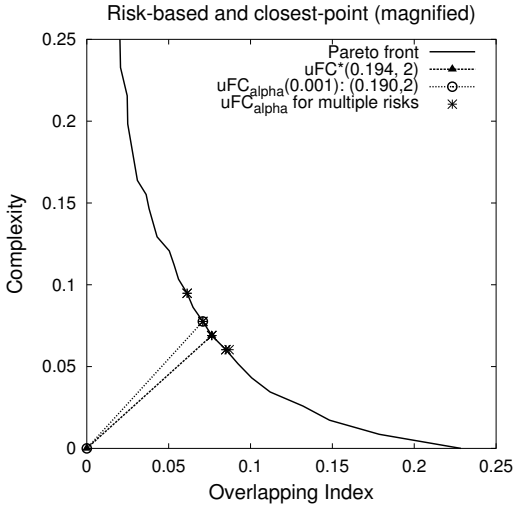


Figure 4.12 – "Closest-point" and "risk-based" for multiple α on `hungarian`.

$limit_{iter}$ between 0 and 30, for `hungarian`, and between 0 and 20 for `spect` and `street`. The evolution of RMS is presented in Figure 4.11.

For all λ, the RMS starts by decreasing, as OI descends more rapidly than the C_0 increases. In just 1 to 3 iterations, RMS reaches its minimum and afterwards its value starts to increase. This is due to the fact that complexity increases rapidly, with only marginal improvement of quality. This behaviour is consistent with the results presented in Section 4.6. As already discussed in Section 4.6.3, λ has a bounding effect over complexity, thus explaining why RMS reaches a maximum for higher values of λ.

The "risk-based" heuristic The "risk-based" heuristic approximates the data-dependent parameters λ and $limit_{iter}$ using the data-independent significance level α and the RMS stopping criterion. The second batch of experiments deals with comparing the "risk-based"

Table 4.4 – "closest-point" and "risk-based" heuristics.

	Strategy	λ	$limit_{iter}$	$\#feat$	$\#common$	$length$	OI	C_0
hung.	**Primitives**	-	-	13	-	1.00	0.235	0.000
	uFC*(0.194, 2)	0.194	2	21	19	2.95	0.076	0.069
	uFC$_\alpha$(0.001)	0.190	2	22		3.18	0.071	0.078
street	**Primitives**	-	-	66	-	1.00	0.121	0.000
	uFC*(0.446, 3)	0.446	3	87	33	2.14	0.062	0.038
	uFC$_\alpha$(0.0001)	0.150	1	90		1.84	0.060	0.060
spect	**Primitives**	-	-	22	-	1.00	0.279	0.000
	uFC*(0.432, 3)	0.432	3	36	19	2.83	0.086	0.071
	uFC$_\alpha$(0.0001)	0.228	2	39		2.97	0.078	0.086

heuristic to the "closest-point" heuristic. The "closest-point" is determined as described in Section 4.6. The "risk-based" heuristic is executed multiple times, with values for $\alpha \in \{0.05, 0.01, 0.005, 0.001, 0.0008, 0.0005, 0.0003, 0.0001, 0.00005, 0.00001\}$

Table 4.4 gives a quantitative comparison between the two heuristics. A risk of 0.001 is used for `hungarian` and 0.0001 for `spect` and `street` (because of the size of these datasets, see the discussion in Section 4.7.1 about repeating an independence test multiple time). The feature sets created by the two approaches are very similar, considering all indicators. Not only the differences between values for OI, C_0, average feature length and feature set dimension are negligible, but most of the created features are identical. On `hungarian`, 19 of the 21 features created by the two heuristics are identical. Table 4.5 shows the two features sets, with non-identical features in bold.

Figure 4.12 presents the distribution of solutions created by the "risk-based" heuristic with multiple α, plotted on the same graphics as the Pareto front in the (OI, C_0) space. Solutions for different values of risk α are grouped closely together. Not all of them are on the Pareto front, but they are never too far from the "closest-point" solution, providing a good equilibrium between quality and complexity.

Degraded performances On `street`, performances of the "risk-based" heuristic start to degrade compared to **uFC***. Table 4.4 shows differences in the resulted complexity and only 33% of the constructed features are common for the two approaches. Figure 4.13a shows that solutions found by the "risk-based" approach are moving away from the "closest-point". The cause is the large size of the `street` dataset. As the sample size increases, the null hypothesis tends to be rejected at lower levels of p-value. The auto-determined λ threshold is set too low and the constructed feature sets are too complex. Pruning solves this problem as shown in Figure 4.13b and Section 4.8.2.

4.8.2 Pruning the candidates

The pruning technique is independent of the "risk-based" heuristic and can be applied in conjunction with the classical **uFC** algorithm. An execution of this type will be denoted

Table 4.5 – Feature sets constructed by "closest-point" and "risk-based" heuristics on `hungarian`.

primitives	**uFC*(0.194, 2)**	**uFC$_\alpha$(0.001)**
$person$	$groups \wedge \overline{road} \wedge interior$	$groups \wedge \overline{road} \wedge interior$
$groups$	$groups \wedge \overline{road} \wedge \overline{interior}$	$groups \wedge \overline{road} \wedge \overline{interior}$
$water$	$\overline{groups \wedge \overline{road}} \wedge interior$	$\overline{groups \wedge \overline{road}} \wedge interior$
$cascade$	$water \wedge cascade \wedge tree \wedge forest$	$water \wedge cascade \wedge tree \wedge forest$
sky	$water \wedge cascade \wedge \overline{tree \wedge forest}$	$water \wedge cascade \wedge \overline{tree \wedge forest}$
$tree$	$\overline{water \wedge cascade} \wedge tree \wedge forest$	$\overline{water \wedge cascade} \wedge tree \wedge forest$
$grass$	$sky \wedge building \wedge tree \wedge \overline{forest}$	$sky \wedge building \wedge tree \wedge \overline{forest}$
$forest$	$sky \wedge building \wedge \overline{tree \wedge \overline{forest}}$	$sky \wedge building \wedge \overline{tree \wedge \overline{forest}}$
$statue$	$\overline{sky \wedge building} \wedge tree \wedge forest$	$\overline{sky \wedge building} \wedge tree \wedge forest$
$building$	$sky \wedge \overline{building} \wedge panorama$	$sky \wedge \overline{building} \wedge panorama$
$road$	$\overline{sky \wedge building} \wedge \overline{panorama}$	$\overline{sky \wedge building} \wedge \overline{panorama}$
$interior$	$\overline{sky \wedge \overline{building}} \wedge panorama$	$\overline{sky \wedge \overline{building}} \wedge panorama$
$panorama$	$\overline{groups} \wedge road \wedge person$	$\overline{groups} \wedge road \wedge person$
	$\overline{groups} \wedge road \wedge \overline{person}$	$\overline{groups} \wedge road \wedge \overline{person}$
	$\overline{\overline{groups} \wedge road} \wedge person$	$\overline{\overline{groups} \wedge road} \wedge person$
	$water \wedge \overline{cascade}$	$\overline{\mathbf{sky}} \wedge \mathbf{building} \wedge \mathbf{groups} \wedge \mathbf{road}$
	$\overline{\mathbf{sky}} \wedge \mathbf{building}$	$\overline{\mathbf{sky} \wedge \mathbf{building}} \wedge \mathbf{groups} \wedge \mathbf{road}$
	$\overline{tree} \wedge forest$	$\overline{\overline{\mathbf{sky}} \wedge \mathbf{building}} \wedge \mathbf{groups} \wedge \mathbf{road}$
	$\mathbf{groups} \wedge \mathbf{road}$	$water \wedge \overline{cascade}$
	$grass$	$\overline{tree} \wedge forest$
	$statue$	$grass$
		$statue$

uFC$_P(\lambda, max_{iter})$. We execute **uFC**$_P(\lambda, max_{iter})$ with the same parameters and on the same datasets as described in Section 4.6.3.

We compare **uFC** with and without pruning by plotting on the same graphic the two Pareto fronts resulted from each set of executions. Figure 4.14a shows the pruned and non-pruned Pareto fronts on `hungarian`. The graphic should be interpreted in a manner similar to a ROC curve, since the algorithm seeks to minimize OI and C_0 at the same time. When one Pareto front runs closer to the origin of the graphic $(0, 0)$ than a second, it means that the first dominates the second one and, thus, its corresponding approach yields better results. For all datasets, the pruned Pareto front dominates the non-pruned one. The difference is marginal, but proves that filtering improves results.

Some conclusions The most important conclusion is that filtering limits complexity. As the initial experiments (Figure 4.9a) showed, most of the non-pruned solutions correspond to very high complexities. Visually, the Pareto front is tangent to the vertical axis (the complexity) and showing complexities around $0.8 - 0.9$ (out of 1). On the other hand, the Pareto front corresponding to the pruned approach stops, for all datasets, for complexities lower

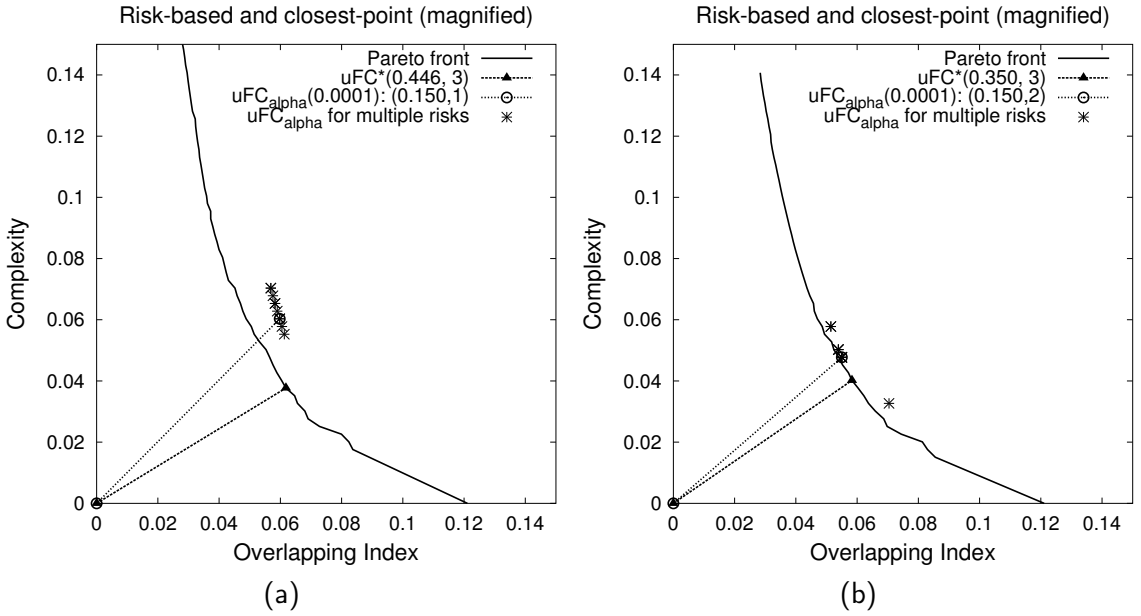


Figure 4.13 – "Closest-point" and "Risk-based" heuristics for `street` without pruning (a) and with pruning (b).

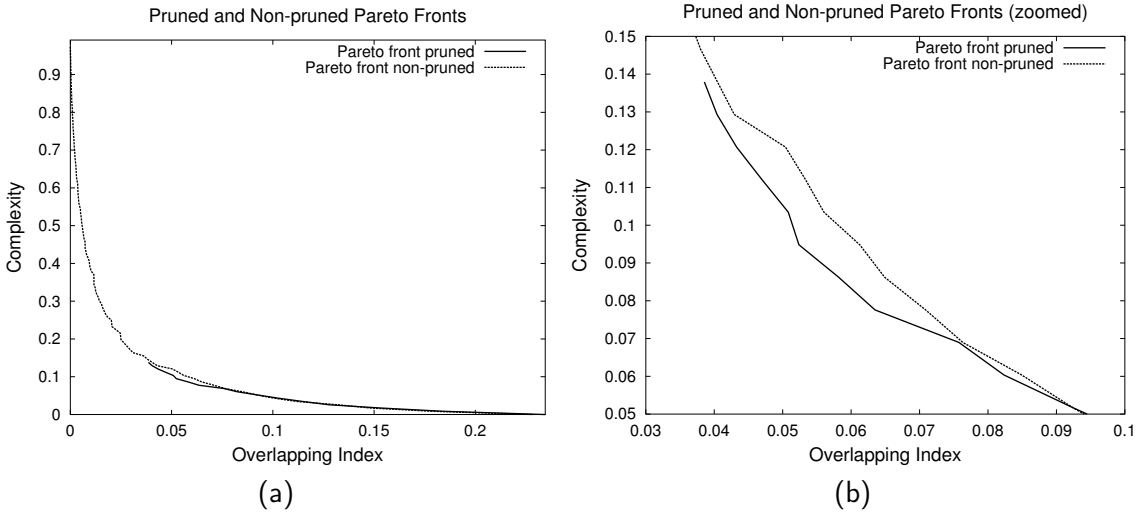


Figure 4.14 – Pruned and Non-pruned Pareto Fronts on `hungarian` (a) and a zoom to the relevant part (b).

than 0.15. This proves that filtering successfully discards solutions that are too complex to be interpretable.

Last, but not least, filtering corrects the problem of degraded performances of the "risk-based" heuristic on big datasets. We ran $\mathbf{uFC}_P$ with risk $\alpha \in \{0.05, 0.01, 0.005, 0.001, 0.0008, 0.0005, 0.0003, 0.0001, 0.00005, 0.00001\}$. Figure 4.13b presents the distributions of solutions found with the "risk-based pruned" heuristic on `street`. Unlike results without

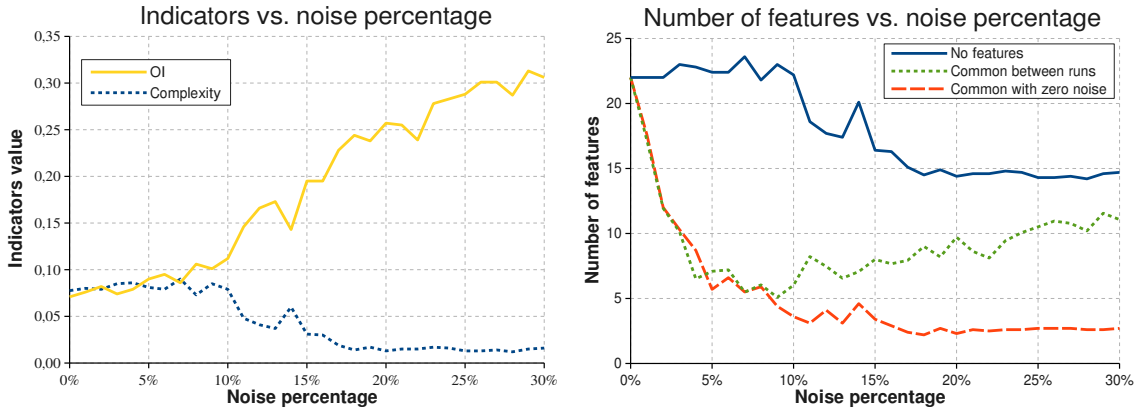


Figure 4.15 – **uFC**$_\alpha(risk)$ stability on `hungarian` when varying the noise percentage: and indicators (a) and number of constructed features (b).

pruning (Figure 4.13a), solutions generated with pruning are distributed closely to those generated by "closest-point" and to the Pareto front.

4.8.3 Algorithm stability

In order to evaluate the stability of the **uFC**$_\alpha$ algorithm, we introduce noise in the `hungarian` dataset. The percentage of noise varied between 0% (no noise) and 30%. Introducing a certain percentage x% of noise means that $x\% \times k \times n$ random features in the datasets are inverted (false becomes true and true becomes false). k is the number of primitives and n is the number of individuals. For each given noise percentage, 10 noised datasets are created and only the averages are presented. **uFC**$_\alpha$ is executed for all the noised datasets, with the same combination of parameters ($risk = 0.001$ and no filtering).

The stability is evaluated using five indicators:

- **Overlapping Index** (OI);
- **Feature set complexity** (C_0);
- **Number of features**: the total number of features constructed by the algorithm;
- **Common with zero noise**: the number of identical features between the feature sets constructed based on the noised datasets and the non-noised dataset. This indicator evaluates the measure in which the algorithm is capable of constructing the same features, even in the presence of noise;
- **Common between runs**: the average number of identical features between feature sets constructed using datasets with the same noise percentage. This indicator evaluates how much the constructed feature sets differ for a given noise level.

As the noise percentage augments, the dataset becomes more random. Less pairs of primitives are considered as correlated and therefore less new features are created. Figure 4.15a shows that the overlapping indicator increases with the noise percentage, while the complexity decreases. Furthermore, most features in the initial dataset are set to false. As the percentage of noise increases, the ratio equilibrates (more false values becoming true, than the contrary). As a consequence, for high noise percentages, the OI score is higher than

for the primitive set.

The same conclusions can be drawn from Figure 4.15b. The indicator **Number of features** descends when the noise percentage increases. This is because fewer features are constructed and the resulting feature set is very similar to the primitive set. The number of constructed features stabilizes around 20% of noise. This is the point where most of the initial correlation between features is lost. **Common with zero noise** has a similar evolution. The number of features identical to the non-noised dataset descends quickly and stabilizes around 20%. After 20%, all the identical features are among the initial primitives. Similarly, the value of **Common between runs** descends at first. For small values of introduced noise, the correlation between certain features is reduced, modifying the order in which pairs of correlated features are selected in Algorithm 3. This results in a diversity of constructed feature sets. As the noise level increases and the noised datasets become more random, the constructed feature sets resemble the primitive set, therefore augmenting the value of **Common between runs**.

Some conclusions Introducing noise in the dataset is synonym to destroying the semantic connections between the features. As the noise percentage increases, the co-occurrence of features is increasingly the result of only hazard. As **uFC** was designed to detect the relationships between features, it behaves as expected: it creates less new features or simply outputs the primitive set from a noise level onward. This can not be attributed to a special sensibility of **uFC** to noise, but simply to the fact that noise destroys the existing semantic information in the dataset, therefore making it impossible for **uFC** to detect any connections.

4.9 Usage of the multi-objective optimization techniques

Throughout this chapter, we have used techniques based on multi-objective optimization to visually evaluate the generated feature sets. We assess the distribution of the solutions in the measure space, the solutions stability, the overfitting, the convergence and the proposed heuristics. In this section, we resume the usage of these techniques into an empirical evaluation approach.

The solution are visualized in the bi-dimensional space defined by the co-occurrence score (OI) and the feature set complexity (C_0) evaluation measures. We make vary the different parameters of our approaches and we plot the obtained solutions in the (OI, C_0) plane. The two are opposing criteria: decreasing one criterion increases the other criterion. Therefore, the notion of Pareto optimality can be applied. We construct the Pareto front *a posteriori*, by simply selecting the solutions that are not dominated. We use the created Pareto front to perform several evaluations.

Quick visualization of the distribution of solutions and the stability of solutions The main usage of the Pareto front is that it allows a quick visualization of the distribution of the solutions in the (OI, C_0) space. The visualization permits an empiric evaluation of how a choice of parameters impacts the obtained dataset's co-occurrence score and complexity, compared to the Pareto-optimal solutions. Figures 4.9 (p. 81) and 4.16 present typical

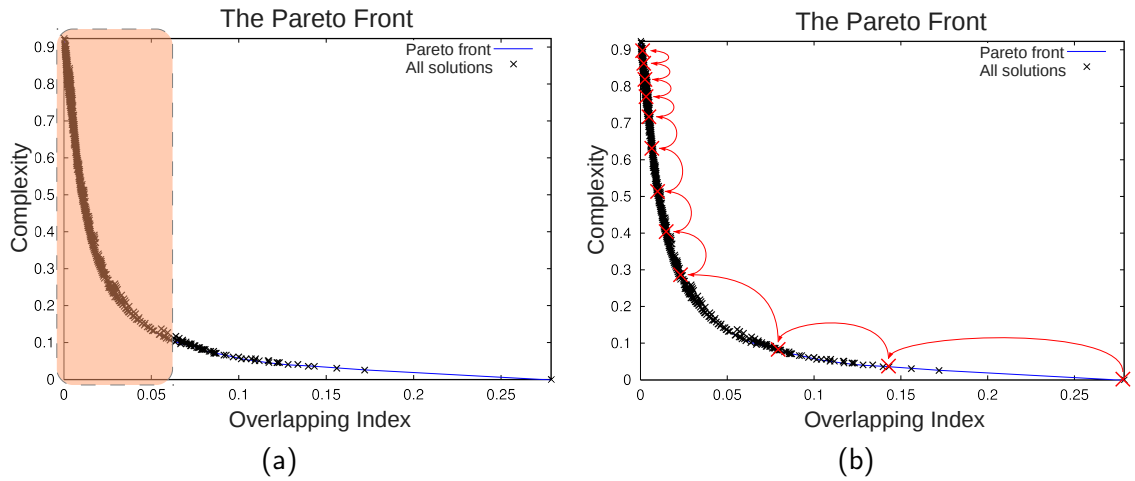


Figure 4.16 – `street` dataset: Visualizing the overfitting (a) and the speed of convergence (b).

distributions of solutions in the measures space. The distance between the different solutions and the Pareto front show how stable the constructed solutions are. The distribution graphics plot the solutions grouped closely to the front, showing little variation and good solution stability.

Visualization of the overfitting and convergence of the algorithm We have shown in Section 4.6.1 that one of the major concerns when building a new feature set is the overfitting of the new features to the data. Overfitted sets are too complex to be comprehensible by a human and they can be numerically detected due to their high complexity and low co-occurrence score. By plotting a solution in the (OI, C_0) space, we can visually assess if a solution is overfitted. Figure 4.16a depicts overfitting as the region close to the vertical axis (low OI and high C_0 score). Visibly, most of constructed solutions can be found in the overfitting region. This is due to the high convergence speed of our algorithm. The speed of convergence (in number of iterations) can be visually evaluated by how fast the algorithm transits from solutions with low complexity and high co-occurrence score (south-est region of the graphic) to solutions with high complexity and low co-occurrence (north-west region). Figure 4.16b shows the points in the (OI, C_0) space correponding to the feature sets constructed by **uFC** at each iteration. Most of the gain in the co-occurrence score is done in the first 2 or 3 iterations. Starting from this point, solutions are usually overfitted.

Limiting the overfitting and detecting abnormal solutions Given the risk of overfitting, we proposed in Section 4.5.2 the "closest-point" heuristic, which consists in choosing on the constructed Pareto Front the point where the loss in complexity and the gain in co-occurrence score are fairly equal. Apart from avoiding overfitting, the "closest-point" heuristic has the advantage of automatically choosing the values of the λ and $limit_{iter}$ parameters. However, the "closest-point" heuristic demands a full sweep of parameters values, which can be quite time consuming. In Section 4.7.1 we propose the "risk-based" heuristic, based on statistical testing. In Figure 4.12 (p. 85) we visually evaluate how the solutions

obtained using the "risk-based" heuristic are positioned in comparison with the solution obtained using the "closest-point" heuristic. We further evaluate the behavior of the "risk-based" heuristic, by plotting the obtained solutions on the previously constructed Pareto front. Therefore, we detect, in Figure 4.13a (p. 88), abnormal solutions on the `street` dataset by visualizing their deviation from the front.

Comparing two approaches We correct the abnormal solution by introducing in Section 4.7.2 a theoretical prunning technique. In Section 4.8.2, we construct the two Pareto fronts corresponding to the execution of **uFC** with and without pruning. In Figure 4.14b (p. 88), we plot the two Pareto fronts on the same graphic and we show that the one corresponding to the filtered version constantly dominates the non-filtered version. We draw the conclusion that filtering is beneficial, since it obtains better scores for both contradicting criteria at the same time. Furthermore, it prevents entering the overfitting region.

4.10 Conclusion and future work

Conclusion The work presented in this chapter tackles one of the core research problems of this thesis: leveraging semantics into data representation. More specifically, we adapt the feature set used to describe the data to the semantic relationships between features induced by the data itself. We propose two approaches towards augmenting the expressive power of the features set employed to describe a boolean dataset. Our proposals construct new features by taking into account the underlying semantics present in the dataset. Unlike the other feature construction algorithms proposed so far in the literature, our proposals work in an unsupervised learning paradigm. **uFRINGE** is an unsupervised adaptation of the FRINGE algorithm, while **uFC** is a new approach that replaces linearly correlated features with conjunctions of literals. We prove that our approaches succeed in reducing the overall correlation in the feature set, while constructing comprehensible features. We have performed extensive experiments to highlight the impact of parameters on the total correlation measure and feature set complexity. Based on the first set of experiments, we have proposed a heuristic that finds a suitable balance between quality and complexity and avoids time consuming multiple executions, followed by a Pareto front construction. We use statistical hypothesis testing and confidence levels for parameter approximation and reasoning on the Pareto front of the solutions for evaluation. We also propose a pruning technique, based on hypothesis testing, that limits the complexity of the generated features and speeds up the construction process.

Conceptual articulation with previously presented work Our interest with the representation of data initially originated in the need to reorganize a label set used for tagging images. We employ these labels later, in Chapter 5, in order to improve the semantic description of image data. This chapter's proposals concerning data representation can be articulated with those presented in Figure 3.1 of Chapter 3 as shown in Figure 4.17. The connecting point is the data being represented in a semantic-aware numeric space. This space can be improved by removing co-occurrences between features, before being used for detecting typical evolutions or identifying social roles. Of course, integrating our approaches

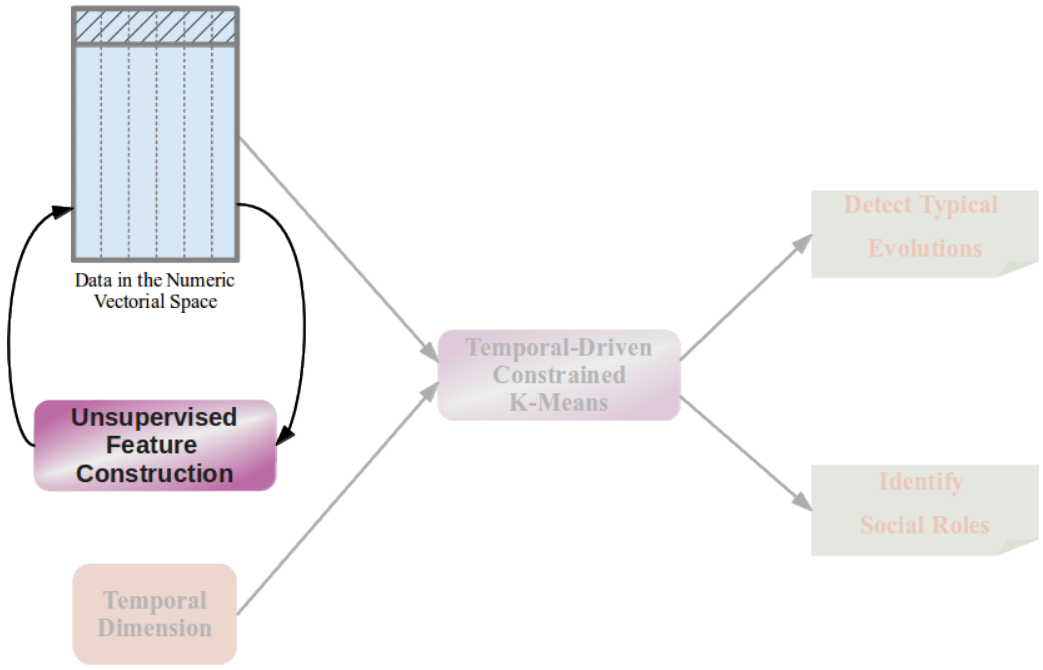


Figure 4.17 – Streamlined schema showing how the contributions in this chapter can be conceptually articulated with those in previous the chapters.

would require additional development, some of which is presented in the next paragraph.

Current work We have already undergoing work toward incorporating temporal information into the feature construction algorithm. This work will allow to simultaneously address the two major research problems of this thesis: the semantic representation and using the temporal dimension of data.

The datasets presented in Section 4.6 have no temporal component. The building block of the **uFC** algorithm is feature co-occurrence. For example, *"manifestation"* co-occurs with *"urban"* because usually manifestations take place in cities. With the introducing of the temporal information, the definition of the problem changes and the question of co-occurrence in a temporal context arises. Some features might co-occur, but not simultaneously. For example, the arrival of power of a socialist government and the increase of the country's public deficit might be correlated, but with a time lag, as the macro-economic indicators have a big inertia. The purpose of this work is to detect such "correlations with a time lag" and create new features like *"socialist"* and *"public_ deficit"* co-occur at a time lag δ. This would allow to (i) improve the data representation by using the temporal dimension in addition to data semantics (ii) detect hidden temporal correlations, which might prove to be causalities. In addition, the newly constructed temporal features can serve to create comprehensible labels to temporal clusters extracted using TDCK-Means (proposed in Chapter 3).

We have extended the correlation coefficient defined in Equation 4.5 to calculate the correlation at with a given fixed lag δ. The experiments we performed so far show that an "optimum" lag δ can be determined, that maximizes the temporal correlation. We are currently working on an extension of the *and* operator to the temporal case. The new

features are no longer constructed as boolean expressions, but as temporal chains of the form $f_i \xrightarrow{\delta_1} f_j \xrightarrow{\delta_2} f_k$, meaning that f_i precedes at at a time distance δ_1, which f_j precedes f_k at a time distance δ_2.

Other future work A research direction we privilege is adapting our algorithms for data of the Web 2.0 (*e.g.*, automatic treatment of labels on the web). Several challenges arise, like very large label sets (it is common to have over 10 000 features), non-standard label names (see standardization preprocessing task that we have performed for the LabelMe dataset in Section 4.6) and missing data. The problem of missing data is intimately linked to the task of semantic-enriched numeric representation construction for images and the assumption of complete labeling that we have made at the beginning of this chapter. In complete labeling, if the label is not present, it therefore means that the object it denotes is absent (in the case of object labeling in images). This assumption supposes binary labeling, where **true** means presence and **false** means absence. In the case of incomplete labeling, the absence of a label might also mean that the user forgot/chose not to label the given image/document. Therefore, a value of false is no longer a sure indicator for the absence of the given object. For example, when a user is labeling an image depicting a cascade and has a choice between *water*, *cascade* or both, he/she might choose only cascade as it is the most specific. This adds new challenges for the feature construction algorithm, since the co-occurrence of water and cascade is no longer present. Similarly, created features which are absent for all individuals can no longer be simply removed, since it is not sure if the objects are really missing or simply their presence was not marked by the user.

Other planned developments include taking into account non-linear correlation between variables by modifying the metric of the search and the co-occurrence measure. We also consider converting generated features to the Disjunctive Normal Form for easier reading and suppressing features that have a low support in the dataset. This would reduce the size of the feature set by removing rare features, but would introduce new difficulties such as detecting nuggets.

A related problem A somehow related class of problems is *multi-label classification* [Tsoumakas & Katakis 2007]. These algorithms work in another paradigm (supervised learning) and tackle another learning task (learning from a dataset with multiple class variables, aiming to learn to differentiate between classes and be able to predict the class of unseen examples). The connection with our learning problem is how these solutions tackle the co-occurrence between the class variable (*i.e.*, when an individual has two labels attached). In our work we do not relate or compare with these approaches, and we mention them here to completeness reasons.

Solutions to the problem of multi-label classification usually fall into two categories [Tsoumakas & Katakis 2007]: (a) *problem transformation methods* and (b) *algorithm adaptation methods*. (a) Problem transformation methods are those methods that transform the multi-label classification problem either into one or more single-label classification or regression problems. For example, protein classification using machine learning algorithms is studied in [Diplaris *et al.* 2005]. For proteins that belong to several classes, they construct new classes using boolean conjunctions: $c' = c_i$ AND c_j. (b) Algorithm adaptation methods

are those methods that extend specific learning algorithms in order to handle multi-label data directly. For example, [Clare & King 2001] adapted the C4.5 algorithm for multi-label data. They modified the formula of entropy in order to take into account the relative frequency of each class. They also allow multiple labels in the leaves of the tree.

While showing some success in the problem of predicting multiple label classification, this family of problems does not tackle the same task as the one stated in Section 4.1: they do not deal with the co-occurrence in the description space. They are mostly interesting at the level of the chosen approach and their similarity towards the construction of new class attributes/labels. Whatsoever, even this construction is limited in the case of multi-label classification: (a) negations cannot exist and, because of this fact, (b) not all boolean formulas can be created (the AND operator is not a complete operator set). Furthermore, (c) only very simple conjunctions of the initial labels can be created: they construct only conjunction of two initial labels, which are not evolved into a more complex formula.

Most of the work presented in this chapter was published in the international **Journal of Intelligent Information Systems** [Rizoiu *et al.* 2013a].

Chapter 5

Dealing with images: Visual Vocabulary Construction

Contents

5.1 Learning task and motivations

In this chapter, we address one of our core research challenges: leveraging semantics when dealing with complex data. Images are one of the most widely encountered types of complex data. Therefore, in this work we are interested in how to leverage external information in order to **construct a semantic-aware numeric representation for images**. The most prevalent learning task involving images is *content-based image classification*. This is a difficult task especially because the low-level features used to digitally describe images usually capture little information about the semantics of the images. In our work, presented schematically in Figure 5.1, we tackle this difficulty by enriching the semantic content of the image representation using external knowledge. The underlying hypothesis of our work is that creating a more semantically rich representation for images would yield higher

machine learning performances, without the need to modify the learning algorithms themselves. This idea is similar to the similarity-based approaches in semi-supervised clustering literature (presented in Section 2.2.1, p. 24), which introduce knowledge into unsupervised algorithms by modifying the distance measure used to judge the similarity of individuals and, afterwards, running the unmodified unsupervised algorithm. As a learning task, we apply our proposition to the task of content-based image classification and we show that semantically enriching the image representation yields higher classification performances.

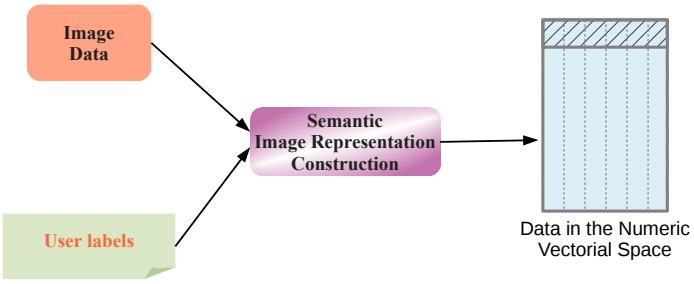


Figure 5.1 – Streamlined schema of our work presented in this chapter: leveraging external knowledge to construct a semantically-enriched numeric description space for images.

The purpose of our work It is noteworthy that semantic are the focus of our work with images, and not the task of content-based image classification. This point of view is crucial for the rest of this chapter. The content-based image classification literature provides many examples (some of which are mentioned in Section 5.2) of systems which achieve good results. Our objective is not to compare with these approaches or show the superiority of our methods on well-known image benchmarks. Likewise, we do not propose a new image representation system. The objective of our work is to show how embedding semantics into an existing image representation can be beneficial for a learning task, in this case image classification. Starting from a baseline image representation construction (described in Section 5.1.2), we propose two algorithms that make use of external information under the form of non-positional tags, to enrich the semantics of the image representation. We use both the baseline representation and our semantically improved representation in an image classification task and we show that leveraging semantics consistently provides higher scores.

Context and motivations Image creation and sharing pre-date written (textual) sources. Some of the oldest cave painting go as far back in time as approximately 40,000 years ago. But the true explosion and large scale production of image data have started in modern days, with the maturing of the image acquisition, storing, transmission and reproduction devices and techniques. At the same time, the Web 2.0 allowed easy image sharing and recently even search capabilities (*e.g.*, Instagram [1], Flickr [2]). Social Networks rely heavily on image sharing (which sometimes may pose privacy problems, but such a discussion

1. http://instagram.com/
2. http://www.flickr.com/

is out of the scope of this document). Because of the sheer volumes of created images, automatic summarization, search and classification methods are required.

The difficulty when analyzing images comes from the fact that digital image numerical format does not embed the needed semantic information. For example, images acquired using a digital photo camera are most often stored in raster format, based on pixels. A pixel is an atomic image element, which has several characteristics the most important being the size (as small as possible) and its color. Other information can be color coding, alpha channel *etc.*. Therefore, an image is stored numerically as a matrix of pixels. The difficulty raises from the fact that low-level features, such as position and color of individual pixels, do not capture too much information about the semantic content of the image (*e.g.*, shapes, objects). To address this issue, multiple representation paradigms have been proposed, some of which will be presented in Section 5.2. The one showing the most promising results is the "bag-of-features" representation, a representation inspired from the textual "bag-of-words" textual representation (detailed later in Section 6.2, p. 128).

The remainder of the chapter is structured as follows: the rest of this section presents an overview of our proposals (in Section 5.1.1) and how to construct a baseline "bag-of-features" image description (in Section 5.1.2). In Section 5.2, we present a brief overview on constructing a numerical image representation, concentrating on some of the state-of-the-art papers that relate to visual vocabulary construction and knowledge injection into image representation. Section 5.3 explains the two proposed approaches, followed, in Section 5.4, by the experiments that were performed. Some conclusions are drawn and future work perspectives are given in Section 5.5.

5.1.1 An overview of our proposals

The focus of our work is embedding semantic information into the construction of image numerical representation. The external information is under the form of non-positional labels, which signal the presence in the image of an object (*e.g.*, car, motorcycle) or give information about the context of the image (*e.g.*, holiday, evening), but do not give any information about its position of the image (in the case of objects). Furthermore, the labels are available only for a part of the image collection, therefore positioning our work in a semi-supervised learning context.

Our work is focused on the *visual vocabulary* construction (which is also referred in the literature as *codebook* or *model*). In the *"bag-of-features"* (*BoF*) representation, the visual words serve a similar role as the real textual words do in the "bag-of-words" representation. We propose two novel contributions that leverage external semantic information and that allow the visual vocabulary to capture more accurately the semantics behind a collection of images. The first proposal deals with introducing the provided additional information early in the creation of the visual vocabulary. A *dedicated visual vocabulary* is constructed starting from the visual features sampled from images labeled with a given label. Therefore, a dedicated vocabulary contains visual words adapted to describing the object denoted by the given label. In the end, the complete visual vocabulary is created by merging the dedicated vocabularies. In the second proposal, we add a filtering phase as a pre-processing of the visual vocabulary construction. For any given image, we construct a known positive set (images labeled with the same labels as the given image) and a known negative set

(images that do not share any labels with the given image). If a visual feature, sampled from the target image, is more similar to features in the known negative set than to features in the known positive set, then there are high chances that it does not belong to the objects denoted by the labels of the given image and it can, therefore, be eliminated. This reduces the influence of irrelevant features in the vocabulary construction and increases the accuracy of the classification process. The two approaches are combined into a visual vocabulary construction technique and shown to consistently provide better performances than the baseline technique presented in Section 5.1.2.

5.1.2 Constructing a baseline "bag-of-features" image numerical description

The "bag-of-features" [Csurka *et al.* 2004, Zhang *et al.* 2007] (*BoF*) representation is an image representation inspired from the "bag-of-words" (*BoW*) textual representation, which is detailed in Section 6.2 (p. 128). The *BoW* representation is an orderless document representation, in which each document is depicted by a vector of frequencies of words over a given dictionary. *BoF* models have proven to be effective for object classification [Csurka *et al.* 2004, Willamowski *et al.* 2004], unsupervised discovery of categories [Fei-Fei & Perona 2005, Quelhas *et al.* 2005, Sivic *et al.* 2005] and video retrieval [Sivic & Zisserman 2003, Chavez *et al.* 2008]. For object recognition tasks, local features play the role of "visual words", being predictive of a certain "topic" or object class. For example, a wheal is highly predictive of a bike being present in the image. If the visual dictionary contains words that are sufficiently discriminative when taken individually, then it is possible to achieve a high degree of success for whole image classification. The identification of the object class contained in the image is possible without attempting to segment or localize that object, simply by looking which visual words are present, regardless of their spatial layout. Overall, there is an emerging consensus in recent literature that *BoF* methods are effective for image description [Zhang *et al.* 2007].

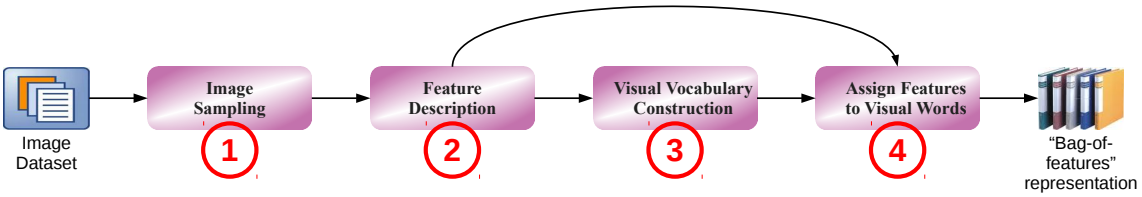


Figure 5.2 – Construction flow of a "bag-of-features" numerical representation for images

Baseline construction Typically, constructing a *BoF* image representation is a four phase process, as shown in Figure 5.2. Starting from a collection $\mathcal{P}$ containing n images, the purpose is to translate the images into a numerical space which the learning algorithm is efficient. In *phase 1*, each image $p_i \in \mathcal{P}$ is sampled and l_i patches (features) [3] are extracted. Many sampling techniques have been proposed, the most popular being dense grid sampling [Fei-Fei & Perona 2005, Vogel & Schiele 2007] and salient keypoint detector [Csurka

3. l_i is dependent on the content on the image (number of objects, shape *etc.*) and the extraction algorithm used. It can vary from a couple of hundreds of features up to several tens of thousands.

(a) (b) (c)

Figure 5.3 – Example of feature corresponding to the visual words associated with “wheal” (in red) and “exhaust pipe” (in green)

et al. 2004, Fei-Fei & Perona 2005, Sivic *et al.* 2005]. In *phase 2*, using a local descriptor, each feature is described using a h-dimensional [4] vector. The SIFT [Lowe 2004] and the SURF [Bay *et al.* 2006] descriptors are popular choices. Therefore, after this phase, each image p_i is numerically described by $V_i \subset \mathbb{R}^h$, the set of h-dimensional vectors describing features sampled from p_i.

Based on these numeric features, in *phase 3*, a visual vocabulary is constructed using, for example, one of the techniques presented in Section 5.2.2. This is usually achieved by means of clustering of the described features, and the choice is usually the K-Means clustering algorithm, for its linear execution time required by the high number of features. The visual vocabulary is a collection of m visual words, which are described in the same numerical space as the features and which serve as the bases of the numerical space in which the images are translated. More precisely, the centroids created by the clustering algorithm serve as visual words. In clustering, centroids are the abstractions of a group of documents, therefore summarizing the common part of the documents. In the above example, all the visual features extracted from the region of an image depicting the wheal of a bike will be regrouped together into one or several clusters. The centroid of each cluster represents a visual word, which is associated with the wheal. Figure 5.3, we depict three examples of images portraying bikes. In each image, we highlight 3 features: two corresponding to visual words associated with “wheal” and one associated with a visual word associated with ”exhaust pipe”.

At *phase 4*, each sampled feature is assigned to a visual word. Similarly to the *BoW* numerical description for texts, each image is described as a distribution over the visual words, using one of the term weighting scheme (*e.g.*, tf, $tfxidf$ *etc.*) described in Section 6.2 (p. 128). In the previous example, the distribution vector associated with each of the images in Figure 5.3 has a high count for the visual words associated with “wheel”, “exhaust pipe”, and “sadle”. The resulting numerical description can then be used for classification, information retrieval or indexation tasks.

4. *e.g.* for the SIFT descriptor $h = 128$.

5.2 Context and related work

Over the past decades computer vision domain has seen a large interest from the research community. Its application are larger than image analysis and include augmented reality, robotic vision, gesture recognition *etc.* Whatsoever, in the context of Internet-originating images, one of the prevailing task is content-based image classification. Some of the initial image classification systems used color histograms [Swain & Ballard 1991] for image representation. Such a representation does not retain any information about the shapes of objects in images and obtains moderate results. Other systems [Haralick & Shanmugam 1973, de Medeiros Martins *et al.* 2002, Varma & Zisserman 2003, Lazebnik *et al.* 2003b] rely on texture detection. Texture is characterized by the repetition of basic elements or *textons*. For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters. The *BoF* orderless representation has imposed itself as the state-of-the-art in image representation, for classification and indexation purposes. The process of constructing the representation includes sampling the image (*phase 1* in Figure 5.2), describing each features using an appearance-based descriptor (*phase 2*), constructing a visual vocabulary (*phase 3*) and describing images as histograms over the visual words (*phase 4*).

The remainder of this section presents a brief overview (i) of the sampling strategies and numerical descriptors for image keypoints present in literature (in Section 5.2.1) and (ii) of the visual vocabulary construction techniques, concentrating on how external information can be used to improve the vocabularies representativity (in Section 5.2.2).

5.2.1 Sampling strategies and numerical description of image features

Image sampling methods Image sampling for the *BoF* representation is the process of deciding which regions of a given image should be numerically described. In Figure 5.2, it corresponds to *phase 1* of the construction of a *BoF* numerical representation. The output of feature detection is a set of patches, identified by their locations in the image and their corresponding scales and orientations. Multiple sampling methods exist [O'Hara & Draper 2011], including *Interest Point Operators*, *Visual Saliency* and random or dense grid sampling.

Interest Point Operators [Lowe 1999, Kadir & Brady 2001] search to find patches that are stable under minor affine and photometric transformations. Interest point operators detect locally discriminating features, such as corners, blob-like regions, or curves. A filter is used to detect these features, measuring the responses in a three dimensional space. Extreme values for the responses are considered as interest points. The popular choice is the Harris-Affine detector [Mikolajczyk & Schmid 2004], which uses a scale space representation with oriented elliptical regions. *Visual Saliency* [Frintrop *et al.* 2010] feature detectors are based on *biomimetic* computational models of the human visual attention system. Less used by the *BoF* literature, these methods are concerned with finding locations in images that are visually salient. In this case, fitness is often measured by how well the computational methods predict human eye fixations recorded by an eye tracker. There are research [Sivic & Zisserman 2003] that argue that interest point-based patch sampling, while useful for image alignment, is not adapted for image classification tasks. Examples are city images,

for which the interest point detector does not consider relevant most of the concrete and asphalt surroundings, but which are good indicators of the images' semantics. Some approaches sample patches by using *random sampling* [Maree *et al.* 2005]. [Nowak *et al.* 2006] compare a random sampler with two interest point detectors: *Laplacian of Gaussian* [Lindeberg 1993] and *Harris-Laplace* [Lazebnik *et al.* 2003a]. They show that when using enough samples, random sampling exceeds the performance of interest point operators. *Spatial Pyramid Matching* is proposed in [Lazebnik *et al.* 2006]. Introduces spacial information in the orderless *BoF* representation by creating a pyramid representation, where each level divides the image in increasingly small regions. Feature histogram is calculated for each of these regions. The distance between two images using this spatial pyramid representation is a weighted histogram intersection function, where weights are largest for the smallest regions.

Feature descriptors With the image sampled and a set of patches extracted, the next questions is how to numerically represent the neighborhood of pixels near a localized region. In Figure 5.2, this corresponds to *phase 2* of the construction of a *BoF* numerical representation. Initial feature descriptors simply used the pixel intensity values, scaled for the size of the region. The *normalized pixel values* have been shown [Fei-Fei & Perona 2005] to be outperformed by more sophisticated feature descriptors, such as the SIFT descriptor. The *SIFT* (Scale Invariant Feature Transform) [Lowe 2004] descriptor is today's most widely used descriptor. The responses to 8 gradient orientations at each of 16 cells of a 4x4 grid generate the 128 components of the description vector. Alternative have been proposed, such as the *SURF* (Speeded Up Robust Features) [Bay *et al.* 2006] descriptor. The SURF algorithm contains both feature detection and description. It is designed to speed up the process of creating features similar to those produced by a SIFT descriptor on Hessian-Laplace interest points by using efficient approximations.

5.2.2 Unsupervised visual vocabulary construction

The visual vocabulary is a mid-level transition key between the low-level features and a high-level representation. It is a prototypic representation of features that are discriminative in a classification context.

The visual vocabulary is used to reduce dimensionality and to create a fixed length numerical representation for all images [5]. Most *BoF* approaches use clustering to created the visual vocabulary, usually the K-Means [Sivic & Zisserman 2003, Lazebnik *et al.* 2006, Jiang *et al.* 2007] algorithm. K-Means is used for the fact that it produces centroids, which are prototypes of similar features in the same cluster. Its linear execution time is a plus considering the high volume of individuals to be processed. Some authors [Jurie & Triggs 2005] argument that in K-Means, centroids are attracted by dense regions and under-represent less denser, but equally informative regions. Therefore, methods were proposed for allocating centers more uniformly, inspired by mean shift [Comaniciu & Meer 2002] and on-line facility location [Meyerson 2001]. Other visual vocabulary construction techniques do not rely on K-Means. For example, [Moosmann *et al.* 2007] use an *Extremely Randomized Clustering Forest*, an ensemble of randomly created clustering trees. This technique provides

5. The number of extracted features can greatly vary depending on the image and the method used for sampling.

good resistance to background clutter, but the main advantage over K-Means is the faster training time.

One of the most important parameters in the construction of the visual vocabulary is its dimension, which has a powerful impact on both performance and computational complexity [Csurka *et al.* 2004, Jurie & Triggs 2005]. It has been shown [Jiang *et al.* 2007, López-Sastre *et al.* 2010, Nowak *et al.* 2006] that a large vocabulary may lead to overfitting for construction techniques based on interest points detection. As our experiments show (in Section 5.4.6), even a random vocabulary (in a random vocabulary, a number of features are randomly chosen to serve as visual words) can lead to overfitting if its dimension is high enough.

5.2.3 Leveraging additional information

The *BoF* representation yields surprising results for image classification and indexing. This is because there is an intrinsic relation between the "quantity" of semantic information captured by the description space and the performances of machine learning algorithms (*e.g.*, in a classification task, the separability of individuals in the description space is crucial). Therefore, one direction to further improve results is to construct new representations that capture even more semantics from the raw image data. Another direction, the one that we privilege in our work, is to use external information to further enrich the semantic content of the constructed representation. In the case of Internet-originating images, precious information is given either by the textual context of images (*e.g.*, titles, descriptions *etc.*), or by labels attached to the images (*e.g.*, on social networks websites, users have the option to label the presence of their friends in images). Of course, the literature presents approaches that leverage other resources to semantically enrich enrich the image representation (*e.g.*, [Athanasiadis *et al.* 2005] propose a system that links low-level visual descriptors to high-level, domain-specific concepts in an ontology). In Section 2.1.2 (p. 13), we have already presented a number of learning tasks that can benefit from using image and textual data simultaneously. In the following paragraphs, we detail some of the methods present in the literature that address the use of additional information under the form of text or labels in order to improve image classification results and we position our work relative to these approaches.

Leveraging the image's textual context In [Morsillo *et al.* 2009], the text that comes alongside the images is used to improve the visual query accuracy. A *BoF* representation for images is created as shown in Section 5.1.2, with the exception that color information is also added to the keypoint description. An 11-dimension vector coding the color information of the sampled patches is added to the 128-dimension vector generated by the SIFT. The text that surrounds the images in the web pages is used to extract topics, using LDA [Blei *et al.* 2003]. The inferred topics are, afterwards, used to describe the textual information (therefore functioning as a dimension reduction technique). The textual and the image data are used together to estimate the parameters of a probabilistic graphical model, which is trained using a small quantity of labeled data. Another approach that uses the text accompanying images originating from the Internet is presented in [Wang *et al.* 2009]. An auxiliary collection of Internet-originating images, with text attached, is used to create a

textual description of a target image. Images are described using three types of features: the SIFT features, the GIST features [Oliva & Torralba 2001] and local patch color information. For each test image, the K most similar images (in terms of visual features) are identified in the auxiliary collection. The text associated with these near neighbor images is summarized to build the text feature. The label of each image is considered as a unit (*i.e.*, a whole phrase is considered as an item) and the text feature is constructed as a normalized histogram over labels. A text classifier and a visual classifier are trained and the outputs of the two classifiers are merged for a more accurate description of the photo. [Mooney *et al.* 2008] use co-training [Blum & Mitchell 1998] to construct a classifier starting from textual and visual data. Text is described using a *BoW* representation, whereas images are described using region-based features. Each image is divided into a number of regions of fixed dimension (4-by-6 pixels), which are described using texture and color features. Co-training is a semi-supervised classification technique, which first learns a separate classifier for textual data and image data, using any labeled examples. The most confident predictions of each classifier on the unlabeled data are then used to iteratively construct additional labeled training data, and the classifiers are re-trained.

Leveraging external semantic knowledge Other solutions rely on external expert knowledge in order to guide the visual vocabulary construction. This knowledge is most often expressed under the form of class/category annotations or labels (*e.g.* signaling the presence of an object inside an image) or semantic resources, such as WordNet. [Zhang *et al.* 2009] uses an iterative boosting-like approach. Each iteration of boosting begins by learning a visual vocabulary according to the weights assigned by the previous boosting iteration. The resulting visual vocabulary is then applied to encode the training examples, a new classifier is learned and new weights are computed. The visual vocabulary is learned by clustering using K-Means a "learning" subset of image features. Features from images with high weights have more chances of being part of the learning subset. To classify a new example, the AdaBoost [Freund & Schapire 1997] weighted voting scheme is used.

[Perronnin *et al.* 2006] construct both a generic vocabulary and a specific one for each class. The generic vocabulary describes the content of all the considered classes of images, while the specific vocabularies are obtained through the adaptation of the universal vocabulary using class-specific data. Any given image can, afterwards, be described using the generic vocabulary or one of the class-specific vocabularies. A semi-supervised technique [Ji *et al.* 2010], based on Hidden Random Markov Fields, uses local features as Observed Fields and Semantic labels as Hidden Fields and employs WordNet to make correlations. Some works [Fulkerson *et al.* 2008, Hsu & Chang 2005, Lazebnik & Raginsky 2009, Winn *et al.* 2005] use mutual information between features and class labels in order to learn class-specific vocabularies, by merging or splitting initial visual words quantized by K-Means. Another work [Liu *et al.* 2009] presents an algorithm used for learning a generic visual vocabulary, while trying to preserve and use the semantic information in the form of a point-wise mutual information vector. It uses the diffusion distance to measure intrinsic geometric relations between features. Other approaches [Marszałek & Schmid 2006] make use of label positioning in the images to distinguish between foreground and background features. They use weights for features, higher for the ones corresponding to objects and lower for the background.

Our positioning In the methods presented earlier, we identify several approaches towards improving the results of classification algorithms: (a) improving image representation semantics by combining multiple types of visual features (*e.g.*, SIFT, color, texture *etc.*, no external information is leveraged), (b) modifying the classification algorithm to take into account the text/label information (usually by training separate classifiers for (i) text and image or (ii) based on each label), (c) training and using multiple vocabularies to describe an image and (d) making use of positional labels to filter features unlikely to be relevant. Positional labels are labels in which the position of the objects in images are known, in addition to their presence. This kind of labeling is usually more costly to perform than non-positional labeling.

Our proposals deal with leveraging external information to enrich the semantics of the image representation. The additional information is taken into account at the level of the representation construction. We do not modify the learning algorithm, therefore our proposals are compatible with existing classification algorithm. Our proposals can be classified under the previous point (c), since we construct multiple *dedicated visual vocabularies*. The original contribution is that we propose a filtering algorithm that removes features unlikely to be relevant for a given object using only non-positional labels.

5.3 Improving the *BoF* representation using semantic knowledge

In this section we present two novel methods that leverage external semantic information, under the form of non-positional *object labels*, into the visual vocabulary construction. Our work is positioned in a weakly supervised context, similar to the one defined by [Zhang *et al.* 2007]. Each label signals the presence of a given object in an image, but not its position or boundaries. Our approaches use the semantic information to increase the relevancy of the visual vocabulary. Our first approach follows an idea similar to some of the systems already presented in Section 5.2.2. For each label, we construct a *dedicated visual vocabulary*, based only on the images with a certain label. Such approaches have been shown [Perronnin *et al.* 2006, Jianjia & Limin 2011] to improve accuracy over a general purpose vocabulary, since specialized vocabularies contain visual words that more appropriately describe the objects appearing in the image collection. In our second approach, we further improve accuracy by proposing a novel pre-processing phase, which filters out features that are unlikely to belong to the respective object. Our filtering proposal follows the framework of the object recognition algorithm proposed in [Lowe 2004] and uses a positive and a negative example set, constructed based on the labels. The filtering pre-processing is combined with the *dedicated visual vocabulary* construction, and we show in Section 5.4 that this approach consistently achieves higher accuracy then both a dedicated vocabulary (with no filtering) and a general purpose vocabulary.

Including semantic knowledge The semantic knowledge is presented under the form of a collection $\mathcal{T}$ of k labels, $\mathcal{T} = \{t_i | i = 1, 2...k\}$. Each label is considered to denote an object in the image (*e.g.*, a car, a person, a tree), but no positional markers are available. We make the assumption that the objects denoted by labels do not overlap in the images

and their appearance in the dataset is not correlated (*e.g.*, if a car appears, it does not necessarily mean that there is a person next to it). While these are strong assumptions, we will discuss ways of relaxing them in Section 5.5. Furthermore, as the case of our work concerning the unsupervised feature construction, we consider the labeling to be complete (*i.e.*, if an image does not have a given label, than the object does not appear in the image). We discuss in further detail the effects of incomplete labeling after presenting our proposals, in Section 5.3.2 and in Chapter 8.

Only a fraction of the image dataset is labeled and we use both labeled and unlabeled images to construct the semantic-aware representation, therefore positioning our work in the domain of semi-supervised learning. We denote by $\mathcal{P}$ the input collection, having n images. n_1 images are labeled, thus forming the labeled set ($\mathcal{P}_1$), while the remaining images have no labels. The *a priori* label information is presented in the form of a boolean matrix $Y \in \{0,1\}^{n_1 \times k}$, having n_1 lines and k columns so that

$$y_{i,j} = \begin{cases} 1 & \text{if image } p_i \in \mathcal{P}_1 \text{ is labeled using } t_j; \\ 0 & \text{otherwise.} \end{cases}$$

5.3.1 Dedicated visual vocabulary generation

The idea behind the *BoF* representation is that the visual words are predictive for certain objects (as seen in Section 5.1.2). The quality of the visual words (and their predictive power) would be enhanced if they are constructed starting only from the features extracted from the respective objects. This would eliminate the background originated features and features belonging to other objects. In a weakly supervised context, the object boundaries are unknown, but selecting only the images that contain a certain object increases the relevant/noise feature ratio. Consequently, the resulted visual words are more accurate descriptions of the objects denoted by the labels. We propose to construct a *dedicated visual vocabulary* for each label $t_i \in \mathcal{T}$, starting only from features extracted from the images labeled with t_i.

The proposed method is presented in Algorithm 4. We make no assumptions about the number of visual words needed to describe each object and, therefore, visual words are distributed equally among objects. We construct k dedicated vocabularies, each one containing m/k visual words. Other division techniques can be imagined and make part of the perspectives of our work. Each dedicated vocabulary is created in the standard *BoF* approach, shown in Section 5.1.2. For a given label t_i, we create $\mathcal{C}_i$, the collection of all the features extracted from images labeled with t_i. Formally:

$$\mathcal{C}_i = \bigcup_{\substack{j=1 \\ y_{j,i}=1}}^{n_1} V_j$$

where V_j is the set of numerically described features sampled from image p_j. The function **choose_features_at_random** is used to initialize the dedicated vocabulary M_i with m/k features randomly picked from $\mathcal{C}_i$. The function **ameliorate_using_K-Means** evolves the visual vocabulary M_i by clustering the features in $\mathcal{C}_i$ around the visual words, using the K-Means algorithm. The Euclidean distance is used to measure the similarity of

Algorithm 4 Dedicated vocabulary generation algorithm.

Input: $\mathcal{C} = \{V_i \mid i = 1, 2..n_1\}$ - set of features sampled from labeled images
Input: $Y \in \{0,1\}^{n_1 \times k}$ - image/label association matrix
Input: m - the dimension of the visual vocabulary M
Output: the visual vocabulary M having m visual words

```
// for each label
for i = 1 to k do
    m_i ← m/k                    // size of the dedicated vocabulary
    C_i = ⋃_{j=1}^{n_1} V_j | y_{j,i} = 1    // set of features in images labeled with t_i
    // construct dedicated visual vocabulary M_i
    M_i ← choose_features_at_random (m_i, C_i)
    M_i ← ameliorate_using_K-Means (M_i, C_i)
// merge the dedicated visual vocabularies
M ← ∅
for i = 1 to k do
    M ← concatenate_vocabularies(M, M_i)
```

the numeric description of two features. The set of resulted visual words represent more accurately the object denoted by the label t_i. At the end of the algorithm, the **concatenate_vocabularies** function merges the *dedicated vocabularies* $M_i, i = 1, 2..k$ into the general visual vocabulary M. This ensures that the generated visual vocabulary contains visual words which describe all the objects labeled with labels in $\mathcal{T}$.

Temporal complexity Algorithm 4 has a linear execution time, if we consider that matrix operations are indivisible and executed in $O(1)$, which is the case in modern vectorial mathematical environments. Since we are executing K-Means k times, the temporal complexity will be $no_{iter} \times k \times O(m/k \times n_{t_i})$, where n_{t_i} is the number of images labeled with t_i and no_{iter} is the number of performed iterations (usually limited, thus ignored in practice). That leads to a theoretical complexity of $O(m \times n)$, equal to that of K-Means.

5.3.2 Filtering irrelevant features

We propose a filtering mechanism in order to further increase the relevant/noise features ratio in the dedicated vocabulary construction technique presented in the previous Section 5.3.1: we detect and filter the features that are unlikely to be related to the object denoted by a given label. Given an image $p_i \in \mathcal{P}_1$, we construct two auxiliary image collections: the *known positive set*, which contains only images that are labeled identically as p_i, and the *known negative set*, which contains images that do not share any tags with p_i (given the complete labeling assumption). In practice, we limit the sizes of the *known positive set* and the *known negative set* to a maximum number of images, given by a parameter $maxFiles$. We define KP_{p_i} as the set of features sampled from images in the positive set and KN_{p_i} as the set of features sampled from the negative set:

$$KP_{p_i} = \{f^+ \in V_j \mid \forall\, t_l \in \mathcal{T} \text{ for which } y_{i,l} = 1 \implies y_{j,l} = 1\}$$
$$KN_{p_i} = \{f^- \in V_j \mid \forall\, t_l \in \mathcal{T} \text{ for which } y_{i,l} = 1 \implies y_{j,l} = 0\}$$

(a) (b) (c)

Figure 5.4 – (a) An image labeled "motorbike", (b) an image from the *known positive set* and (c) an image from the *known negative set*

Consider a feature sampled from p_i ($f \in V_i$), which is more similar to the features in the negative collection ($f^- \in KN_{p_i}$) rather than the ones in the positive collection ($f^+ \in KP_{p_i}$). Such a feature has a higher chance of belonging to the background of p_i rather than to the objects in the image. It can, therefore, be filtered. To measure the similarity of two features, the *euclidean distance* is usually used: $||f_1 - f_2|| = \sqrt{\Sigma_{i=1}^{h} (f_{1,i} - f_{2,i})^2}$. Formally, for a feature f sampled from an image p_i:

$$\begin{aligned} f \in V_i \text{ is filtered } &\Leftrightarrow \nexists f^+ \in KP_{p_i} \text{ so that } ||f - f^+|| \leq \delta \\ \text{with } \delta &= \alpha \times \min_{f \in KN_{p_i}} ||f - f^-|| \end{aligned} \tag{5.1}$$

where δ is the filtering threshold and $\alpha \in \mathbb{R}^+$ is a parameter, which allows the fine tuning of the filtering threshold. The filtering threshold δ is defined as the distance from the feature f to the closest feature in the known negative set, scaled by tuning parameter α. The influence of parameter α on the effectiveness of the filtering is studied in Section 5.4.7. A feature f is considered similar to a feature $f^+ \in KP_{p_i}$ if and only if $||f - f^+||$ is lower than the filtering threshold. Therefore, the feature f is removed when it has no similar feature in the known positive set.

Let's take the example of image collection depicted in Figure 5.4. The images in Figures 5.4a and 5.4b are labeled "motorbike", whereas the image in Figure 5.4c is labeled "city". The target image in Figure 5.4a has buildings in the background, and any feature sampled from that region of the image would be irrelevant for the object motorbike. Figure 5.4b serves as *known positive set*, while Figure 5.4c serves as *known negative set*. We take the example of two features f_1 sampled from the wheal of the motorbike (shown in green) and f_2 sampled from the buildings in the background (shown in red), of the target image. For f_1, at least one similar feature exists in the positive set. For f_2, no similar features exist in the known positive set. f_2 is, therefore, eliminated as it is considered not relevant for the object motorbike.

Algorithm 5 presents the proposed filtering algorithm. The algorithm has two parameters $maxFiles$, which controls the maximum size of the KP_{p_i} and KN_{p_i} sets, and α, which controls how strict is the filtering. For each labeled image p_i, the functions **create_KP** and **create_KN** are used to created the feature sets KP_{p_i} and, respectively, KN_{p_i}. The **count_similar** function is used to count how many features in KP_{p_i} have the similarity

Algorithm 5 Filtering irrelevant features.

Input: $\mathcal{C} = \{V_i \mid i = 1, 2..n_1\}$ - set of features sampled from labeled images
Input: $Y \in \{0,1\}^{n_1 \times k}$ - image/label association matrix
  Parameter: α - parameter controlling the threshold
  Parameter: $maxFiles$ - controls the size of the known positive and known negative sets
Output: V_i^f, $i = 1, 2..n_1$ - sets of the filtered features in each labeled image

```
// for each labeled image
for i = 1 to n_1 do
    V_i^f ← ∅
    T_i ← {t_j | y_{i,j} = 1}          // the labels of image p_i
    KP_{p_i} ← create_KP(i, T_i, Y, C, maxFiles)      // KnownPositive set
    KN_{p_i} ← create_KN(i, T_i, Y, C, maxFiles)      // KnownNegative set
    // process each feature in current image p_i
    for each f ∈ V_i do
        δ ← α× min_distance(f, KN_{p_i})
        count ← count_similar(f, KP_{p_i}, δ)
        if count > 0 then
            V_i^f ← V_i^f ⋃ {f}
```

distance lower than the filtering threshold. If there exists at least one such feature in the KP_{p_i} set, then f is added to V_i^f, the filtered feature set of p_i.

Temporal complexity In Algorithm 5, for comprehension reasons, operations are presented for each feature f sampled from the image p_i. In reality, in vectorial mathematical environments (*e.g. Octave*), matrix operations are unitary and considered as executed in $O(1)$. Thus, the algorithm has a linear execution time $O(n_1 \times maxFiles)$.

Incomplete labeling In the proposed approaches, as well as in the experiments presented in Section 5.4, we make the assumption of complete labeling: if an object occurs in an image, then it is sure that the image has its corresponding label attached. In the case of incomplete labeling, an object might appear in an image p, but the associate label t is not set for the image p. For the dedicated vocabulary construction, incomplete labeling has a limited impact, especially if the dataset is large enough. It only means that the image p is left out when constructing the vocabulary for label t. For the filtering proposal, missing labels mean that the image p has a chance of being selected for the *known negative set* for an image labeled with t. This translates into a very high filtering threshold. Still, this should not pose problems if the *known positive set* also contains images depicting the given object. A given feature needs to have only one similar feature in the known positive set to be considered representative for the object. Furthermore, considering that our algorithms are devised to work in a semi-supervised context, a limited number of completely labeled images is required. This reduces considerably the manual labeling effort.

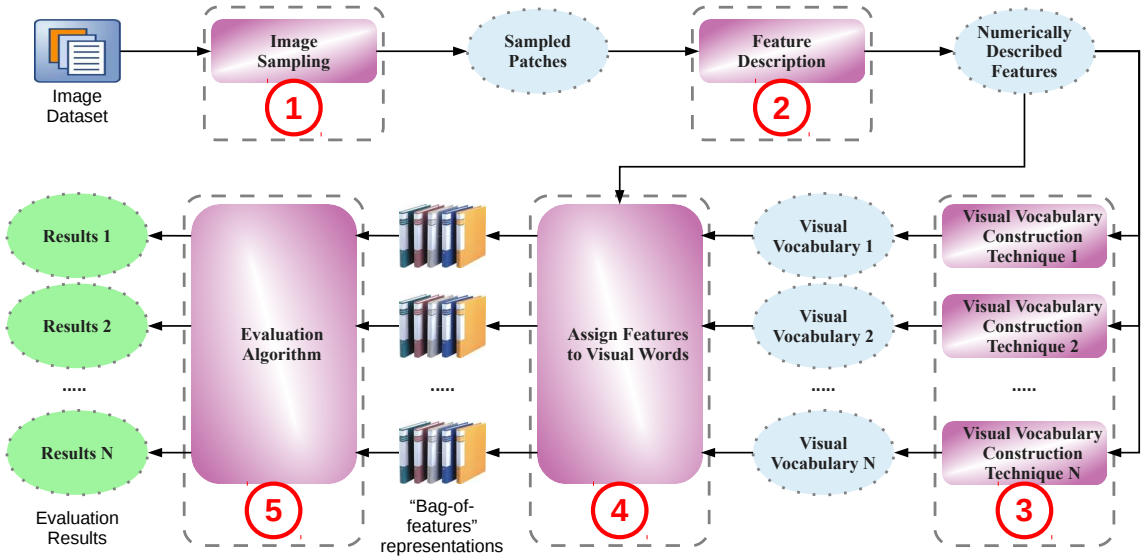


Figure 5.5 – Schema for evaluating multiple visual vocabulary construction techniques.

5.4 Experiments and results

As already pointed out in Section 5.1, the focus of our work is enriching the semantics of the numerical representation of images. Therefore, the purpose of the experiments presented in this section is to compare the semantically enriched representations created by our proposals to a standard baseline representation, created as described in Section 5.1.2. Whatsoever, comparing the semantically-enriched and the baseline representations cannot be done directly, and they are evaluated in the context of a machine learning task, in this case a content-based image classification.

More precisely, given the fact that we perform the semantic injection at the level of the visual vocabulary construction, the experimental protocol streamlined in Figure 5.5 and further detailed in Section 5.4.1, is designed to quantify the differences of performance due only to the visual vocabulary construction. The evaluation is a five phase process, out of which four phases (*1, 2, 3* and *5*) are identical for all techniques. The first four phases correspond to the *BoF* representation construction (see Figure 5.2, p. 100), while the last phase corresponds to the learning algorithm.

We summarize here after each of the phases, which are detailed in the next sections:

- *phase 1*: **image sampling**, identical for all compared approaches;
- *phase 2*: **feature numerical description** of patches, identical for all compared approaches;
- *phase 3*: **visual vocabulary construction**, using the baseline approaches and our semantically-enriching approaches;
- *phase 4*: **feature assignment to visual words**, identical for all compared approaches;
- *phase 5*: **learning algorithm**, each resulted representation is used with two classifiers (a clustering-based and an SVM), identical for all compared approaches.

5.4.1 Experimental protocol

Starting from a given image dataset, we construct, for each image, four *BoF* representations corresponding to the four evaluated visual vocabulary construction techniques (in *phase 3*). The image sampling (*phase 1*), the feature description (*phase 2*) and the image description (*phase 4*) are performed each time using the same algorithms and with the same parameters. In the end, the performances of each obtained *BoF* representation are measured and compared in the context of a content-based image classification task (detailed in Section 5.4.2). The visual vocabulary construction phase is the only phase to vary between the different constructed representations. Therefore, we consider the classifier performance differences a direct consequence of the vocabulary construction.

The invariant *phases 1, 2* and *4* In *phase 1*, images are sampled using a Hessian-Affine region detector and patches are described, in *phase 2*, using the SIFT descriptor [Lowe 2004]. We use the default parameters for these algorithms and we keep them unchanged during the experiments. The visual vocabulary is constructed in *phase 3* using the construction technique to be evaluated. In *phase 4*, the final numerical representation is created, for each image, by associating features to visual words, using the tf term weighting scheme. To reduce the hazard component that appears in all the considered techniques, each construction is repeated 3 times and average results are presented.

Compared vocabulary construction techniques (*phase 3*) Four visual vocabulary construction techniques are evaluated: two classical techniques **random**, **random+km** and our proposals **model** and **filt+model**. **random** constructs a random vocabulary (features are randomly chosen to serve as visual words). For **random+km**, we take the random features selected previously and we ameliorate them by using the **ameliorate_using_K-Means** function presented in Section 5.3.1. **random+km** is the baseline construction technique presented in Section 5.1.2. **model** is our proposal for dedicated vocabulary construction presented in Algorithm 4. In **filt+model** we applied the filtering technique presented in Algorithm 5 as a pre-processing phase before the dedicated vocabulary construction.

5.4.2 The learning task: content-based image classification

Each of the image representations obtained as shown in the previous sections, are used in a content-based image classification task. Two classifiers, an SVM and a clustering-based classifier, are trained and evaluated on each representation, as described in the following paragraphs.

The SVM classifier [Cortes & Vapnik 1995] The SVM classifier evaluation respects the experimental setup recommended by the authors of the `Caltech101`[6] dataset. One of the challenges when evaluating in Data Mining is the disequilibrium between the class cardinality (usually it is the minority class that is of interest). This disequilibrium can cause errors in estimating the generalization error of the constructed model. Usually, the

6. http://www.vision.caltech.edu/Image_Datasets/Caltech101/

disequilibrium is the result of a certain reality in the population from which the sample was extracted (*e.g.* the population of sick individuals is a minority compared to the healthy population). But in the case of image datasets like `Caltech101`, the disequilibrium is only the result of the choice of its creator and represents no reality that needs to be taken into account. We choose to equilibrate the classes before training the classifier, by randomly selecting 30 examples for each label to be part of the learning set. 15 images in the learning corpus are randomly selected to be part of the labeled set $\mathcal{P}_1$. We test on all remaining individuals, which means that the generalization error on majority classes will be better estimated. Evaluation indicators are calculated for each class and we report only the non-weighted averages. The process is repeated 10 times: we create 10 learning sets and the corresponding 10 testing sets. We report the average performances over the 10 executions. The results are expressed using the True Positive Rate, because this measure is usually used in the literature when reporting results on `Caltech101` and `RandCaltech101`.

A clustering-based classifier The clustering-based evaluation task is inspired from the unsupervised information retrieval field and it is based on clustering. A learning set of the image collection is clustered into a number of clusters and each cluster is assigned a label, using a majority vote. Each image in the test corpus is assigned to its nearest centroid and it is given the predicted label of the cluster. Predicted labels are compared to the real labels and classical information retrieval measures (*i.e.*, precision, recall, F_{score}) are calculated.

The evaluation of the clustering-based classifier is performed using a stratified holdout strategy. The images are divided into a learning corpus (67% of images in each category) and a test corpus (33% of the images in each category). 50% of images in the learning corpus are randomly selected to be part of the labeled set $\mathcal{P}_1$. For the rest, the labels are hidden. Images in the learning set are then clustered into nc clusters using K-Means. nc varies between 50 and 1000 (step 50) for `Caltech101` and `RandCaltech101` and between 3 and 90 (step 3) for `Caltech101-3` (`Caltech101-3` contains only 3 classes, see Section 5.4.3). To eliminate the effect of disequilibrium between class sizes, we calculate and report the non-weighted averages over tags of these indicators. To measure the classification accuracy, we use the F_{score} (the harmonic average of precision and recall), a classical Information Retrieval measure. For each combination (vocabulary dimension, nc, vocabulary algorithm), the clustering and prevision phase is repeated 25 times, to eliminate the influence of the random initialization of the K-Means in the clustering-based classifier.

5.4.3 Datasets

Experiments were performed on the `Caltech101` [Fei-Fei *et al.* 2007] and `RandCaltech101` [Kinnunen *et al.* 2010] datasets. `Caltech101` contains 9144 images, most of them in medium resolution (300×300 pixels). It is a heterogeneous dataset, having 101 object categories and one reserve. Each category class is considered to be a label. Spatial positioning of objects is not used, therefore positioning ourselves in a weakly supervised context. Some authors argue that `Caltech101` is not diverse enough and that backgrounds often provide more information than the objects themselves. `RandCaltech101` is obtained from `Caltech101` by randomly modifying the backgrounds and the posture (position, orientation) of objects. It has been shown [Kinnunen *et al.* 2010] that classification is more

challenging on `RandCaltech101` than on `Caltech101`.

Because `Caltech101` is an unbalanced dataset, with category sizes ranging from 31 to 800 images, we have taken 3 out of the biggest categories (*airplanes*, *Motorbikes* and *Faces_easy*) and created another corpus, denoted `Caltech101-3`. It contains 2033 images. The advantage of the new corpus is that it provides many examples for each category and it is balanced category-wise. This allows us to study how our propositions behave on both balanced and unbalanced datasets.

5.4.4 Qualitative evaluation

Figure 5.6 – Example of images from "easy" classes (top row) and "difficult" classes (bottom row)

In a classification tasks, some classes are naturally easier to recognize than others. This happens when the numerical description is better adapted to translate them into a separable numerical space. On `Caltech101`, the best classification scores are almost invariably obtained by the same categories, independent of the choice of visual construction algorithms or parameters.

Figure 5.6 shows some examples of images belonging to "easy classes", categories that obtain good classification scores (on the upper row), and examples of "difficult classes", categories that obtain low scores (on the bottom row). The objects belonging to the "easy classes" either appear in the same posture in all examples or they have a specific color pattern that makes them easily recognisable. Most of the examples of *airplanes* and *garfield* appear with the same shape, size and orientation. Other categories like *yin_yang*, *soccer_ball* or *dalmatian* have a specific white-black alternation pattern, which makes them easily recognizable even in the real world. By contrast, the objects depicted in picture of "difficult classes", like *seahorse* or *butterfly* appear in different colors, multiple postures and sometimes hidden in the background.

We perform the same analysis on `RandCaltech101`. Table 5.1 presents a comparative view of "easy classes" and "difficult classes" constructed for `Caltech101` and `RandCaltech101`, with the non-identical categories (between the two datasets) printed in boldface. We observe the high degree of overlapping of the constructed sets: most of the "easy classes" in `Caltech101` also appear as "easily" recognizable for `RandCaltech101`. Similarly, difficult classes on `Caltech101` remain difficult on `RandCaltech101`. In Table 5.1, the only category that changes difficulty is *metronome*, which is an "easy class" in `Caltech101`

Table 5.1 – "Easy" classes and "difficult" classes in `Caltech101` and `RandCaltech101`

"Easy" classes		**"Difficult" classes**	
Caltech101	*RandCaltech101*	*Caltech101*	*RandCaltech101*
airplanes	**accordion**	**beaver**	**bass**
car_side	*airplanes*	*buddha*	**binocular**
dalmatian	*car_side*	*butterfly*	**brontosaurus**
dollar_bill	*dalmatian*	**ceiling_fan**	*buddha*
Faces_easy	*dollar_bill*	**cougar_body**	*butterfly*
garfield	*Faces_easy*	*crab*	*crab*
grand_piano	*garfield*	*crayfish*	*crayfish*
Leopards	**laptop**	*cup*	**crocodile**
metronome	*Motorbikes*	*dragonfly*	*cup*
Motorbikes	*panda*	*ewer*	*dragonfly*
panda	*snoopy*	**ferry**	*ewer*
scissors	*soccer_ball*	*flamingo*	*flamingo*
snoopy	*stop_sign*	*flamingo_head*	*flamingo_head*
soccer_ball	*watch*	*ibis*	**gerenuk**
stop_sign	*windsor_chair*	*kangaroo*	**helicopter**
tick	*yin_yang*	*lamp*	*ibis*
watch		*lobster*	*kangaroo*
windsor_chair		*mandolin*	*lamp*
yin_yang		*mayfly*	*lobster*
		minaret	*mandolin*
		pigeon	*mayfly*
		platypus	**metronome**
		pyramid	*minaret*
		rhino	**okapi**
		saxophone	*pigeon*
		schooner	*platypus*
		sea_horse	*saxophone*
		stapler	*sea_horse*
		strawberry	*stapler*
		wild_cat	*wrench*
		wrench	

and a "difficult class" in `RandCaltech101`. This proves that the background randomization performed in order to create `RandCaltech101`, while it makes the dataset more challenging to classify as a whole, does not change the relative difficulty between categories. Categories that obtain good classification scores for `Caltech101` also obtain good scores for `RandCaltech101`.

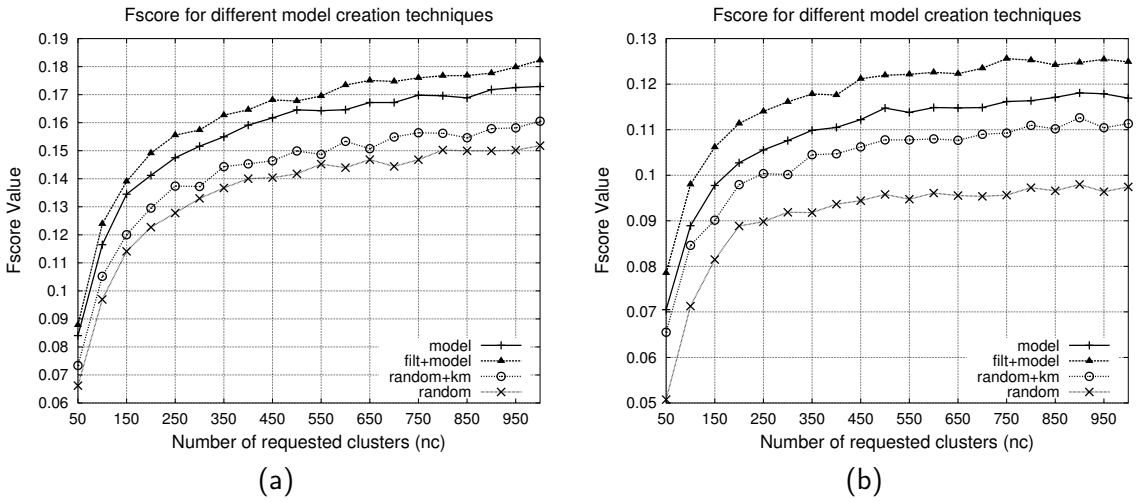


Figure 5.7 – A typical F_{score} evolution for the clustering-based classifier for $m = 1000$ on `Caltech101` (a) and on `RandCaltech101` (b)

5.4.5 Quantitative evaluation

In this section, we show how the performances of the two classifiers vary, depending on the visual vocabulary construction technique and the size of the visual vocabulary. We show that the semantically-enriched representation clearly outperform the baseline approach, mostly by increasing the score of "difficult" categories, and we discuss the overfitting. For all the experiments presented in this subsection, the parameter α (introduced in Equation 5.1, p. 108) of the filtering heuristic **filt+model** is set at one ($\alpha = 1$) and its influence is studied later in Section 5.4.7.

Aggregating the number of clusters in the *clustering-based classifier* When using the clustering-based classification algorithm, for a fixed visual vocabulary size, varying the number of clusters nc leads to an F_{score} variation as shown in Figure 5.7. For all visual vocabulary techniques, the F_{score} has a steep amelioration for lower values of nc and stabilizes once nc reaches a value which is approximately two-three times bigger than the number of categories. Starting from this point F_{score} augments slowly and reaches its theoretical maximum when nc equals the number of individuals in the testing set. Due to the fact that once stabilized, the score can be considered relatively constant, we compute the mean F_{score} over all the values for nc. We obtain, for each visual vocabulary dimension, an aggregated F_{score}.

Obtained graphics Figures 5.8, 5.9 and 5.10 present the score evolution as a function of the visual vocabulary size on, respectively, the datasets `Caltech101`, `Caltech101-3` and `RandCaltech101`. More precisely, Figures 5.8a, 5.9a and 5.10a show the evolution of the aggregated F_{score}, for the *clustering-based classifier*, and Figures 5.8b, 5.9b and 5.10b show the variation of the $TruePositiveRate$, using the SVM classifier.

We make vary the vocabulary dimension between 100 and 5300 for `Caltech101` and

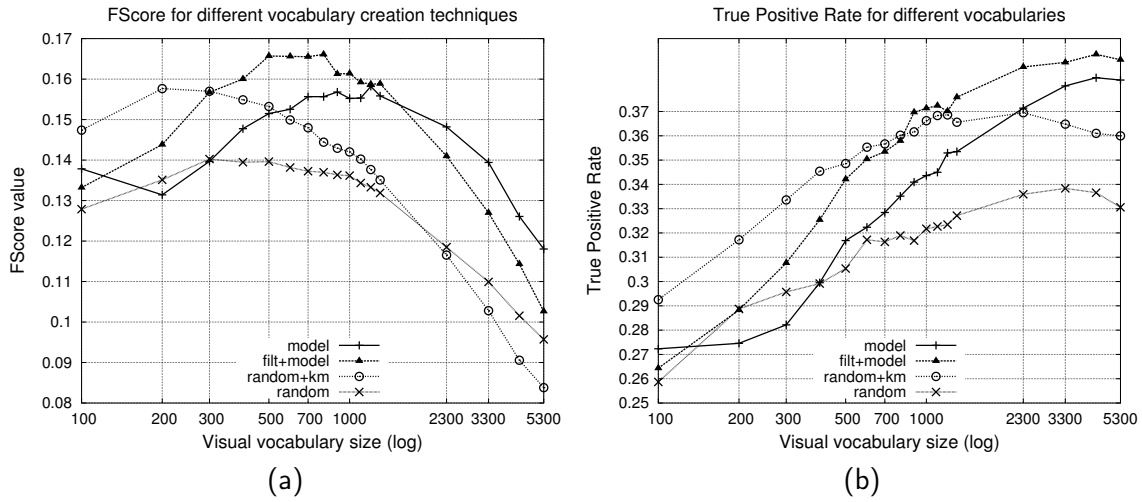


Figure 5.8 – `Caltech101`: Aggregated F_{score} with clustering-based classifier (a) and $TruePosiviteRate$ for SVM (b) as functions of the vocabulary size

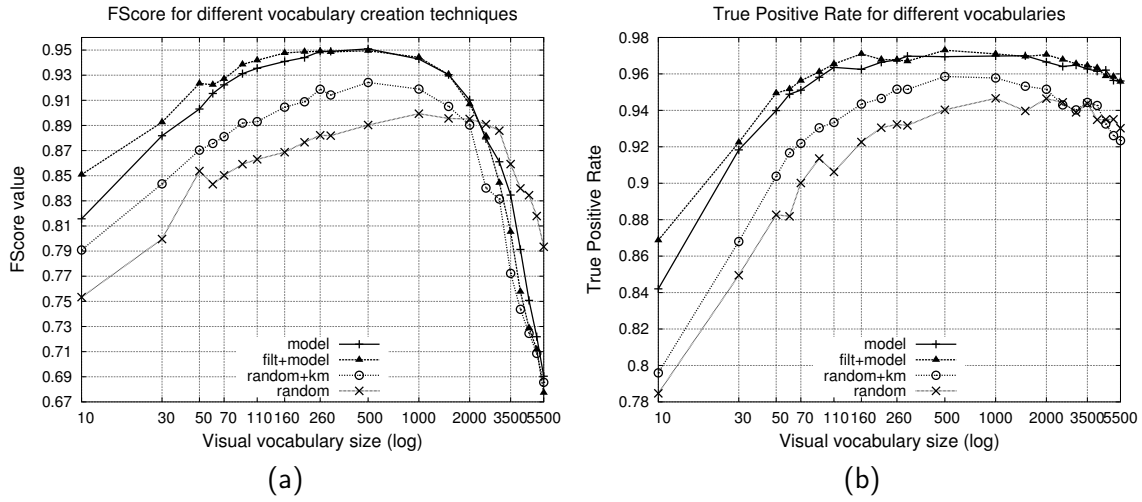


Figure 5.9 – `Caltech101-3`: Aggregated F_{score} with clustering-based classifier (a) and $TruePosiviteRate$ for SVM (b) as functions of the vocabulary size

`RandCaltech101` and between 10 and 5500 for the `Caltech101-3`, using a variable step. For the three datasets, the horizontal axis is logarithmic. When observing the graphics for every tuple *(dataset, classifier, vocabulary construction technique)*, we observe the pattern of a dome-like shape, corresponding to the three phases: under-fitting, maximum performance and overfitting. We analyze more in detail the overfitting behavior for each vocabulary construction technique in Section 5.4.6. Furthermore, the somehow low results obtained by the clustering-based classifier can be explained by the fact that the clustering-based classifier is a weak classifier (*i.e.*, a classifier which perform only slightly better than a random classifier), whereas the SVM is a strong classifier.

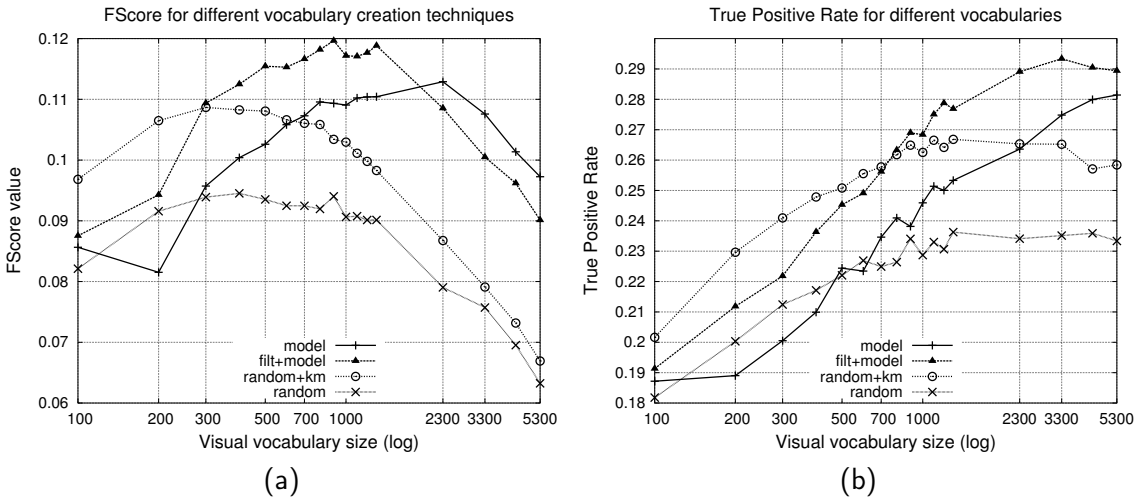


Figure 5.10 – `RandCaltech101`: Aggregated F_{score} with clustering-based classifier (a) and $TruePosiviteRate$ for SVM (b) as functions of the vocabulary size

Results interpretation When comparing the relative performances of the different techniques presented in Figures 5.8, 5.9 and 5.10, we observe that our semantic-aware proposals (**model** and **filt+model**) generally obtain better results than the generic (**random+km**) and **random** ones. The three regions of evolution are wider (they enter overfitting later) for **model** and **filt+model** than for **random** and **random+km**. On the other hand, they also exit the under-fitting later. The generic **random+km** obtains better results than **model** and **filt+model**, for lower dimensions of visual vocabulary, on `Caltech101` and `RandCaltech101`. After exiting the under-fitting region, **model** and **filt+model** constantly obtain better scores than **random+km**, even when overfitted. Applying our filtering proposal (**filt+model**) consistently provides a plus of performance (over **model**), but also causes the visual vocabulary to enter overfitting earlier.

Table 5.2 – Average gain of performance relative to **random**.

		model	**filt+model**	**random+km**
clust.	`Caltech101`	13.96%	**15,69%**	4,36%
	`Caltech101-3`	6.58%	**7.36%**	2.73%
	`RandCaltech101`	20,49%	**26,27%**	12,07%
SVM	`Caltech101`	5,98%	12,02%	**12,05%**
	`Caltech101-3`	4,71%	**5.24%**	1,90%
	`RandCaltech101`	5,89%	**15,20%**	13,21%

Table 5.2 gives the average gain of performance relative to **random** for the generic **random+km** and our semantic-aware proposals **model** and **filt+model**. For the clustering-based classifier, we show the average relative F_{score} gain, while for the SVM we show the average relative $TruePositiveRate$ gain. The best scores for each dataset are shown in bold. In five out of six cases, the best scores are obtained by **filt+model**. **model** also

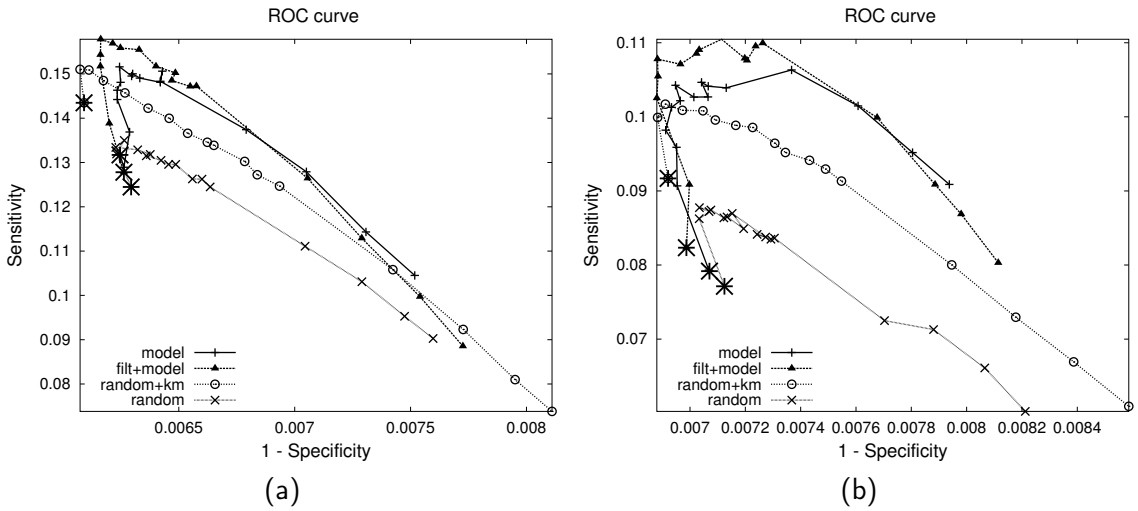


Figure 5.11 – ROC curves: clustering-based classifier on `Caltech101` (a) and `RandCaltech101` (b)

performs better than the generic **random+km** in four out of the six cases. This shows that a semantically-enriched representation outperforms the generic method **random+km** in a classification task. The maximum gain of performance is achieved on `RandCaltech101`, where, by eliminating the background noise, our filtering algorithm considerably improves the classification performances. When used with the SVM classifier on `Caltech101` and `RandCaltech101`, the **model** technique obtains average scores lower than **random+km**. This is because **model** exits the under-fitting later than the other techniques, thus lowering its average score (as shown in Figures 5.8b and 5.10b).

The ROC curves Similar conclusions regarding the overfitting and the relative performances of the different visual vocabulary construction techniques can be drawn by plotting the evolution using *ROC* [Fawcett 2006] curves. Figure 5.11 shows the ROC curves obtained using the clustering-based classifier on `Caltech101`(Figure 5.11a) and on `RandCaltech101` (Figure 5.11b). The visual vocabulary size varied between 100 and 5300. The sign * on the graphic indicates the smallest size. The plots are zoomed to the relevant part. Overfitting is clearly visible on the ROC curves. All the curves start by climbing towards the ideal point $(0, 1)$ (first and second region on the graphics in Figures 5.8a and 5.10a). After reaching a maximum, the ROC curves start descending towards the "worst" point $(1, 0)$, showing the overfitting region. The curve corresponding to **filt+model** clearly dominates all the other, confirming the conclusions drawn from studying Table 5.2: the proposed approaches and especially their combination in **filt+model**, achieve higher classification results.

Scores for "easy" and "difficult" categories In Section 5.4.4, we have shown that in both `Caltech101` and `RandCaltech101` some classes are easier to learn than others. Regardless of the visual vocabulary construction technique, "easy classes" obtain higher classification scores. Nonetheless, the construction particularities of each technique influence the accuracy for difficult categories. In **random**, features are randomly picked to serve as

visual words. Score differences between easy and difficult categories are pronounced and the overall accuracy is low. The K-Means iterations in **random+km** fit the visual vocabulary to "easy" classes. Few categories achieve good scores, accentuating the gap between easy and difficult categories. **model** and **filt+model** techniques achieve for "difficult" categories, better scores than **random** and **random+km**. The visual vocabulary is representative for all categories and difficult categories like *pyramid*, *minaret* or *stapler* obtain higher scores than those obtained with a baseline representation.

5.4.6 Overfitting

Evaluating using the clustering-based classifier In the clustering-based classifier, for each pair (dataset, vocabulary construction technique), the F_{score} graphic shows a dome-like shape with three regions. In the first one, corresponding to low vocabulary dimensions, the visual vocabulary is under-fitted, there are not enough visual words to describe the objects [Jiang *et al.* 2007]. Consequently, in the *assign phase* (*phase 4* in "bag-of-features" construction schema in Figure 5.2), features are assigned to the same visual word even if they are not similar to each other. The second region represents the interval in which the vocabulary obtains the best results. In the third region (corresponding to large sizes of the visual vocabulary), performance degrades gradually. This is due to the fact that, in the *assign phase*, relevant features are grouped densely, while noise is evenly distributed. Some of the visual words regroup relevant features, while other regroup only the noise. As the visual vocabulary dimension augments, more and more visual words will regroup only noise. This generates a numerical space of high dimensionality, which is separable only on a few dimension. This leads to degrading the overall separability of the numerical space and the classification performances.

Evaluating using the SVM classifier The same conclusions apply for the SVM classifier. Being a strong classifier, in Figures 5.8b (`Caltech101`) and 5.10b (`RandCaltech101`) the dome-shape is less visible for the SVM. The overfitting appears for higher visual vocabulary sizes than in the clustering-based classifier. For example, in Figure 5.10a, for **random+km**, clustering-based classifier starts to overfit at a vocabulary size of 300. When using the SVM, in Figure 5.10b, overfitting starts only at 1300. The **model** technique does not appear to enter overfitting in Figure 5.10b. But this is likely to happen for dimensions higher than 5300 (the maximum considered), because **model** is the last technique to enter overfitting for the clustering-based classifier (as shown in Figure 5.10a).

The overfitting region is even more visible for `Caltech101-3` (Figure 5.9). The visual vocabulary sizes are considerably higher than for the other datasets, relative to the number of classes. In Figure 5.9a performances of all visual vocabulary techniques descend sharply for higher values of vocabulary size. The evaluation using the SVM classifier, in Figure 5.9b, also clearly shows the dome-like shape.

5.4.7 Influence of parameter α

In Equation 5.1 (p. 108), we have defined δ, the filtering threshold, which is used to decide if a feature has any similar features in the known positive set. The parameter α is

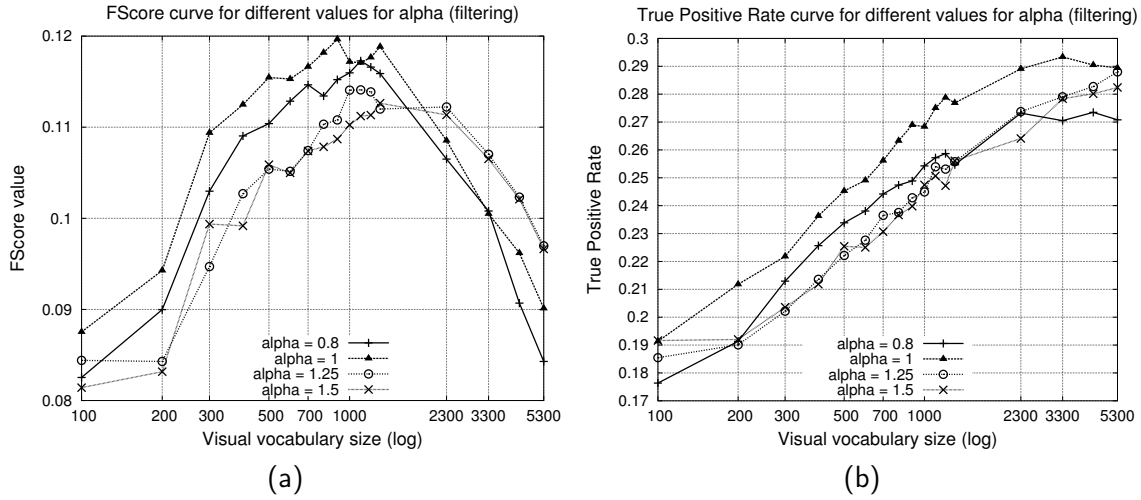


Figure 5.12 – `RandCaltech101`: influence of parameter α on **filt+model** construction technique in the clustering-based task (a) and the SVM classifier (b)

used to fine-tune this threshold. If α is set too low, only the features that are very close (in terms of Euclidean distance) are considered to be similar. Consequently, the filtering is very strict, lowering the number of *false positives*, with the risk of an inflation of *false negatives*. On the other hand, setting α too high allows distant features to be considered as similar, causing a high number of *false positives*. In the previous experiments, we have set the parameter $\alpha = 1$. In this section, we study the influence of this parameter on the performances obtained by the **filt+model** construction technique.

Figure 5.12 shows the evolution of the **filt+model** visual vocabulary construction technique as a function of visual vocabulary size, when using $\alpha \in \{0.8, 1, 1.25, 1.5\}$. The horizontal axis is logarithmic. A value for $\alpha = 0.8$ is too strict and the high number of *false negatives* decreases classification performances. Augmenting $\alpha = 1$ improves performances, both when using the clustering-based classifier (Figure 5.12a) and when using the SVM classifier (Figure 5.12b).

If α is set too high, performances decrease again. Too many features are considered similar and less features get filtered. Performances approach those obtained when no filtering is applied. $\alpha = 1.25$ and $\alpha = 1.5$ show similar performances, since both levels are already too high for filtering to be effective. For $\alpha \geq 1.25$ **filt+model** is equivalent to the **model** visual vocabulary construction technique. In Figure 5.12a, **filt+model** with $\alpha \in \{1.25, 1.5\}$ obtains, for high visual vocabulary sizes ($m > 2000$), better results than **filt+model** with $\alpha \in \{0.8, 1\}$. This behaviour is similar with that already seen in Figure 5.10a, when **model** enters overfitting later than **filt+model** and obtains better results for high vocabulary sizes.

These initial experiments make us believe that α is dataset independent (a value of 1 provided best results on all three datasets), but further experiments on other datasets are required in order to conclude this. Furthermore, a heuristic for automatically determining its value is part of our future plans.

5.5 Conclusions and future work

Conclusion In this chapter, we have focused on one of the core research challenges of this thesis: **leveraging semantics when dealing with complex data**. More precisely, we are interested in constructing a semantically-enriched image representation, by leveraging additional information under the form of non-positional image labels. We argue that enriching the semantics of the image representation would boost the performances of learning algorithms and we apply our proposed method to the learning task of content-based image classification.

We use the additional information in the phase of visual vocabulary construction, when building a "bag-of-features" image representation. We have proposed two novel approaches for incorporating this semantic knowledge into the visual vocabulary creation. The first approach creates dedicated vocabularies for each label, while the second uses a pre-processing phase for filtering visual features unlikely to be associated with a given object. We have shown that the semantically-enriched image representation built using our proposals obtain higher scores than a baseline *BoF* representation, in the context of a task of content-based image classification. This shows that incorporating semantic knowledge in the vocabulary construction results in more descriptive visual words, especially on datasets where the background noise is significant. Even when overfitted, our proposals continue to outperform the generic approach.

Passage to semantic scene classification and label co-occurrence Passing from object categorization to scene classification raises the difficulty of object co-occurrence. For example, a picnic scene is defined by the simultaneous presence of *"people", "trees", "grass"* and *"food"*. In terms of labels, this translates in label co-occurrence. In the approaches proposed in Section 5.3, we assume that labels which denote objects appear independently. The label correlation in complex scenes can be difficult when using the keypoint filtering proposal together with the dedicated vocabulary proposal (the **filt+model** vocabulary construction technique). For the filtering, the known positive set contains images labeled identically with the target image. Therefore, in the case of scene classification, the filtering is applied for a set of labels. But the dedicated model is constructed for each label individually. This inconsistency does not necessarily pose problems: filtering the keypoints for the labels $\{t_1, t_2, t_3\}$ and constructing the vocabulary only for label t_1 does not produce erroneous results, only noisy results.

Our approaches can be scaled to image classification by addressing the label co-occurrence issue. For this purpose, we can adapt the label set to the image collection by reconstructing labels to reduce, even eliminate, their co-occurrence. The feature [7] construction technique presented in Chapter 4 can be used to construct a new label set that properly describes the image collection. The new labels are constructed as conjunctions of existing labels and their negations, and would actually no longer be used to label objects, but scenes. For example, if the labels *"motorcycle"* and *"rider"* appear often together, a new label *"motorcycle* $\wedge$ *rider"* will be created to mark the scene identified by the presence of the two

7. Note that, in this context, the word **feature** is used in the sense defined in Chapter 4, synonym to attribute. It should not be confused with the definition of **feature** in image processing literature, where it has the sense of visual feature.

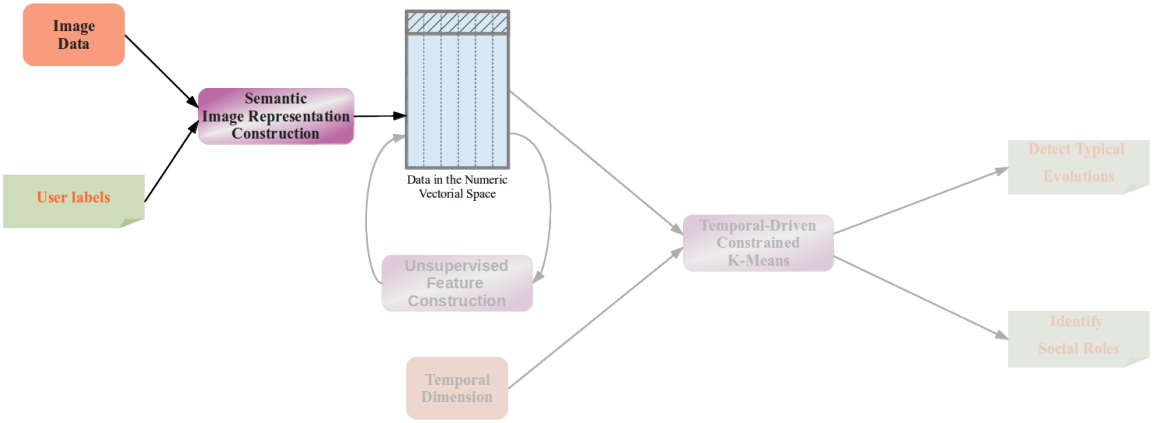


Figure 5.13 – Streamlined schema showing how the contributions in this chapter can be articulated with those in previous chapters.

objects.

Articulation with the previous work Conceptually, the work presented in this chapter articulates with the work of previous chapters as shown in Figure 5.13. The work presented in previous chapters is presented with faded colors. The joining point is, as in the case of the previous chapters, the description of the data in a semantic-aware numeric description space. Using the "bag-of-features" format, presented in this chapter, an image can be described as a multidimensional vector of distributions over visual words. Furthermore, the work presented in Chapter 4 has a direct application in label re-organizing for scene classification, as described in the previous paragraph.

Most of the work presented in this chapter was submitted and is under review for the **International Journal of Artificial Intelligence Tools** (IJAIT) [Rizoiu *et al.* 2013b].

Chapter 6

Dealing with text: Extracting, Labeling and Evaluating Topics

Contents

6.1 Learning task and motivations

This chapter presents our work concerning one of the most important and most abundant types of complex data: text. More specifically, we focus on topics, which provide quick means of summarizing the main "ideas" that emerge from a collection of textual documents. Topics are usually defined as statistical distributions of probabilities over words and, therefore, they are sometimes hard to interpret for human beings. The work detailed in this chapter, and presented schematically in Figure 6.1, approaches the research challenge of **leveraging semantics when analyzing textual data**.

This core research challenge is materialized into three learning tasks, associated with two of the guidelines of this thesis: (i) extracting topics, (ii) labeling topics with humanly comprehensible labels (this task is related to the guideline of creating humanly comprehensible outputs) and (ii) using semantic knowledge, under the form of concept hierarchies, in topic evaluation (this task is related to the guideline of devising methods that embed semantics

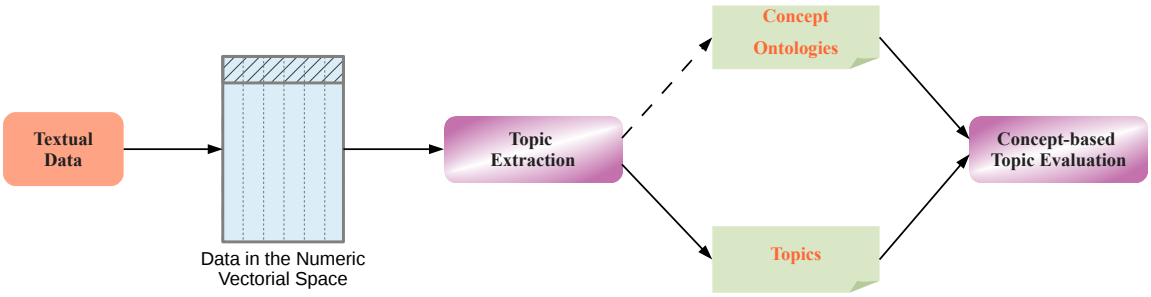


Figure 6.1 – Streamlined schema of our work with the textual dimension of the complex data: extracting and evaluating topics.

in the learning process). For the first task, we propose a solution for topic extraction using overlapping text clustering, which allows a document to be assigned to multiple topics, based on its semantics. For the second task, we assign to topics humanly comprehensible labels by using an approach based of suffix arrays. For the last task, we propose to map topics to subtrees of concepts associated to a knowledge base (here, WordNet [Miller 1995]), by passing through words. The semantic cohesion of topics is evaluated based on the height and depth of the corresponding topical subtrees in the concept hierarchy.

Practical and applied context Unlike the work presented in previous chapters, for which the motivations were mainly research driven, the work presented in this chapter has a very close connection to my applied work and the various research projects in which I was involved along my thesis. While not being central to my research work (for which the accent lies on semantic representation and the temporal dimension), the work concerning text analysis through topic extraction and evaluation constantly doubled it.

Topic extraction is an important tool, especially in the context of applied projects, in which textual analysis is involved. The topic extraction system described in Section 6.4.1 was implemented [1] into the CKP [2] topic extraction platform, in the context of the project ConverSession, in which the creation of a start-up enterprise [3] was involved (see Annex A). The object of the project was Online Media Watching and the development of the resulted prototype was later continued in the context of the projects Eric-Elico, Crtt-Eric and ImagiWeb. All these projects shared the need to retrieve text from the Internet, usually from forum online discussions, and analyze it from various points of view (*e.g.*, detect the discussion topics, their emergence and evolution, differences of discourse *etc.*). The software resulted from this continuous development is `CommentWatcher`, an open-source web-based platform for analyzing online discussions on forums. `CommentWatcher` will be described in detail in Chapter 7.

Multiple publications also resulted from the context of these projects and collaborations. Some of the works concerning topic extraction were initially developed during my Masters research internship. The further extensions were published in the proceedings of a French

1. The first beta version was implemented during my Master's thesis.
2. Download here: `http://eric.univ-lyon2.fr/~arizoiu/files/CKP-src.jar`
3. `http://www.conversationnel.fr/`

national conference [Rizoiu *et al.* 2010], while the application of topic extraction to ontology learning was published in a book chapter [Rizoiu & Velcin 2011]. The topic evaluation methodology was developed in collaboration with the Computer Science department of the Polytechnic University of Bucharest, and more precisely, the PhD research internship of Claudiu Cristian Muşat at the ERIC laboratory. It was proposed in the proceedings of an international conference [Musat *et al.* 2011b]. The application of outlier detection was also published in the proceedings of an international conference [Musat *et al.* 2011a].

Research context and motivations When automatically analyzing text, difficulties arise, similar to those raised by image data. The textual native digital representation (which handles encoding at the level of character) usually captures little information about the semantics of the text. We define the semantics of a text as the information that is transmitted though the text, its intent. To correctly structure text, most languages possess syntactic and morphological rules, which were employed by the Natural Language Processing research community to develop automatic language processing systems. These systems usually provide good accuracy results, but they are costly to develop and maintain (due to the expert supervised component), they are specialized on particular domains and often fail to capture the subtleties of language (*e.g.*, humor, irony, complex opinions). Other, more statistically-oriented approaches, were proposed. The most widely used textual representation employed by numeric systems, is the orderless "bag-of-words" representation, presented in detail in Section 6.2. Among these numeric approaches, topic extraction systems emerged as a response to the need to summarize large quantities of textual data and infer the main "ideas" behind a collection of textual documents.

The work presented in this chapter deals with **leveraging semantic information into the tasks of topic extraction and topic evaluation**. We also present some of their applications, such as ontology learning and topic improvement. We propose a topic extraction system that infers topics from text, by means of overlapping text clustering. We address the problem of human comprehension of topics by assigning each cluster a "humanly-comprehensible" name, made out of a frequent complete expression. As one of the guidelines of our work, we consider crucial to generate humanly-comprehensible outputs, since topics defined as distributions of probabilities over words are difficult to interpret for a human being. For the topic evaluation task, the idea is to emulate the human judgment of topics. To this end, we use a semantic resource, such as a concept hierarchy (*e.g.* WordNet). Each topic is mapped to a subtree in the concept hierarchy and we redefine the specificity and coverage of the subtree based on its height and depth in the concept hierarchy. We use this measures to evaluate the cohesion of topics.

The remainder of this chapter is structured as follows. In Section 6.2, we present how raw text can be translated into a numerical format using the "bag-of-words" representation. An overview of research related to topic extraction, labeling and evaluation will be presented in Section 6.3. In Section 6.4, we propose a topic extraction system and concept-based topic evaluation system. We continue in Section 6.5 by presenting two applications of our proposals: topic extraction for ontology learning and the improvement of topic, based on concepts, by removing the topic's spurious words. We end with Section 6.6, in which we draw the conclusions and plan some future work concerning the textual dimension of the complex data.

6.2 Transforming text into numerical format

One of the simplest text numerical representations is the "bag-of-words" representation [Harris 1954, Harris 1968] (BoW). This representation is extensively used in Natural Language Processing and Information Retrieval tasks. Lately, this representation has also been used in Image Processing, as shown in Chapter 5, under the form of "bag-of-features" representation. In the BoW model, text is represented as an unordered collection of words, disregarding grammar and even word order. Each word is assigned a score, calculated according to a term weighting scheme.

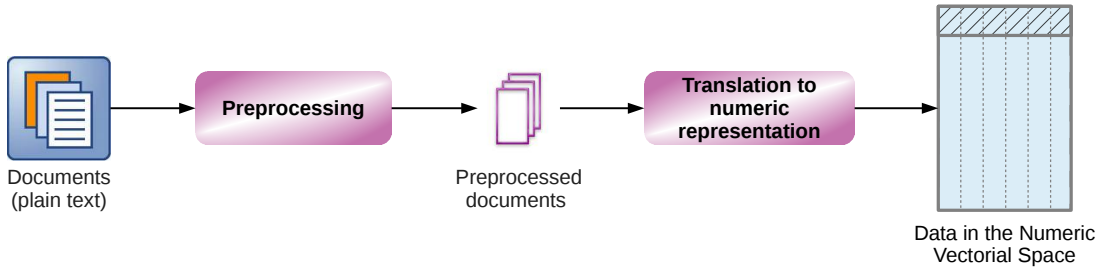


Figure 6.2 – Schema of transforming texts to a numerical representation.

Figure 6.2 shows the typical treatment chain that allows transforming textual data into the tabular numeric format. In the **Preprocessing** phase, each document is preprocessed, as shown in Section 6.2.1. This usually includes (i) eliminating common words that do not bring any information about the thematic of the text and (ii) stemming or lemmatizing inflected words, in order to increase their descriptive value. In the **Translation to numeric representation** phase, the preprocessed documents are translated into the a numeric vectorial description space, as shown in Section 6.2.2, using one of the term weighting schemes (*e.g.*, term frequency, TFxIDF *etc.*). At the end of this phase, the textual collection can be represented as a **term/document matrix**. Each numeric feature (the columns) corresponds to a word in the vocabulary and each vector (the rows) defined in this multidimensional space corresponds to a textual document. We present in further detail each of these two phases over the next two subsections.

6.2.1 Preprocessing

The purpose of preprocessing is to augment the descriptive power of the terms and limit the size of the vocabulary. There are typically two problems that arise when dealing with natural language text using a BoW approach: *inflected/derived words* and *stopwords*. Therefore, the preprocessing is usually composed of two elements: *stemming/lemmatization* and *stopwords removal*.

Stemming/Lemmatization Many languages apply inflections on words to express possession, verb tenses *etc.* For example, in English, the verb "to walk" may appear as "walk", "walked", "walks", "walking". The base form (*i.e.*, "walk"), the infinitive for verbs, masculine singular form for nouns, is called the lemma for the word. While crucial for the human comprehension of the text, inflections can usually be ignored for many applications. When using

the *BoW* representation, having different forms for the same base form leads to creating multiple bags for a single word and incorrectly evaluating the word's score in the document.

Stemming is the process for reducing inflected words to their stem or root form, by removing inflections, prefixes and/or suffixes. The stem of a word is not necessarily identical to the word's lemma and it may not even be a valid word or the morphological root of the word. It is sufficient that all inflected variants of a words map to the same stem. Stemming is dependent on language, but algorithms have been developed for the majority of widely used languages. For English, the stemmer mostly used is Porter's stemmer [Porter 1980] and various initiatives (*e.g.*, the CLEF Project [4]) have proposed solutions for European languages.

Lemmatization refers to determining the lemma for a given word. This process usually involves complex tasks such as understanding context and determining the part of speech of a word in a sentence. Lemmatization is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech (*e.g.*, "meeting" as a noun in "during our last meeting" or as a verb in "we are meeting again tomorrow"). Stemmers are typically easier to implement and run faster, while lemmatization can in principle select the appropriate lemma depending on the context.

Stopwords removal Stopwords are commonly used words, such as articles, prepositions *etc.*, that do not present any descriptive value, as they are not associated to a certain thematic. When using a *BoW* representation, they increase the size of the dictionary and bias the values of certain term weighting schemes (such as Term Frequency). Stopwords are typically removed using stopword lists, which can contain short function words (*e.g.*, the, is, at, which, on etc.), but any words judged unnecessary can also be included. Whatsoever, stopwords are important for the human reader (as shown in Section 6.4.1) and their removal might render the results humanly-incomprehensible. Therefore, when results are shown to the reader (*e.g.*, cluster labels, search results), they typically include them. Our work concerning the topic labeling, presented in Section 6.4.1, uses complete expressions that also include stopwords.

6.2.2 Text numeric representation

Numerous textual representations exist in the Information Retrieval domain [Singhal 2001]. The *Boolean Model* compares True/False query statements with the word set that describes a document. Such a system has the shortcoming that there is no inherent notion of document ranking. The *Probabilistic Model* [Maron & Kuhns 1960] is based on the general principle that documents in a collection should be ranked by decreasing probability of their relevance to a query. In the *Inference Network Model* [Turtle & Croft 1989], document retrieval is modeled as an inference process in an inference network. A document instantiates a term with a certain strength, and the credit from multiple terms is accumulated given a query to compute the equivalent of a numeric score for the document.

The *Vector Space Model* [Salton *et al.* 1975] is the representation most widely used in modern text clustering algorithms and Information Retrieval tasks. Just as the *BoW* rep-

4. http://www.clef-campaign.org/

resentation, the Vector Space Model does not conserve the order of words or their semantic relations. Each document is represented as a multidimensional vector in a space having as many dimension as there are terms [5] in the considered dictionary and where each dimension is associated to a term. The score on each dimension is directly proportional to the strength of the relationship between the considered word and the document. The Vector Space Model allows to define a distance between textual documents, which can be interpreted as the similarity between documents. It is usually calculated using the **cosine distance** of the two multidimensional vectors:

$$||d_i - d_j||_{cos} = 1 - cos(\vec{d_i}, \vec{d_j}) = 1 - \frac{\sum_{k=1}^{|\mathcal{V}|} d_{i,k} \times d_{j,k}}{\sqrt{\sum_{k=1}^{|\mathcal{V}|} d_{i,k}^2}\sqrt{\sum_{k=1}^{|\mathcal{V}|} d_{j,k}^2}} \tag{6.1}$$

where d_i and d_j are two documents in the document collection $\mathcal{D}$, $\mathcal{V}$ is the word vocabulary and $d_{i,k}$ is the score of word $w_k \in \mathcal{V}$ associated to the document d_i.

There are numerous methods for measuring the scores for words, also known as **term weighting schemes** [Salton & Buckley 1988], out of which we single out (i) the presence/absence scheme, (ii) the term frequency scheme, (iii) the inverse document frequency scheme and (iv) the TFxIDF scheme, detailed here after.

1. **Presence/Absence** is also known as **binary weighting** and it is the simplest way to measure the belonging of a word to a document. Its formula is:

$$d_{i,k} = pa(d_i, w_k) = \begin{cases} 1, & \text{if the word } w_k \text{ is found in document } d_i \\ 0, & \text{otherwise} \end{cases}$$

   This weighting scheme can only show if a word is related to a document, but it does not measure the strength of the relationship [Dumais *et al.* 1998].

2. **Term Frequency** is also known as **term count**. It is defined as the number of times a given word appears in a document. To avoid favoring longer documents, normalization is usually used:

$$d_{i,k} = tf(d_i, w_k) = \frac{n_{i,k}}{\sum_{l=1}^{|\mathcal{V}|} n_{i,l}}$$

   where $n_{i,k}$ is the number of occurrences of the word w_k in the document d_i and the denominator is the total number of words in the document d_i.

3. **Inverse Document Frequency** is a measure of the general importance of a word in the whole corpus. It favors rare words, giving a low score to words that appear in many documents. IDF is defined as:

$$idf(w_k) = \log \frac{|\mathcal{D}|}{|\{d \,|\, w_k \in d\,,\, d \in \mathcal{D}\}|}$$

   where $\mathcal{D}$ is the collection of textual documents and $|\{d \,|\, w_k \in d\,,\, d \in \mathcal{D}\}|$ is the total number of documents in which the word w_k appears. In practice, IDF is never used alone, as it lacks the power to quantify the relationship between a word and a document and it favors very rare words, which often prove to be typographic errors. IDF is usually used in conjunction with TF in the TFxIDF weighting scheme.

5. A term is a word that has a certain semantic (*i.e.*, most often a term refers to objects, ideas, events, states of affair *etc.*). All terms are words, but only some words are terms.

4. **TFxIDF** [Jones 1972] is the most used scheme in Information Retrieval. It is the product of Term Frequency and Inverse Document Frequency:

$$d_{i,k} = tfxidf(d_i, w_k) = tf(d_i, w_k) \times idf(w_k)$$

TFxIDF aims at balancing local and global occurrences of a word. It assigns a high score to a word w_k that appears multiple times in a document d_i (high Term Frequency), but which is scarce in the rest of the collection (high inverse document frequency).

Other term weighting schemes for the BoW representation exist, often variations of the classical schemes. **Okapi BM25** [Robertson *et al.* 1995] is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document. It is based on the probabilistic retrieval framework. Other weighting schemes search to include additional external information into the scoring function. For example [Karkali *et al.* 2012] propose tDF (temporal Document Frequency), which is a variation of the BM25 measure. It includes temporal information in order to "keep topics fresh", by adding a decay factor under the form of a temporal penalization function.

6.3 An overview on Topic Extraction

It is generally accepted that a topic represents an "idea" that emerges from a collection of textual documents. But there is no widely accepted formal topic definition. Throughout the literature, a topic is considered to be (i) a cluster of documents that share a thematic [Cleuziou 2008], (ii) a distribution of probabilities over words and over documents [Blei *et al.* 2003] or (iii) an abstraction of the regrouped texts that needs a linguistic materialization: a word, a phrase or a sentence that summarizes the idea emerging from the texts [Osinski 2003]. Plenty of applications can benefit greatly from topic inference from text, including information retrieval systems, database summarization, ontology learning, document clustering and querying large document collections.

We present an overview of topic extraction, focusing on **flat clustering** techniques, which create a partition of documents with a single level. These algorithms divide a collection of textual documents into groups, so that documents inside a group are similar in terms of their topic (politics, economics, social *etc.*) and dissimilar with documents in other groups. For each cluster, K-Means-based clustering algorithms present a centroid, which is an abstraction of the cluster and which summarizes the common part of the documents in the cluster.

The centroid (a multidimensional vector or a distribution of probabilities) is not a real document and rarely makes any sense to a human. Therefore, it is convenient to choose a comprehensible name for it. There are a number of ways of labeling a topic [Roche 2004]: (i) choosing an arbitrary number of high rated words, (ii) selecting a document as representative, (iii) assigning a meaningful, human-readable expression or phrase *etc.* The word property of **polysemy** is important when labeling topics. The same word can have different meanings, depending on the context. For example, the word "mining" has one meaning in the expression "coal mining" and another in "data mining".

Other topic extraction methods exist, most notably those issued from computational linguistics [Ferret 2006] or graph-based methods [Ng *et al.* 2002]. In the following sections we concentrate only on statistical-based methods: we present text clustering based methods (in Section 6.3.1), probabilistic topic models (in Section 6.3.2). In Section 6.3.3, we address the issue of topic labeling, and we finish in Section 6.3.4 with a short overview of methods which deal with topic evaluation and improvement.

6.3.1 Text Clustering

We choose to divide text clustering algorithms into categories based on their ability to create **overlapping groups**. Other classifications are possible, some authors [Dermouche *et al.* 2013] divide them into families of methods: (i) distance-based methods, (ii) matrix factorization-based methods and (iii) probabilistic methods. Some of the solutions presented below were designed specifically for text mining (like LDA), others are general purpose clustering algorithms. The latter can be used for text clustering by translating textual documents into the Vector Space Model (described in section 6.2.2).

Crisp solutions

Crisp clustering algorithms regroup a collection of documents into a partition of disjointed classess. **K-Means** [MacQueen 1967] is one of the most well-known crisp clustering algorithms. The algorithm iteratively optimizes an objective criterion, typically the distortion function. In the case of text mining and information retrieval, the cosine distance is used to calculate the similarity between texts. **Bisecting K-Means** [Steinbach *et al.* 2000] is a hierarchical variant of K-Means, which has been emphasized as more accurate than K-Means for text clustering. It is a top-down algorithm that partitions, at each step, one cluster of documents into two crisp sub-clusters. At the first iteration, the collection is divided into two subsets according to multiple restarting 2-means. At the successive iterations, one subset is split into two and $n+1$ text clusters are obtained from n initial clusters. The process is iterated until a stopping criterion is satisfied (*e.g.*, a fixed number of clusters). The final output of Bisecting K-Means can be seen as a truncated dendrogram. **Hierarchical agglomerative clustering (HAC)** [Jain & Dubes 1988] is a hierarchical clustering technique used more frequently in Information Retrieval. It constructs the hierarchy bottom-up and consists in merging at each step pairs of clusters. A single-level crisp partition can be obtained by cutting the dendogram at certain level, chosen using a given heuristic (*e.g.*, biggest gap between levels). Whatsoever, hierarchical clustering methods are ill adapted to treat the great volumes of data encountered in text mining.

Overlapping solutions

In overlapping clustering, documents are authorized to simultaneously be part of multiple clusters. The result is no longer a strict partition of the document collection, since groups have non-empty intersections. Considering that longer texts have the tendency of approaching multiple subjects, it is only natural to allow them to be part of each of the corresponding topics. (*e.g.*, a text that talks about the "economical outcomes of a political

decision" should be part of both the "politics" and the "economics" group). Therefore, an overlapping technique seems more appropriate for text clustering [Cleuziou *et al.* 2004].

Overlapping K-Means (OKM) [Cleuziou 2008] is an extension of **K-Means**. It shares the same general outline, trying to iteratively minimize an objective function. In OKM, a document can be assigned to multiple clusters. Whereas in K-Means, each document is assigned to the closest centroid, OKM assigns documents to the closest *Gravity Center* of multiple centroids. The objective function minimized by OKM is the distortion in the dataset:

$$distorsion\,(\mathcal{D}) = \frac{1}{k \times |\mathcal{D}|} \sum_{d_i \in \mathcal{D}} ||d_i - \bar{d}_i||^2 \tag{6.2}$$

where $||\bullet||$ is the considered distance between documents (usually the cosine distance $||\bullet||_{cos}$ defined in Equation 6.1), k is the number of clusters, $\mathcal{D}$ is the document collection and $\bar{d}_i$ is the image of document d_i ($\bar{d}_i$ is the gravity center of clusters to which d_i is assigned). wOKM [Cleuziou 2009] is a weighted version of OKM, which uses weights internally to limit the overlapping of clusters. wOKM can be seen as a special case of subspace clustering [Parsons *et al.* 2004].

Fuzzy solutions

In fuzzy clustering, each document belongs with a certain degree (or probability) to all clusters, rather than belonging completely to just one cluster (in crisp clustering) or several clusters (in overlapping clustering). Each document d is associated with the cluster μ_l with the degree $u(d, \mu_l)$. A document d_i situated at the edge of a cluster μ_l is associated with the cluster in a lower degree than a central document d_j:

$$u(d_i, \mu_l) < u(d_j, \mu_l), \forall d_i, d_j \in \mathcal{D} \text{ and } ||d_i - \mu_l||^2 < ||d_j - \mu_l||^2$$

Fuzzy logic clustering algorithms can be adapted to output a crisp partition by selecting for each document d, the cluster with the highest degree of belonging:

$$\textbf{chosen\_cluster}(d) = \underset{l=1,2,...,k}{argmax}\,(u(d, \mu_l))$$

and to output an overlapping partition by choosing a threshold θ and considering only clusters for which the degree of belonging is greater than the threshold:

$$\textbf{chosen\_clusters}(d) = \{\mu_l \,|\, u(d, \mu_l) > \theta,\, l = 1, 2, ..., k\}$$

Fuzzy K-Means [Dunn 1973] is an adaptation of the K-Means algorithm to the fuzzy logic. Fuzzy K-Means differs from the original version in several aspects: the way the objective function is calculated, the centroid update phase and the output of the algorithm. Every pair (document, cluster) contributes to the objective function proportionally to the membership degree of the documents in the cluster. Similarly, in the centroid update phase, all documents contribute accordingly to their weights. The output of the algorithm is, for each document, a vector with the probabilities of membership to clusters.

Latent Semantic Indexing (LSI) [Berry *et al.* 1995] is a statistical topic discovery algorithm using Singular Value Decomposition (SVD) as the underlying mathematical ground.

The algorithm decomposes the term/document matrix A in a product of three matrices: $A = USV^T$. U and V are orthogonal matrices, containing the left and the right singular vectors of A. S is a diagonal matrix with the singular values of A ordered decreasingly.

LSI allows defining heuristics for reducing dimensionality and determining the number of topics. A_k, the k-approximation of A ($A_k = U_k S_k V_k^T$) is obtained by selecting the k highest ranking singular values from S, alongside with the corresponding columns in U and the lines in V. There exist heuristic for determining k, for example, (i) to order decreasingly the singular values of A and cut at the highest difference between two consecutive values [Osinski 2003] or (ii) the scree test [Cattell 1966]. The columns in U corresponding to the k highest singular values create an orthogonal basis for the document space. Any multidimensional vector in this space can be expressed as a weighted sum of the elements of the base:

$$d_i = \alpha_1\mu_1 + \alpha_2\mu_2 + \ \ldots\ + \alpha_k\mu_k\,,$$
$$i \in 1, 2, \ldots, |\mathcal{D}| \ \text{ and } l \in 1, 2, \ldots, k$$

Considering the base elements μ_l as the centroids of clusters, documents are described in a fuzzy logic: the document d_i has the probability α_l of belonging to the cluster μ_l. LSI has the inconvenience that it can produce negative singular values, which can pose interpretability problems [Lee & Seung 1999]. Non-negative Matrix Factorization [Lee & Seung 1999] (NMF) deals with this problem by finding a non-unique factorization of the non-negative matrix V, so that $V = WH$. NMF-based methods have been shows efficient in topic extraction [Seung & Lee 2001, Xu *et al.* 2003].

6.3.2 Topic Models

Many concurrent solutions exist for topic extraction, some of which that fall in the category of text clustering have been presented in the Section 6.3.1. However, in recent years, generative methods, and specifically topic models, imposed as the state-of-the-art. Starting from the assumption that observable data can be randomly generated following an *a priori* determined set of rules, topic models seek to detect the abstract "topics" that occur in a collection of documents.

Latent Dirichlet Allocation (LDA) [Blei *et al.* 2003] is probably the most well-known probabilistic generative model designed to extract topics from text corpora. It considers documents as "bags-of-words" and models each word in a document as a sample from a mixture model. Each component of the mixture can be seen as a "topic". Each word is generated from a single topic, but different words in a document are generally generated from different topics. Each document is represented as a list of mixing proportions of these mixture components and thereby reduced to a probability distribution on a fixed set of topics.

LDA is highly related to **probabilistic Latent Semantic Analysis** [Hofmann 1999] (pLSA), except that in LDA the topic distribution is assumed to have a *Dirichlet* prior. This point is highly important because it permits to go beyond the often-criticized shortcoming of PLSA, namely that it is not a proper generative model for new documents and overfitting. More precisely, LDA is based on the hierarchical generative process illustrated in Figure 6.3.

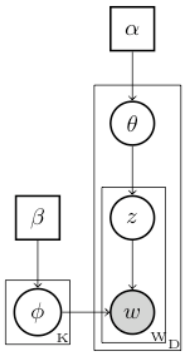


Figure 6.3 – Schema of Latent Dirichlet Allocation.

The hyperparameters α and β are the basis of two Dirichlet distributions. The first Dirichlet distribution deals with the generation of the topic mixture for each of the $|\mathcal{D}|$ documents. The second Dirichlet distribution regards the generation of the word mixture for each of the k topics. Each topic is then a distribution over the $|\mathcal{W}|$ words of the vocabulary. The generative process is the following: for each word $w_{d,i}$ of the corpus, draw a topic μ depending on the mixture θ associated to the document d and then draw a word from the topic μ.

Note that words without special relevance, like articles and prepositions, will have roughly even probability between classes (or they can be placed into a separate category). As each document is a mixture of different topics, in the clustering process, the same document can be placed into more than one group, though resulting in a (kind of) **Overlapping Clustering Process**. Learning the parameters θ and μ, and sometimes the hyper-parameters α and β, is rather difficult because the posterior $p(\theta, \mu/D, \alpha, \beta, k)$ cannot be fully calculated, because of an infinite sum in the denominator. Therefore various approximation algorithms must be used, such as variational EM, Monte-Carlo Markov processes, *etc.*

This probabilistic approach presents advantages and disadvantages:

- The theoretical framework is well-known in bayesian statistics and well-grounded. It has led to many fruitful researches (see below).
- It is designed to make inferences on new documents: what are the associated topics and with what proportions? What part of the document is associated to which topic? Depending on the likelihood $p(d/\Theta)$, does a new document describe an original mixture of topics or a new, never seen before topic?
- LDA is a complex mathematical model, which considers each document as a combination of possibly many topics. While this may be interesting for describing the documents, in the case of clustering, it could lead to a situation where each document belongs, more or less, to many clusters (similar to a fuzzy approach). An issue is therefore to be able to choose a finite (and hopefully short) list of topics to be associated to the document, beyond setting a simple threshold parameter.
- This method does not present a center for each cluster, but a distribution of the topics over the terms. This could make it difficult to associate a readable name to the cluster. Note that recent works relative to LDA are seeking to find useful names using

n-grams [Wang *et al.* 2007].
- As in the other presented methods, this probabilistic approach does not solve the classical problem of finding the global optimum and choosing the number k of topics. For the latter, some methods are proposed inspired by the works in model selection [Rodríguez 2005].

Numerous works have followed the hierarchical generative model of LDA to deal with various related issues: extracting topic trees (hLDA) [Blei *et al.* 2004], inducing a correlation structure between topics [Blei & Lafferty 2006a], modeling topics through time [Blei & Lafferty 2006b, Wang *et al.* 2008], finding n-grams instead of single words to describe topics [Wang *et al.* 2007], social networks [Chang *et al.* 2009b], opinion mining [Mei *et al.* 2007a] *etc.*

6.3.3 Topic Labeling

Distributions of probabilities and multidimensional vectors are hardly comprehensible for a human reader. As shown in Section 6.3, there is no unanimously accepted method for presenting topics. Most authors limit themselves to showing the top scoring words for each topic. While a list of words already gives a main idea of a topic, it is more interesting to present the human reader a phrase to summarize the idea behind the group of documents.

To provide better topic descriptions, systems like Topical N-Grams [Wang *et al.* 2007] embed "spatial" connections between words in the topical learning process. Therefore, topic names are inferred simultaneously with the topics. Topical N-Grams extracts topics as distribution over n-grams (sequences of multiple words which appear often together). Presenting highly scored expressions is more comprehensible than a list of words [Wang *et al.* 2007].

Most topic labeling algorithms assign topic names in a post-processing phase of the topic extraction. This type of approach is a two phase process: (i) the name candidates are extracted from the text corpus and (ii) the meaningfulness of each candidate is calculated and the highest scoring one is chosen. [Mei *et al.* 2007b] address the labeling task as an optimization problem that aims at minimizing the Kullback-Leibler (KL) divergence [Kullback & Leibler 1951] between word distributions. In this case, the two compared distributions are the topic itself and the name candidate's word distribution. The obtained distance is an indicator of the name candidate's meaningfulness to the analyzed topic. In [Osinski 2003], frequent keyphrases are extracted from web search snippets and assigned to topics, by using the cosine distance to calculate their similarity.

Phrase learning All algorithms that label topics in a post-processing phase share the need for relevant and unambiguous phrases, which later serve as name candidates and are used to synthesize the idea of the group of documents associated to the topic. Comprehensible topic names need to be complete phrases: they take into account the property of polysemy and they are humanly readable. As shown in Section 6.3, words can have different meanings, depending on the context. This is one of the reasons why single words rarely make good topic names. Name candidates should be expressions which are precise enough to specify the meaning of words (*e.g.*, "data mining" compared to the single word "mining"). Humanly readable topic names contain words in their original textual form (not lemmatized, nor stemmed) and all the needed prepositions and articles (*e.g.*, "the Ministry of Internal

Affairs"). Name candidates are also called **keyphrases**, sequences of one or multiple words that are considered highly relevant as a whole.

Based on the *learning paradigm*, keyphase learning algorithms can be divided into **constructive algorithms** and **extractive algorithms** [Hammouda *et al.* 2005]. **Constructing keyphrases** is usually a supervised learning task, often regarded as a more intelligent way of summarizing the text. These approaches make use of the external knowledge, under the form of expert validation of the extracted phrases. The obtained results are usually less noisy, but involving human supervisor makes the extraction process slow, expensive and biased towards the specific field (*e.g.*, specialized to microbiology vocabulary). These approaches do not scale to large datasets of general purpose texts. Examples of such systems follow: *ESATEC* [Biskri *et al.* 2004], *EXIT* [Roche 2004], *XTRACT* [Smadja 1991]. **Extracting keyphrases** is an unsupervised learning task, in which candidate names are discovered using predefined patterns. This kind of approaches scale well to large datasets, but they have the drawback of almost exponential quantity of extracted keyphrases and a noisy output. Examples of such systems follow: *CorePhrase* [Hammouda *et al.* 2005], *Armil* [Geraci *et al.* 2006], *SuffixTree Extraction* [Osinski 2003].

According to the employed method, keyphrase learning algorithms can be divided into: **linguistic approaches**, **statistical approaches** and **hybrid approaches** [Roche 2004, Buitelaar *et al.* 2005, Cimiano *et al.* 2006].

Linguistic approaches

Linguistic approaches are issued from the domains of Terminology and Natural Language Processing. They employ linguistic processing, like phrase analysis or dependency structure analysis. A part of speech tagger is usually employed for the morphological characterization of words (*e.g.*, determine (i) if the word is a noun, adjective, verb or adverb, and (ii) its number, person, mode, time *etc.*). Words are further characterized with the semantic information and their lemma (see Section 6.2.1). From the texts enriched with syntactic and morphological information, keyphrases are extracted, most often using predefined patterns. TERMINO [Lauriston 1994] uses patterns like <Head> <Prepositional Group> <Adjectival Group> to discover keyphrases. LEXTER [Bourigault 1992] uses the morphological information to extract from the text nominal groups and then searches for dis-ambiguous maximal nominal groups. INTEX [Silberztein 1994, Ibekwe-SanJuan & SanJuan 2004] allows simple definition of morpho-syntactic rules to infer keyphrases. Such rules are (i) a keyphrase can contain an adverb, one or multiple adjectives, a single preposition and at least a noun, (ii) phrases made of preposition and determinant are excluded *etc.* The keyphrases extracted by INTEX can be further enriched [Ibekwe-SanJuan & SanJuan 2004] by adding variant phrases, using the FASTR system [Jacquemin 1999]. Variant phrases are variations of initial phrases by altering the word order or inserting words (*e.g.*, "online customer support" is a variant phrase of "online support").

Keyphrase extraction methods based on linguistic approaches obtain less noisy output than purely statistical methods, but they are vulnerable to multilingual corpora, neologisms and they have the tendency to adapt to stereotypical texts (texts from a specified narrow field) [Biskri *et al.* 2004]. Furthermore, the morphological and syntactic analyzers they employ, as well as their predefined rules are sensible to the text quality. This makes

linguistic-based approached scale badly to internet-originated text, which usually written in an informal style (this is especially true for text on social networks and micro-blogging). Furthermore, they are costly to develop and maintain, due to the expert supervised task of defining patters, updating them *etc.*

Statistical approaches

Statistical keyphrase extraction algorithms are based on information retrieval techniques for term indexing [Salton & Buckley 1988]. The underlying assumption is that the sense of a word (see the property of polysemy) is given by its context (*i.e.* other words which have a strong relationship with the given word). Statistical measures are employed to detect keyphrases of strongly related words. A widely used measure which quantifies the dependency between two words in the binary collocation (also called bigram) is the *Mutual Information*, given by the formula:

$$mi(w_i, w_j) = p(w_i, w_j) \times log\left(\frac{p(w_i, w_j)}{p(w_i) \times p(w_j)}\right)$$

where $p(w_i)$ and $p(w_j)$ are the probabilities with which the words w_i and, respectively, w_j appear in the text, while $p(w_i, w_j)$ represents the probability of the words w_i and w_j appearing together in a window of a specified length. In [Anaya-Sánchez *et al.* 2008], bigrams are detected using a window of dimension 11 (5 words before + considered word + 5 words after). Other tools (*e.g.*, **EXIT** [Roche 2004], **ESATEC** [Biskri *et al.* 2004]) rely on constructing n-grams, by iteratively combining bigrams or increasing the length of a previously discovered (n-1)gram. Longer collocations are obtained, with higher *Mutual Information* score.

Many other statistical measures have been proposed to calculate the strength of the relationship between two words. In [Anaya-Sánchez *et al.* 2008] a modified entropy function is used to determine frequent bigrams from a set of frequent terms. LocalMaxs [da Silva *et al.* 1999, Dias *et al.* 2000] uses the *Symmetric Conditional Probability* measure to extract continuous multiple word units and the *Mutual Expectation* measure for extracting non-continuous multiple word units. Some works study the impact of some of the most used measures, judging their ability to identify lexically associated bigrams. The compared measures are: *t-score*, *Pearson's χ-square test*, *log-likelihood ratio*, *pointwise mutual information* and *mutual dependency* [Thanopoulos *et al.* 2002].

There are other approaches that do not rely on bigram detection and n-gram construction for keyphrase extraction. In **CorePhrase** [Hammouda *et al.* 2005] keyphrases are considered to naturally lie at the intersection of textual documents in a cluster. The algorithm compares every pair of documents to extract matching phrases. It employs a document phrase indexing graph structure, the Document Index Graph. It constructs a cumulative graph, representing currently processed documents. When introducing a new document, its associated subgraph is matched against the existing cumulative graph to extract the matching phrases between the new document and all previous documents. The graph maintains a complete phrase structure identifying the containing document and phrase location, so cycles can be uniquely identified. Simultaneously, it calculates some predefined phrase features that are used for later ranking. Suffix Array-based techniques [Osinski *et al.* 2004, Osinski

& Weiss 2004] can also used to discover keyphrases. Incomplete phrases (*e.g.*, "President Barack" instead of "President Barack Obama"), which are often meaningless, can be avoided by using the notion of *phrase completeness*. A phrase is complete if and only if all of its components appear together in all occurrences of the phrase. In the previous example, if the phrase "President Barack" is followed in all of its occurrences by the term "Obama", then it is not a complete phrase. Starting from this definition, right and left completeness can be defined (the example above is left complete, but not right complete). Using a Suffix Array data structure [Manber & Myers 1993], the complete phrases can be detected and the ones occurring with a minimum frequency populate the topic label candidate set. More details about a suffix array-based algorithm for keyphrase extraction can be found in Section 6.4.1.

Unlike linguistic approaches, statistical keyphrase extraction systems produce noisy results [Biskri *et al.* 2004]. While extracted name candidates pass the frequency threshold and get good statistical scores, many of them hardly capture any semantic meaning. Such are the phrases which (i) are made out of common words, like articles, prepositions, certain verbs *etc.* (*e.g.*, "he responded that", "the biggest part of the") and (ii) bring no information to the topic. The advantage of statistical methods is that they scale well to big datasets and they are virtually language independent.

Hybrid approaches

The advantages and inconveniences of linguistic and statistical approaches are complementary. Hybrid approaches aim at picking the best of the two worlds: scaling well to big datasets and producing less noisy outputs. Hybrid systems usually add linguistic information to an essentially statistical system or add statistical information to an essentially linguistic system. For example, predefined keyphrase formats (*e.g.*, <Subject> <Verb> or <Verb> <Adverb>) can be used to filter the output of statistical methods.

[Roche 2004] presents a review of hybrid approaches. For example, **XTRACT** [Smadja 1991] uses morphological and syntactic taggers as a final phase to filter out the noise from the name candidate set resulted using statistical extraction. In the first phase, bigrams are extracted from a grammatically tagged corpus, using an 11-word window. The second phase consists in extracting longer frequent phrases, called "rigid noun phrases". The third phase is the linguistic phase. It consists in using statistic filtering based on the bigram's frequency to associate a syntactic etiquette to extracted bigrams (*e.g.*, <Noun>-<Verb>, <Adjective>-<Noun>). Longer phrases can be constructed based on the bigrams and filtered using predefined syntactic rules.

6.3.4 Topic Evaluation and Improvement

Topic extraction is an unsupervised machine learning task, meaning that no ground truth exists to evaluate the results. This is a classical problem in clustering. Traditional unsupervised tasks are usually evaluated using adapted statistical measures, which quantify the fitness of the obtained results over a particular criterion. Evaluating topics adds another level of difficulty, since constructed topic must not only regroup similar documents, but a semantically coherent idea must also emerge from the topic. For example, topic extraction algorithms based on text clustering can be evaluated using any traditional internal clustering

measure [Halkidi *et al.* 2001] (*e.g.*, cluster variance). But applying such measures does not give any semantic information about the fitness of topics. To take into account the semantic dimension, some authors [Cleuziou 2008, Rizoiu *et al.* 2010] use semantically annotated datasets for evaluating topics. Each text in the dataset is labeled using one or multiple given tags (*e.g.*, economics, oil). Techniques issued from the supervised learning literature are used: topics are trained on a learning set and measures like precision and recall are calculated on a previously unseen test set. We present such an evaluation technique in Section 6.4.1.

Generative topic modeling is today's *de facto* state of the art in topic extraction. Traditionally, these approaches have been evaluated qualitatively and quantitatively. From the qualitative point of view, a sample of topics is usually exhibited in order to convince the reader. Each exhibited topic is a short list of the first terms, ordered decreasingly from a probability perspective. Most of the times, a topic label is manually assigned by the authors. A topic is considered to be good if most of its most probable words are semantically similar. Quantitatively, statistical measures, such as the *perplexity* [Wallach *et al.* 2009], are employed to assess the ability of extracted topics to generalize on unseen data (either through a train set/test set approach or through cross-validation). Even though topic models show good predictive power on new text, their weak point is their underlying assumption is that the latent space is semantically meaningful. In [Chang *et al.* 2009a], an extensive quantitative and qualitative evaluation of the interpretability of the latent space is performed. It is shown that the human judgment does not always coincide with the statistical evaluation measures. Highly scoring topics are sometimes not humanly comprehensible, showing that statistic measures do not always achieve to capture the semantics of the dataset. [Newman *et al.* 2010] addresses this lack of semantic comprehension by using external semantic resources (*e.g.*, Wikipedia, Google) for the evaluation task and uses the *topic coherence* as an evaluation metric. Similarly, concept hierarchies (such as WordNet [Miller 1995]) are used for topic evaluation purposes. In [Musat *et al.* 2011b, Musat 2011], we propose a topic-concept mapping, in which each topic is assigned a topical subtree in the concept hierarchy. The measures of *specificity* and *coverage* are defined and used to evaluate topics. This method will be presented in detail in Section 6.4.2.

Many approaches have been proposed in recent years for improving topic extraction or for adapting it to specific applications. Most often, the improvement techniques deal with embedding or leveraging external semantic information. In [Musat *et al.* 2011a, Musat 2011], we propose a technique for removing topic outliers (*i.e.*, words unrelated to a certain topic). It uses the same topic-concept mapping presented earlier and it will be further detailed in Section 6.5.1. **sLDA** [Blei & McAuliffe 2008] introduces the semantic information under the form of supervision of the topic modeling process. Other approaches adapt the topical model to specific applications. **Latent Dirichlet Allocation with WordNet** [Boyd-Graber *et al.* 2007] is a topic model that uses semantic information (under the form of WordNet) for word sense disambiguation. It is a version of LDA that considers the word senses as hidden variables and attempts to select the right sense when constructing topics.

6.4 Extract, Evaluate and Improve topics

In this section, we present our solutions to the tasks of topic extraction and topic evaluation, discussed in Section 6.1. The focus of our work is, as for most of this thesis, leveraging semantic information into the learning algorithms. The topic extraction system presented in Section 6.4.1 addresses the problem of inferring topics through means of overlapping text clustering. Such an approach has the advantage of authorizing textual documents to be part of multiple topics, depending on the approached subjects. Unlike previous work which uses overlapping textual clustering [Cleuziou 2008], our approach also addresses the problem of human comprehension of topics by assigning each cluster a "humanly-comprehensible" name, chose from a list of frequent complete expressions. Therefore, the user is presented with a readable label instead of a distribution of frequencies. The topic extraction system we present in Section 6.4.1 was implemented in the open-source CKP topic extraction software, one of the algorithms used by the platform `CommentWatcher`.

For the task topic evaluation, the underlying assumption is that statistical measures do not completely succeed in emulating the human judgment of topics. Therefore, we propose, in Section 6.4.2, an approach which uses an existing semantic resource, such as a concept hierarchy (*e.g.* WordNet [Miller 1995]). Using a topic's highly ranked terms, we map it to a subtree in the concept hierarchy, therefore linking a statistically extracted distribution of frequencies to a semantic-aware structure. We redefine the specificity and coverage of the subtree, based on its height and depth in the concept hierarchy, and we evaluate the semantic cohesion of topics. We have undergoing work to integrate the proposed semantic-aware evaluation of topics into `CommentWatcher`.

Once the topic evaluation is implemented into the `CommentWatcher` framework, our developped software will become a veritable topic extraction and evaluation integrated system. Given its modular nature, multiple topic extraction systems could be compared and semantically evaluated. In the remainder of this section, we present in detail the topic extraction component (in Section 6.4.1) and the topic evaluation component (in Section 6.4.2). The experimental validation for each of the components is succinctly presented. For more details about the validation, the reader is invited to refer to the concerning publications.

6.4.1 Topic Extraction using Overlapping Clustering

We present a topic extraction system, which relies on overlapping text clustering and complete keyphrase extraction. Starting from a collection of textual documents (*e.g.*, online discussions, forums, chats, newspaper articles *etc.*), the algorithm extract the discussion topics and presents (i) the topic labels under the form of humanly readable keyphrases and (ii) the partition of texts around the topics.

Figure 6.4 presents the schema of the proposed topic extraction system. In *phase 1*, each of the documents in the dataset is preprocessed, as discussed in Section 6.2.1: stopwords are removed and the inflected words are stemmed. After the preprocessing, the documents are translated into the Vector Space Model representation (see Section 6.2.2) using one of the term weighting schemes. In *phase 2*, the documents are clustered using the OKM algorithm and an overlapping partition is obtained. Each document can be part of one or multiple groups. From the original text of the documents, complete frequent keyphrases

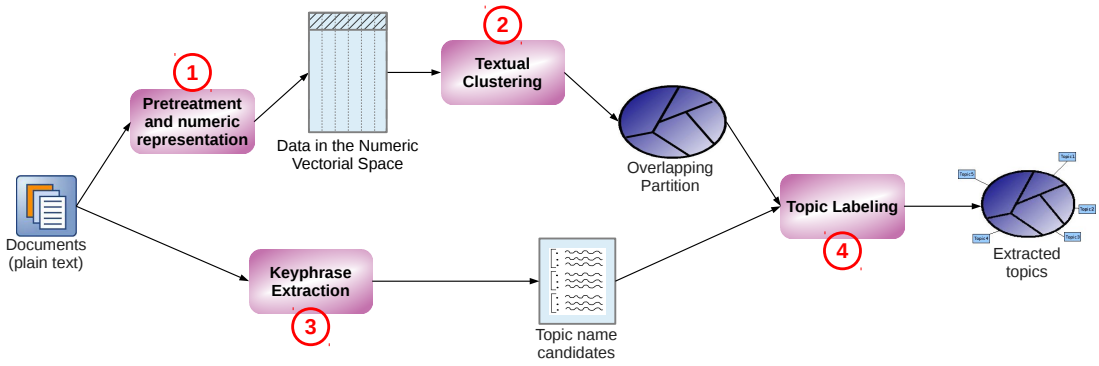


Figure 6.4 – Schema of a topic extraction algorithm using overlapping text clustering.

are extracted, in *phase 3*, using a Suffix Array based algorithm. The extracted keyphrases serve as topic label candidates. In *phase 4*, the topic label candidates are are reintroduced as pseudo-documents into the multidimensional space defined by Vector Space Model and the cosine distance is used to chose the best name for each topic.

6.4.1.1 *Phase 2*: Clustering

The text clustering is performed using OKM [Cleuziou 2008], which is a K-Means variant which authorizes documents to belong to more that one cluster. It inherits from K-Means most of its drawbacks (*i.e.*, its powerful dependence on the initialization and the number of clusters which must be set arbitrarily by the user) and its advantages (*i.e.*, linear execution time, exposing a centroid). Nonetheless, OKM is chosen for the text clustering task for its capacity to create an overlapping partition of the dataset. This is especially important for two reasons: the property of polysemy of words and multi-topic documents. Words should be allowed to be associated with multiple topics, given their corresponding senses. Similarly, documents that approach multiple thematics should authorized to be part of their multiple corresponding topics.

OKM follows the general K-Means schema, of iteratively optimizing an objective function by relocating the cluster centroids and re-assigning documents to clusters. In K-Means, each document is associated to only one centroid, the one closest in terms of the employed distance. OKM assigns a document to multiple clusters by constructing a document image as the gravity center of all the associated centroids. Lets take the example of a document d and centroids c_1, c_2 and c_3. Without losing generality, we consider that

$$||d - c_1||_{cos} < ||d - c_2||_{cos} < ||d - c_3||_{cos}$$

where $||\bullet||_{cos}$ is the cosine distance defined in Equation 6.1. Let $\bar{d}$ be the image of the document d and A be the set of centroids to which the document d is assigned. Initially, the document d is assigned to the closest centroid ($A \leftarrow c_1$) and, therefore, $\bar{d} = c_1$. The second closest centroid (c_2) is now considered and the new document image is computed as the gravity center of c_1 and c_2 ($\bar{d}_{new} = gravity(c_1, c_2)$). If the new document image is closer to the document than the old image ($||d - \bar{d}_{new}||_{cos} < ||d - \bar{d}_{old}||_{cos}$) then the document d is also assigned to the cluster with the centroid c_2 ($A \leftarrow A \cup c_2 = \{c_1, c_2\}$). The process

is continued with c_3: $\bar{d}_{new} = gravity(c_1, c_2, c_3)$. If the new document image is not closer to d than the old image ($||d - \bar{d}_{new}||_{cos} \geq ||d - \bar{d}_{old}||_{cos}$), then the process is finished and the document d is assigned to clusters with the centers c_1 and c_2.

In the centroid update step, each document d_i contributes to each associated centroid (each $c_j \in A_i$). The contribution is inversely proportional with the number of clusters to which the document belongs to: the more clusters the document is part of, the less influence it has in the update of their respective centroids. The update formula for the centroids:

$$c_j = \frac{1}{\sum_{d_i \in \mathcal{C}_j} \frac{1}{\delta_i^2}} \times \sum_{d_i \in \mathcal{C}_j} \frac{\hat{d}_i^j}{\delta_i^2} \tag{6.3}$$

where $\hat{d}_i^j = \delta_i d_i - (\delta_i - 1)\bar{d}_i^{A_i \setminus \{c_j\}}$. Notations:

- A_i is the set of centroids to which the document d_i is assigned;
- $\delta_i = |A_i|$ is the number of centroids associated with document d_i;
- $\mathcal{C}_j$ is the collection of documents associated to the centroid c_j;
- $\bar{d}_i^{A_i \setminus \{c_j\}}$ is the image of document d, excluding centroid c_j (gravity center of the centroids to which d_i is associated, except centroid c_j);
- c_j is the centroid to be updated.

Unlike K-Means, in OKM the update of a centroid (Equation 6.3) is dependent not only on the documents under its cluster, but also on the other centroids (through the document image). The result is that centroids can continue to evolve in the multidimensional space, if the cluster composition stopped changing. In the process of topic labeling, topic names are associated to each resulted cluster based on the similarity between the name candidate and the cluster's centroid. Therefore, we modify the stopping criterion of the original OKM algorithm proposed in [Cleuziou 2008] in order to allow the centroids to evolve until they are fully adapted to the documents in the cluster. We define a threshold ε and we stop the clustering process when the variation of the objective function (see Equation 6.2, p. 133) between two iterations descends under the threshold. In practice, this means that the last couple of iterations of the clustering algorithm are performed only to refine the centroids and adapt them to the documents in their respective clusters.

The clustering phase creates a data partition that regroups documents relative to their topic similarity. Simultaneously, this phase outputs the centroids of each class, which can be regarded as abstract representations of the topics. These centroids are multimensional vectors in the Vector Space Model, having a high scores for the words that are specific to the cluster (*i.e.*, the words that are characteristic for the specific topic).

6.4.1.2 *Phase 3*: Keyphrase Extraction

In order to populate the topic label candidate set, we extract frequent keyphrases from the original text of the document. Common words (*e.g.*, prepositions, articles), that we removed in the preprocessing phase of the text clustering are necessary for the human comprehension of topic labels. The topic labels need to be correctly formed expressions. Initial documents is usually correctly formed, therefore, it suffices to extract sequences of words from the initial text, which fulfill several conditions [Osinski 2003]:

Table 6.1 – The suffix array for the phrase "to be or not to be"

No	Suffix	Start Pos
1	be	6
2	be or not to be	2
3	not to be	4
4	or not to be	3
5	to be	5
6	to be or not to be	1

- they appear in the text with a minimum specified frequency. The underlying assumption is that keyphrases which occur often in the text are related to the discussed topic in a higher degree than the rare ones.
- they do not cross the sentence boundary. Usually, meaningful keyphrases are contained into a sentence, since sentences represent markers of topical shift.
- they are a complete phrase, as defined in Section 6.3.3 (*e.g.* "President Barack" vs. "President Barack Obama").
- they do not begin or end with a stopword. For increased readability, cluster name candidates are stripped of leading and trailing stopwords, though stopwords inside the phrase will be preserved.

The keyphrases are extracted using a **Suffix Array**-based [Manber & Myers 1993] algorithm, motivated by its capability to process raw untreated text, its language independence, efficient execution time and the power to extract humanly readable phrases. This approach has been proved very efficient. [Yamamoto & Church 2001] used a Suffix Array-based algorithm to compute term frequency and document frequency for all n-grams in large corpora. Further temporal execution optimizations are proposed in [Abouelhoda *et al.* 2002], who apply it to the problem of optimum exact string matching. [Kim *et al.* 2003] propose an algorithm for linear time construction of the suffix array data structure.

The keyphrase extraction algorithm uses the property of phrase completeness and it has two phases: in the first phase, the left and right complete expressions are found. In the second phrase, the two sets are intersected to obtain the set of complete expressions.

Suffix Array Construction A Suffix Array is an alphabetically ordered array of all suffixes of a string. We note that in the case of keyphrase extraction, the fundamental unit is not the letter (as in the case of classical strings), but the word. For example, the suffix array for the phrase "to be or not to be" is shown in Table 6.1.

The bottleneck of the construction of the suffix array data structure is the sorting of the suffixes. Two approaches are compared in [Larsson 1998] from the theoretical and practical performance point of view: **Manber and Myers** and **Sadakane's algorithm**. Our keyphrase extraction algorithm implements the latter, as shown to obtain better results in terms of time execution efficiency. Although in Table 6.1, for the sake of clarity, we have ordered the suffixes alphabetically, the sorting algorithm only requires that the terms have a lexicographic order. The arrival order of words into the collection can also be used, which further speeds up the sorting. The **Sadakane's sorting algorithm** is a modified bucket

sorting, which takes into consideration the unequal dimensions of the suffixes. Considering that a keyphrase can not pass the boundary of a sentence, we modify the construction of the suffix array so that it is sentence-based. In practice, we build a suffix array for each sentence and then we merge everything into a single structure. Therefore, the identification of a suffix is given by the beginning of the suffix and the index of the its containing sentence.

Complete Phrase Discovery The right complete keyphrases are discovered by linearly scanning the constructed suffix array in search for frequent prefixes and counting their number of occurrences. Information about position and frequency is stored alongside the identified prefixes. Discovering the left complete phrases is achieved by applying the same algorithm to the inverse of the document. A version of the document which has the words in reverse order is created, right complete phrases are detected and the correct complete left phrases are recovered by inverting the order of the extracted prefixes. Both left complete and right complete phrase sets are in lexicographic order, therefore they can be intersected in linear time. Name candidates are returned along with their frequency.

The last phase is filtering the obtained phrase set using the minimum frequency condition and the stripping of leading and trailing stopwords. Only phrases that appear in the text with minimum frequency are kept, the rest are eliminated. [Osinski 2003] set the value of this threshold between 2 and 5: the most frequent expressions are not necessarily the most expressive. They are usually frequent expressions made out of common words (*e.g.* "in order to"). In the end, leading and trailing stopwords are recursively eliminated from the phrases. This further filters the candidate set. Some of the most frequent candidates disappear completely (they are composed solely from stopwords). Others become duplicates of existing phrases (*e.g.*, "the president" and "president of", both duplicate "president" when the leading "the" and the "trailing" of are stripped).

6.4.1.3 *Phase 4*: Topic Labeling

The result of the text clustering space is a multidimensional space in which documents are translated, the clusters that thematically regroup the documents and the centroids, which summarize each cluster. The keyphrase extraction phase generates a list of topic label candidates. In the last phase of the topic extraction, a suitable name is chosen among the name candidates to label each topics. This is done by introducing the name candidates as "pseudo-documents" in the same vectorial space defined for the document collection. The keyphrases are extracted from natural language texts, so they may contain inflected words and stopwords. The same preprocessing (*i.e.*, stopwords removal, stemming) and the same term weighting scheme are used with the name candidates as with the original documents. After translating into the Vector Space Model, the last step is to calculate the similarity between the "pseudo-documents" and the centroids of each class. The highest scoring candidate is chosen to serve as topic label.

Other that labeling topics, this approach can be used to filter semantically irrelevant expressions from a phrase set. As centroids are abstractions of the documents in their respective classes, choosing phrases that are close in term of the considered distance, naturally eliminates phrases that are too general or semantically irrelevant. This has a similar effect as adding linguistic filters to statistic methods (presented in Section 6.3.3), but without

their language and field dependency. For example, in a document group that talks mainly about politics, the highest scoring would naturally be "parliament", "govern", "president", "party", "politics" *etc.* When calculating the similarity between the group's centroid and the phrase candidates, phrases like "presidential elections" would have a higher similarity than semantically irrelevant phrases like "as a matter of fact".

6.4.1.4 Experimental Validation

In this subsection, we briefly present some experiments and results that can be obtained with this topic extraction system. The performed experiments are still preliminary, being performed on small datasets. We have planned to perform a more complete batch of experiments and to compare to other topic extraction algorithm once we complete the integration of the semantic-aware topic evaluation into `CommentWatcher`. We choose to present these preliminary experiments mostly from a qualitative point of view, to show the capabilities of the proposed system, and most notably the topic labeling part.

The experiments are performed on an English and a French dataset. The English dataset is a subset of the `Reuters` [6] corpus, composed of 262 documents. The writing style is journal article, each text contains between 21 and 1000 words. The French dataset [7] is issued out of a reader discussion forum attached to a news article, entitled `Commemoration` [8]. It has an informal writing style, it is composed of 272 documents, each one containing between 1 and 713 words.

Qualitative evaluation Table 6.2 presents an example of three extracted topics from the `Reuters` subset. The 10 highest rated words, 3 highest scoring documents and the chosen topic label are presented for each topic. Note that the highest rated words are in their stemmed version, and, therefore, they are not always existing words. The first column shows the extracted topic labels: "oil and gas company", "tonnes of copper" and "united food and commercial workers". The second column presents the ten top scoring words associated with each topic. The words are presented in their stemmed version. The next two columns indicate the number of documents covered by each topic and three examples of documents from each clusters. Let's consider the example of the topic which covers the maximum number of texts: "oil and gas company". The first two texts talk explicitly about the economical activities of companies that operate in the business of oil and natural gas (buying oil and natural gas proprieties in the first case and estimating reserves in the second case). On the other hand, the third document talks about the "food for oil" program between Brazil and Iraq. Unlike the first two documents, the text does not refer to an oil company. Whatsoever, the document is placed under the same topic, because it approaches the thematic of "oil and gas". It also approaches with the thematic of food and this is why the document is also found under the topic "united food and commercial workers". This emphasis the importance of the overlapping property of the clustering algorithm.

6. http://mlr.cs.umass.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection
7. http://eric.univ-lyon2.fr/~arizoiu/files/commemoration.tar.bz2
8. http://www.liberation.fr/societe/0101220668-y-a-t-il-trop-de-commemorations-en-france

Table 6.2 – Example of topics extracted from the `Reuters` dataset.

Topic Label	Highest Rated Words	Number of docs	Text Excerpt
oil and gas company	oil, mln, ga, year, barrel, billion, lt, compani, reserv, natur	169	Kelley Oil and Gas Partners Ltd said it has agreed to purchase all of CF Industries Inc's oil and natural gas properties for about 5,500,000 dlrs, effective July 1. It said the Louisiana properties had proven reserves at year-end of 11 billion cubic feet of natural gas and 85,000 barrels of oil, condensate and natural gas liquids. Kelley said it currently owns working interests in some of the properties.
			Hamilton Oil Corp said reserves at the end of 1986 were 59.8 mln barrels of oil and 905.5 billion cubic feet of natural gas, or 211 mln barrels equivalent, up 10 mln equivalent barrels from a year before.
			Brazil will export 6,000 tonnes of poultry and 10,000 tonnes of frozen meat to Iraq in exchange for oil, Petrobras Commercial Director Carlos Sant'Anna said. Brazil has a barter deal with Iraq and currently imports 215,000 barrels per day of oil, of which 170,000 bpd are paid for with exports of Brazilian goods to that country.
tonnes of copper	tonn, copper, cent, price, mine, effect, beef, lb, meat, export	100	Mountain States Resources Corp said it acquired two properties to add to its strategic minerals holdings. The acquisitions include a total of 5,100 acres of titanium, zirconium and rare earth resources, the company said. (...) The company also announced the formation of Rare Tech Minerals Inc, a wholly-owned subsidiary.
			Magma Copper Co, a subsidiary of Newmont Mining Corp, said it is cutting its copper cathode price by 0.75 cent to 66 cents a lb, effective immediately.
			Newmont Mining Corp said Magma Copper Co anticipates being able to produce copper at a profit by 1991, assuming copper prices remain at their current levels. In an information statement distributed to Newmont shareholders explaining the dividend of Magma shares declared Tuesday
united food and commercial workers	unit, compani, plant, union, beef, lt, offer, contract, iowa, term	93	The United Food and Commercial Workers union, Local 222 said its members voted Sunday to strike the Iowa Beef Processors Inc Dakota City, Neb., plant, effective Tuesday. The company said it submitted its latest offer to the union at the same time announcing that on Tuesday it would end a lockout that started December 14. (...)
			Brazil will export 6,000 tonnes of poultry and 10,000 tonnes of frozen meat to Iraq in exchange for oil, Petrobras Commercial Director Carlos Sant'Anna said. Brazil has a barter deal with Iraq and currently imports 215,000 barrels per day of oil, of which 170,000 bpd are paid for with exports of Brazilian goods to that country.
			European Community agriculture ministers agreed to extend the 1986/87 milk and beef marketing years to the end of May, Belgian minister Paul de Keersmaeker told a news conference. He said the reason for the two-month extension of the only EC farm product marketing years which end during the spring months was that it would be impossible for ministers formally to agree 1987/88 farm price arrangements before May 12. (...)

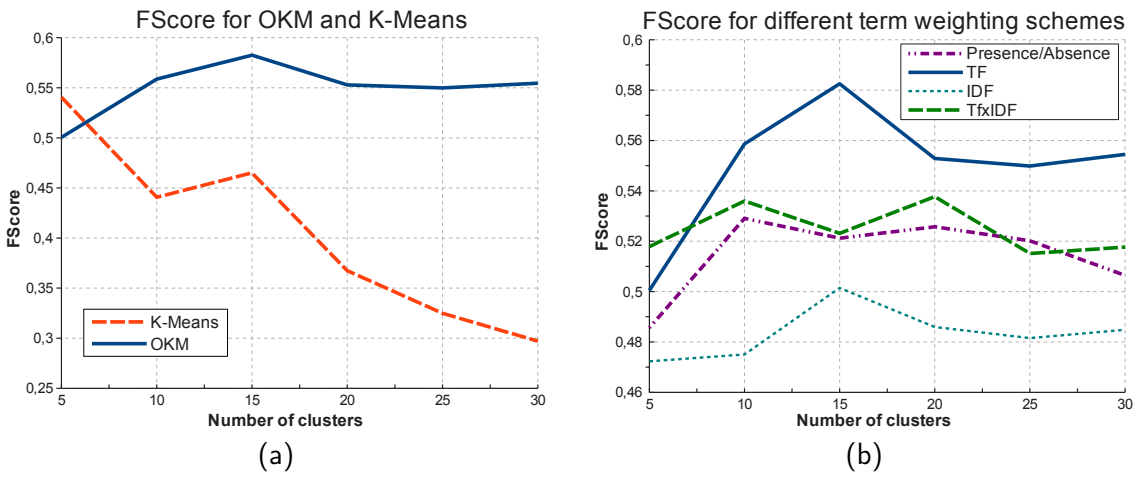


Figure 6.5 – F_{score} as a function of the number of clusters for (a) OKM and K-Means with *Term Frequency* and (b) OKM with different term weighting schemes.

Quantitative evaluation The text clustering phase and topic labeling phase are evaluated individually. We study (a) the behavior of OKM in text clustering compared to the baseline K-Means and (b) the influence of the term weighting scheme on text clustering results. The number of clusters is varied between 5 and 30 and the clustering is re-initialized 10 times and averages are reported. The main approach towards evaluating the quality of the resulted partition is to use the classical precision, recall and F_{score} indicators on a corpus that has been tagged *a priori* by human experts. A sub-collection of the `Reuters` corpus is used, more precisely documents that have associated at least one tag. Two documents are considered to be "correctly" clustered if they are partitioned into the same cluster and they have at least a tag in common. The results in Figure 6.5a show that OKM out-performs the classical crisp algorithms, when being used for text clustering. In Figure 6.5b, we show the evolution of the F_{score} for multiple term weighting schemes. Our experiments show that the *Term Frequency* weighting scheme outperforms the other classical schemes presented in Section 6.2.2.

Human judges are often used in the literature (*e.g.*, [Osinski 2003, Chang *et al.* 2009a]) to assess the interpretability of the extracted topic. The argued reason is that the literature does not provide a widely accepted measure for quantifying topic label quality. Moreover, topic names need to be humanly-readable and they need to synthesize the idea behind a group of texts. Therefore, evaluating them is trying to evaluate "human tastes". We choose a similar approach in order to evaluate the topic labels associated to each cluster. A number of 5 experts were given the labeled topics and were demanded to assess if the label given no information about the topic (grade 0), has an average quality (grade 1) or it expresses a comprehensible idea (grade 2). The results are presented in Figure 6.6a for `Reuters` and in Figure 6.6b for `Commemoration`, under the form of stacked bars. The results show a rather good acceptance by the users of the constructed labels. The *Term Frequency* and the *Presence/Absence* term weighting schemes obtain around 90% of good and average scores.

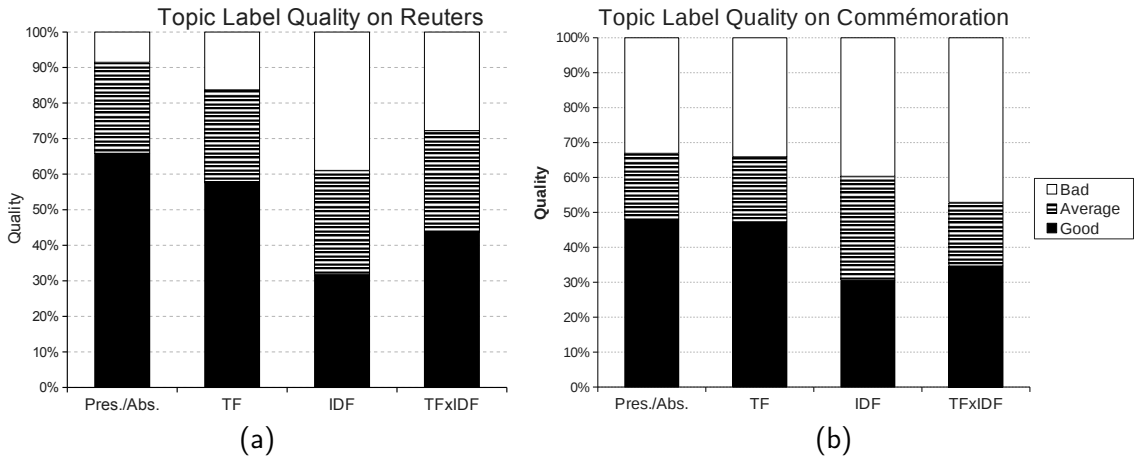


Figure 6.6 – Topic labeling quality function of the term weighting schemes for (a) `Reuters` dataset and the (b) `Commemoration` dataset.

6.4.2 Topic Evaluation using a Concept Hierarchy

As shown in Section 6.3, most topic extraction algorithms are statistical-only approaches. With the exception of a minority of algorithms, some of which concerning topic labeling were already presented in Section 6.3.3, text clustering-based algorithms and generative model-based algorithms use numeric-only methods to capture the semantics of Natural Language. Furthermore, even the evaluation is often performed using statistical measures only, as seen in Section 6.3.4. Therefore, the entire topic extraction and evaluation process short-circuits language semantics, assuming that the statistical process succeeds in capturing the meaning. Recent works like [Chang *et al.* 2009a] have proved this assumptions to be too optimistic, by showing that human judgment does not always coincide with the statistical results.

The topic extraction community has addressed this problem and the literature shows examples of algorithms that leverage external semantic resources (see Section 6.3.2). One of the most popular and most employed lexical resources in Natural Language Processing is **WordNet** [Miller 1995]. It is also an important resource used in computational linguistics, text analysis and other areas. WordNet is a lexical database for the English language and is one of the largest existing semantic networks. The relations of hypernymy and hyponymy exist between certain WordNet building blocks. These relations can be interpreted as specialization relations between conceptual categories, thus WordNet can be interpreted and used as a lexical ontology [Gangemi *et al.* 2003]. Words are grouped in WordNet into *synsets*, sets of synonyms. These are paired with *glosses*, which are short, general definitions of the synset. Then the resulting tuples are linked with various types of relations, including hypernymy, hyponymy, meronymy, holonymy or antonymy. Given the property of polysemy, each word may have several senses and for each sense it has a set of synonyms.

As for topic evaluation, automatic semantic-based evaluation systems are scarce (see Section 6.3.4). We propose an original system, that uses a semantic resource under the form of a concept hierarchy (in our case applied to WordNet) to automatically evaluate topics. The underlying idea is, when evaluating topics, to emulate the human judgment

Table 6.3 – Notations used in the next sections.

Notation	Equation definition	Meaning
$\mathcal{C}$	6.4 (p. 151)	The employed concept hierarchy.
$c \in \mathcal{C}$	6.4 (p. 151)	A concept in the concept hierarchy $\mathcal{C}$.
$c_0(\mathcal{C})$	6.6 (p. 155)	The root concept of the concept hierarchy $\mathcal{C}$.
$\mathcal{C}_c$	6.9 (p. 156)	The subtree having the concept c as a root.
μ	p. 154	A topic to be evaluated.
$P(\mu)$	6.9 (p. 156)	The set of relevant words of the topic μ
w	p. 152	A word that appears in a given topic.
$\mathcal{C}_\mu$	6.11 (p. 157)	The topical subtree of topic μ in the entire concept hierarchy.
$c_{opt}(\mu)$	6.11 (p. 157)	The root of the "optimum" subtree of the topic μ.
δ	6.4 (p. 151)	An operator that returns the set of nodes in a given structure.
$\delta(\mathcal{C})$	6.4 (p. 151)	The set of nodes in the concept hierarchy $\mathcal{C}$.
$\delta(c_i, c_j)$	6.4 (p. 151)	Set of nodes within the branch connecting the concepts c_i and c_j.
$d(c_i, c_j)$	6.4 (p. 151)	Edge-based distance between concepts c_i and c_j.
$\delta(w)$	p. 152	Set of concepts (senses of the word w) covered by the hierarchy $\mathcal{C}$.
$\delta(\mu)$	6.7 (p. 156)	Set of concepts related to the topic μ.
$\delta(\mu, \mathcal{C}_c)$	6.9 (p. 156)	Set of concepts in the topical subtree of the topic μ and concept c.
$d(w, c)$	6.5 (p. 152)	The distance between a word w and a concept c.
$depth(c)$	6.6 (p. 155)	The depth of the concept c in the the hierarchy $\mathcal{C}$.
$iheight(\mu, c)$	6.7 (p. 156)	Inverse height of the topical subtree of the topic μ and concept c.
$spec(\mu, c)$	6.8 (p. 156)	Specificity of the topical subtree of the topic μ and concept c.
$cov(\mu, c)$	6.9 (p. 156)	Coverage of the topical subtree of the topic μ and concept c.
$\phi(\mu, c)$	6.10 (p. 156)	The fitness of the topical subtree of the topic μ and concept c.
$\phi(\mu, c_{opt}(\mu))$	p. 157	The topic fitness score of the topic μ.
ω_d	6.8 (p. 156)	The weight of $depth(c)$ in $spec(\mu, c)$.
ω_h	6.8 (p. 156)	The weight of $iheight(\mu, c)$ in $spec(\mu, c)$.
ω_{spec}	6.10 (p. 156)	The weight of $spec(\mu, c)$ in $\phi(\mu, c)$.
ω_{cov}	6.10 (p. 156)	The weight of $cov(\mu, c)$ in $\phi(\mu, c)$.

which served in the creation of the semantic resource. The key point of the system is a topic-concept mapping, that passes thought words. This is equivalent to associating a statistical-only result to a semantic-aware structure. The mapping is very similar to the task of Word Sense Disambiguation (WSD, also known as term disambiguation) needed in automatic ontology learning.

The idea is to search for a concept or a set of concepts which are semantically related to at least one of each of the senses of highly ranking words of a topic. For each topic, a topical subtree is associated in the concept hierarchy. This allows to associate each topic with a "most related" concept and evaluate the topics based on their topic-concept relation strength. The relation's strength is calculated by redefining the topical subtree's coverage of the topic and the specificity of its root concept.

In the next subsections, we present in detail the topic-concept mapping, and the employed notions and measures. In order to help the reader, we create a reference table (presented in Table 6.3), in which we resemble all the notations that gradually appear in the

next subsections. For each notation, we give its meaning, the equation and/or the page where it appears first.

6.4.2.1 Topic - Concept Mapping

The main difficulty when embedding semantic knowledge into an essentially statistic process is the mapping between the two. Given a semantic resource under the form of concept hierarchy, the key point is to map topics to concepts (or a structure of concepts). We consider that each of the concepts in the hierarchy represents senses of words. Knowing that most often a topic is defined as a list of words, ordered decreasingly by their score, we propose a mapping function between topics and concepts that passes by words. Figure 6.7 presents, in a nutshell, the proposed mapping. The most relevant words are selected for each topic, by using, for example, a threshold on their topic score. Each word can be mapped to a set of concepts, which represent the different senses of the word. Therefore, each topic is mapped to a topical subtree in the concept hierarchy (denoted with a dashed line).

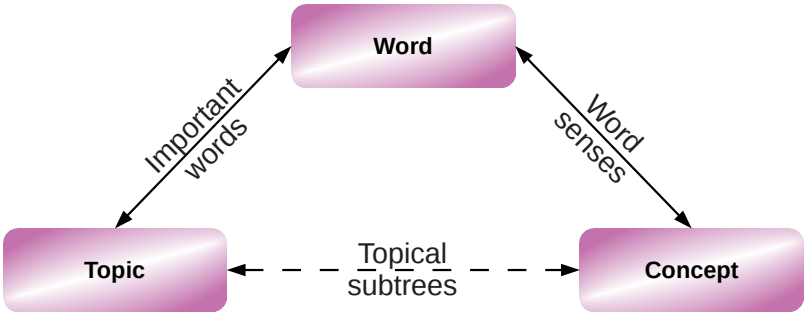


Figure 6.7 – Mapping topics to concepts through words.

Topic - word mapping Each topic is associated with its most relevant words. This idea is similar to the one used in [Chang *et al.* 2009a], where those words alone are sufficient to satisfyingly assess a topic's quality. The most relevant words are usually the highest scoring words for a topic (*e.g.*, in the case of generative topic modeling it is the word's probability associated with a given topic). Any user-defined function can be used, as long as it outputs numerical scores for couples (topic, word) which can be used to rank words and filter a set of relevant words by using a threshold. The topic-concept mapping and the consequent evaluation consider each topic to be defined by its most relevant words. Thus it does not interfere with the creation of the topics and is compatible with most existent topic models.

Concept hierarchy The prior semantic knowledge is available to our evaluation approach under the form of a hierarchy of concepts, linked in a tree structure using a specific relation. Our system uses WordNet as the source of semantic knowledge. We assimilate a *(synset, gloss)* tuple to a concept and use the hypernymy/hyponymy relation as the linking relation of the concepts. A branch in the concept hierarchy is path between two concepts, which are either related directly (in which case the branch contains just the two concepts) or indirectly (in which case the branch contains the two respective concepts and all the concepts in

between). In the case of WordNet's hypernymy/hyponymy relation, a branch between two concepts implies that one concept is the generalization of the other.

Given such a structure, it is possible to determine when concepts are related and in what degree. We consider that the semantic similarity of concepts is inversely correlated with a measure of how apart they are positioned within the tree. If the distance between two concepts is low, than they are considered to be semantically similar. We choose to define an edge-based distance between two concepts: the length of the shortest branch between the two concepts or infinity, if the concepts are unrelated. Of course, other measures exist in the literature [Pedersen *et al.* 2004, Patwardhan & Pedersen 2006], and trying them is a future perspective of our work. Formally, we define the distance between two concepts:

$$d(c_i, c_j) = \begin{cases} |\delta(c_i, c_j)| - 1, & \text{there exists a branch between } c_i \text{ and } c_j \\ \infty, & \text{otherwise} \end{cases} \tag{6.4}$$

where c_i and c_j are two concepts in the concept hierarchy $\mathcal{C}$ and $\delta(c_i, c_j)$ is the set of all the nodes within the branch connecting c_i and c_j. More generally, δ is an operator that returns the set of nodes in a given structure. $\delta(\mathcal{C})$ is, for example, the set of nodes in the concept hierarchy $\mathcal{C}$.

Word - concept mapping Each concept in the concept hierarchy that we employ is associated with a list of words that have the same meaning (the synset). Inversely, given the polysemy of words, each word can have multiple meanings and be, therefore, part of the synsets belonging to multiple concepts. When a word is part of the synset of a concept, we say that the concept is a *sense* of the respective word. In conclusion, each word w is associated with a set of concepts $\delta(w) \subset \delta(\mathcal{C})$, which is the set of senses covered by the concept hierarchy $\mathcal{C}$.

Given the set of senses of a word and a distance which quantifies the strength of the semantic relation between two concepts, we can define the distance between a word w and a concept c as the minimum distance between c and a sense of w. For example, in Figure 6.8 are shown two of the senses of the word "mining": the concepts "metal mining" and "data mining". The distance between the word "mining" and the concept "knowledge management" is 2 (they are semantically related), since there is a branch of length 2 (highlighted in Figure 6.8) between target concept "knowledge management" and the concept "data mining". The concepts of "metal mining" and "knowledge management" are unrelated and have a distance of ∞. Formally, we define the distance between a word w and a concept c as:

$$d(w, c) = \min_{c_w \in \delta(w)} (d(c_w, c)) \tag{6.5}$$

A subtree of a concept c is the subtree that has c as a root. The subtree of concept c is constructed as the reunion of all the branches that connect the concept c to leaf concepts (concepts which are on the last level, the most specific concepts in the hierarchy). Intuitively, the subtree of a concept contains all the possible specializations of the given concept (*e.g.*, the subtree of "mammal" would contain "cat", "dog", "bear" and the even more specific "birman cat", "hunting dog" or "polar bear", but not "lizard").

A word's subtree is the reunion of all the branches that connect the concepts that are senses of the given word ($\delta(w)$) to the root. A word's subtree contains all the possible

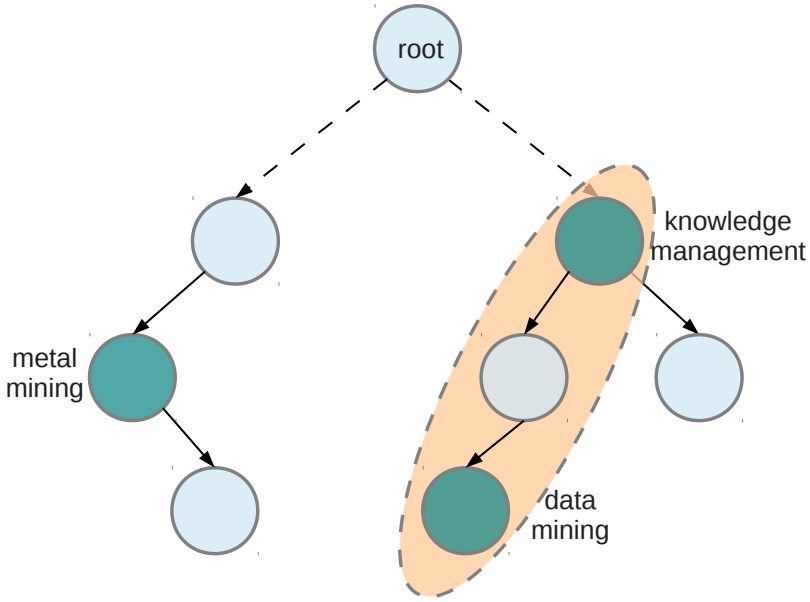


Figure 6.8 – Calculating the distance between the word "mining" and the concept "knowledge management". Here the distance is 2.

generalizations of a word (more precisely the generalizations of the senses of a word). In the example of the word "mining", the word's subtree would contain "resource mining", "natural resource extraction", "activity", but also "data handling", "artificial intelligence", "computer science". Figure 6.8 shows an excerpt of the subtree of the word "mining".

A word's subtree of a concept is the concept hierarchy that contains all the generalizations of the senses of a word up to a certain concept. It is defined as the reunion of all the branches that connect senses of the word w to the concept c. It is the intersection of the word's subtree and the subtree of the concept c. This allows to specify the semantic domain in which to generalize the senses of the word. In the previous example, the word's "mining" subtree of the concept "computer" would contain only "data handling", "artificial intelligence", "computer science" and it would remove the natural resources mining related concepts. The distance between a word and a concept defined earlier can be redefined as the minimum height of the word's subtree of the specified concept. For the example show in Figure 6.8, the distance if 2.

Topic-concept mapping Mapping topic to a concept structure is a similar task to that of mapping words. We start from a topic's relevant words, defined earlier in the topic-word mapping. We define the *topic's related concept set*, which is the union of all senses of the topic's relevant words present in the concept hierarchy. While this set is complete, meaning that is contains all the possible senses associated with the topic, it is too large since many (or most) of the concepts in it are not relevant for the topic. In a given text, words have rarely the tendency to be associated with more than one sense. Let us take the example of a topic extracted from texts talking about data mining. The topic's related concept set will contain "data mining" and "text mining", but also "metal mining", "oil mining" and "data structures". We need to filter the set to retain only concepts that are semantically related

to the topic. This task is similar to the task of term disambiguation needed in automatic ontology learning, which will be presented in Section 6.5.2.

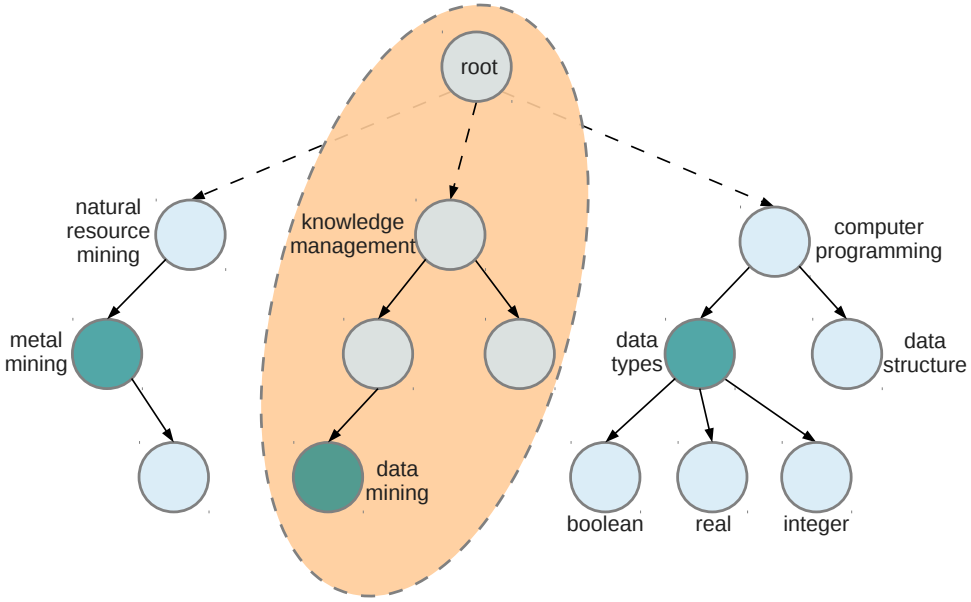


Figure 6.9 – Topical subtree of a topic having the relevant words "data" and "mining", with the common subtrees highlighted

A topical subtree is the union of all the word subtrees of all the topic's relevant words. Similarly to a word's subtree, the topical subtree contains all the generalizations of all the possible senses associated with the topic. Considering that the relevant words of a topic are semantically related, then their senses are semantically related and many of their generalization are identical. This translates into overlapping word subtrees and makes possible the identification of semantically relevant concepts in the topic's related concept set. Going back to the earlier example of topic extract from data mining related texts. The relevant words for this topic are "data" and "mining". Figure 6.9 shows a part from the topical subtree. All the possible senses of the relevant words are present, such as "metal mining", "data mining" and "data types". When looking at the individual words subtrees, intuitivelly the subtree of the concept "natural resource mining" is related to the word "mining", just like the subtree of the concept "knowledge management". The left and the central subtrees in Figure 6.9 correspond to two of the senses of the word "mining". The central and the right subtrees (those of the concepts "knowledge manangement" and "computer programming") are contained in the subtree of the word "data". The concepts that are relevant to the given topic emerge from the overlapping of the two word subtrees. The topical subtree is generally a very wide structure that covers many semantically irrelevant synsets. A filtering mechanism is, therefore, needed to select only the overlapping region of the subtree, which has the highest chances of mapping to the topics semantics.

A topical subtree of a concept c is the intersection between the topical subtree and the subtree of the concept *c*. It is made out of all the generalizations of all the concepts associated with a topic which that are specializations of the topic *c*. In the example given in Figure 6.9, the topical subtree of the concept "computer programming" contains only

"data types", "data structure", "boolean", "real" and "integer". Using the topical subtree of a concept c it is possible to filter regions that are not semantically relevant and focus on smaller, denser structures. Starting from a given concept it is possible to follow the hyperonymy/hyponymy relations to specialize or generalize the topical subtree.

6.4.2.2 Metrics and Topic Evaluation

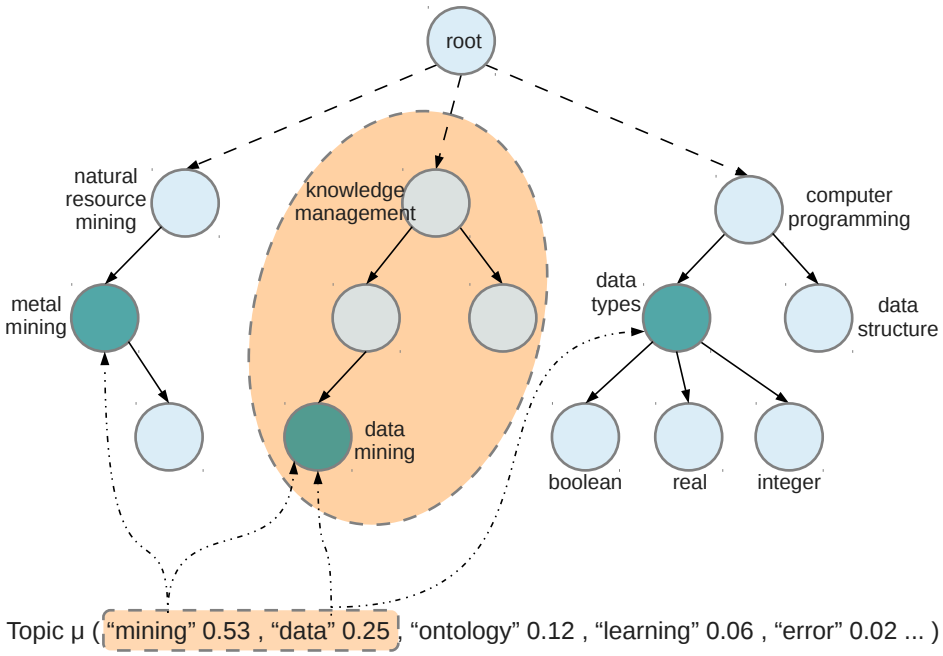


Figure 6.10 – Topical subtree of a topic having the relevant words "data" and "mining", with the common subtrees highlighted.

Example of desired output In the previous section, we have shown how to map a topic μ to a concept c, more precisely to a topical subtree whose root concept is c. Among all the possible topical subtrees of the concepts $c_i \in \delta(\mathcal{C})$, we aim to identify the topical subtree that includes at least one sense for as many of the topic's words as possible, while having a root concept as specific as possible. The idea is to map the topic to a dense and compact concept structure, while not loosing too much of the topics meaning. Figure 6.10 shows the desired mapping for the previously mentioned example. The topic μ is defined by a list of words, ordered decreasingly by their score. We select "mining" and "data" as the relevant words for μ (highlighted in the word list). Each of the relevant words is mapped to their senses in the concept hierarchy ((i) "mining" is mapped to "metal mining" and "data mining" and (ii) "data" is mapped to "data mining" and "data types"). Therefore the concept set related to the topic μ is { "metal mining", "data mining", "data types"}. The topical subtree of the topic μ is the entire subtree shown in Figure 6.10. We want to chose a denser, smaller substructure of this tree, which covers at least one sense for each of the relevant words of μ. This optimum structure is the topical subtree of the concept "knowledge management" which covers one sense for both "mining" and "data".

Topical tree measures When choosing the optimum subtree, we are facing the old Machine Learning dilemma of choosing between precision and recall. In our context, we define the precision as the specificity of the root concept of the topical subtree in the concept hierarchy. Concepts close to the root of the hierarchy are very general and, thus, have a low specificity. The specificity is also dependent on the distance between the concept and the topic's relevant words (the subtree's height). The recall is defined in this context as the coverage of the topical tree: the proportion of the topic's relevant words that have at least a sense in the topical subtree. Both specificity and coverage need to be maximized.

The *specificity of the topical subtree of the concept* c is dependent on depth of c in the concept hierarchy and the subtree's height. *The depth* ($depth(c)$) is defined as the normalized distance between the concept c and the root of the hierarchy. When the concept c is found deep into the concept hierarchy (depth close to 1), it means that the topical subtree of the concept c is very specialized, which means that the concepts in the subtree have the tendency of being specialized. Conversely, when c is close to the root of the hierarchy (depth close to 0), concepts in the topical tree have the tendency of being general. Formally, the depth $\in [0, 1]$, needs to be maximized and has the following formula:

$$depth(c) = \frac{d(c, c_0(\mathcal{C}))}{\max\limits_{c_j \in \delta(\mathcal{C})} (d(c_j, c_0(\mathcal{C})))} \tag{6.6}$$

where $c_0(\mathcal{C})$ is the root concept of the entire concept hierarchy. *The iheight* (inverse height) of the topical subtree is inversely proportional with the maximum distance between the topical subtree's root concept c and the leaves, the topical related concepts (concepts that are senses of the topical relevant words), normalized to the topical subtree's height (in the entire concept hierarchy). The idea is to assess how general is the concept c compared to the senses of the relevant words. A topical tree with a high *iheight* is very specific. When the *iheight* is low, then the tree is very general. Formally, the *iheight* $\in [0, 1]$, needs to be maximized and has the following formula:

$$iheight(\mu, c) = 1 - \frac{\max\limits_{c_\mu \in \delta(\mu)} (d(c, c_\mu))}{\max\limits_{c_\mu \in \delta(\mu)} (d(c_\mu, c_0(\mathcal{C})))} \tag{6.7}$$

where $\delta(\mu)$ is the set of concepts related to the topic μ. Finally, we define the specificity as a weighted sum of height and depth, in order to allow the fine tuning of the two components. When the weight of the *depth* (ω_d) is high, then the evaluation algorithm tends to map topics to topical subtrees which are deep in the concept hierarchy. When the weight of the *iheight* (ω_h) is high, than the topical subtrees have the tendency of having few levels (they are compact). Formally, $spec, \omega_d, \omega_h \in [0, 1]$, $\omega_d + \omega_h = 1$, *spec* needs to be maximized and has the following formula:

$$spec(\mu, c) = \omega_d \times depth(c) + \omega_h \times iheight(\mu, c) \tag{6.8}$$

We define *the coverage of the topical subtree of the topic* μ *and concept* c as:

$$cov(\mu, c) = \frac{|\{w | w \in P(\mu)\,,\, \delta(w) \cap \delta(\mu, \mathcal{C}_c) \neq \emptyset\}|}{|P(\mu)|} \tag{6.9}$$

where $P(\mu)$ is the set of relevant words of the topic μ, $\delta(w)$ is the set of senses (concepts) associated with the word w in the concept hierarchy $\mathcal{C}$ and $\delta(\mu, \mathcal{C}_c)$ is the set of concepts in the topical subtree of the topic μ and concept c. The coverage $\in [0, 1]$ and needs to be maximized.

Topic fitness function Using the specificity and the coverage, the strength of the relation between a topic and concept subtree can be quantified. The used concept hierarchy is a semantic resource, in which small distances mean semantically close concepts. Therefore, a compact, dense subtree structure means that all concepts in the structure are semantically very similar. A topical tree with a high specificity and high coverage is a tree in which all concepts are semantically close, which is specific and which covers most of the relevant words of a topic. We define the general score of a topical subtree of a given concept c as the weighted sum of the specificity and coverage, to allow the fine tuning of the two components. Note that other aggregating formulas can be used (though not tested), like the classical F_{score}. Formally, $\phi, \omega_{spec}, \omega_{cov} \in [0, 1]$, $\omega_{spec} + \omega_{cov} = 1$, ϕ needs to be maximized and has the following formula:

$$\phi(\mu, c) = \omega_{spec} \times spec(\mu, c) + \omega_{cov} \times cov(\mu, c) \tag{6.10}$$

We define as the "optimum" topical subtree of a topic μ, the subtree of a concept c_{opt} that maximizes $\phi(\mu, c_\mu)$. The concept c_{opt} chosen as the root of the topical subtree is defined as:

$$c_{opt}(\mu) = \underset{c_\mu \in \mathcal{C}_\mu}{argmax}(\phi(\mu, c_\mu)) \tag{6.11}$$

where $\mathcal{C}_\mu$ is the topical subtree of topic μ in the entire concept hierarchy. In practice, the optimum subtree is found by performing a tree search starting from $c_0(\mathcal{C})$ (the roof of the concept hierarchy) and following the specialization relations (*ergo* the hyponymy relation in the case of WordNet) in the tree.

We define the topic fitness score as the score obtained by its optimum topical subtree ($\phi(\mu, c_{opt}(\mu))$). A topic with a high fitness score is a topic with a high semantic cohesion, since its relevant words can be mapped onto a compact semantic concept structure. The semantic cohesion is the degree in which the most relevant words of a topic are similar in meanings. The general score obtained by a set of topics extracted from a document collection is the average fitness score of individual topics. The fitness function permits to leverage semantic knowledge under the form of a concept hierarchy to evaluate the semantic cohesion of topics extracted from a document collection.

6.4.2.3 Experimental Evaluation

In this subsection, we briefly present some experiments and results that can be obtained with this system. The purpose of the experiments is to prove the connection between the proposed automatic concept-based topic evaluation and the way the human mind judges topics. The performed experiments are still very exploratory, being performed only two datasets, with a limited range of values for parameters ($\omega_d = \omega_h = \omega_{spec} = \omega_{cov} = 1$ and $k \in \{30, 50, 100, 200, 300\}$). We choose to present here a part of the results that we have shown in [Musat *et al.* 2011b]. While the experiments still need improvement (*e.g.*, more

datasets, varying the parameters, trying other distances between concepts, comparison with other methods [Newman *et al.* 2010] present in literature *etc.*), we consider these first results encouraging since they show that our proposed method achieves the emulation of human judgement.

Datasets and protocol The experiments were performed on two datasets: the `Suall11` [9] dataset and a custom-made dataset [10], containing economic news articles. `Suall11`, initially used in [Wang & McCallum 2006], is a general dataset on American history. The economic dataset was built from publicly available Associated Press articles published in the Yahoo! Finance section. A total of 23986 news broadcasts which had originally appeared between July and October 2010 were gathered.

The experimental protocol was devised to assess the validity of our assumption that mapping topics to concept subtrees and measuring the strength of mapping relation (topic fitness function) is a good indicator of topical coherence. The aim is capturing the correlation between the human verdict and the calculated topic fitness. The **Latent Dirichlet Allocation** [Blei *et al.* 2003] (presented in Section 6.3.2 and built into the Mallet suite [McCallum 2002]) was used to generate the topics and the word probabilities scores used to detect the relevant words for each topic. Human evaluations was performed by 37 external judges and followed a similar framework as the one employed in [Chang *et al.* 2009a]. The evaluators were asked to extract the unrelated (spurious) words from a group containing the 5 most relevant words associated with one topic and an additional spurious word. One or more unrelated words were chosen for each group, by each evaluator.

Evaluation of two dimensions The analyzed topics are separated into high fitness scoring and low fitness scoring, on the basis of the automatically calculated topic fitness. The **relevant set** contains the 10 highest scoring topics, while the **irrelevant set** contains the 10 bottom scoring topics. The aim is to see whether an improvement of the spurious word detection is visible from one category to the other. For each topic, two spurious words were chosen. The Kullback-Leibler divergence [Kullback & Leibler 1951] is used to select the closest and farthest topics and one of their relevant words were used as the spurious word. Intuitively, spurious words from close topics are more difficult to detect than the one from distant topics. Therefore, the experiments have two dimensions: to assess if the spurious word detection improves (i) between topics with high fitness scores and topics with low fitness scores and (ii) between spurious words originating from close topics and from distant topics.

Based on the evaluators responses, we calculate $\overline{hit}_+$ (the average percentage of correctly identified spurious words for topics in the relevant set) and $\overline{hit}_-$ (the average percentage of correctly identified spurious words for topics in the irrelevant set). Each indicator is calculated in two situations: (i) when the spurious word is chosen from a close topic and (ii) when the spurious word is chosen from distant topics. Table 6.4 presents the obtained results, and the relative gain between (a) $\overline{hit}_+$ and $\overline{hit}_-$ and (b) the values for close and distant topics.

9. Download: `http://www.gutenberg.org/dirs/etext04/suall11.txt`
10. Download: `http://eric.univ-lyon2.fr/~arizoiu/files/economic_corpus_AP.tar.bz2`

The results show that the source of the spurious word has a crucial importance: the detection rate is almost double (an average increase of close-distant gain of 92.7%) when the spurious word is chosen from distant topics, instead of close topics. This proves that spurious words from distant topics are easier to identify that those chosen from close topics. The detection rate also increases between topics from the relevant set and topics from the irrelevant set. The relative $\overline{hit}$ gain in the detection rate varies between 6.93% to 66.76%. The fact that the detection rate is consistently better for topics with high fitness score gives a first positive evaluation of the fact that the topic-concept mapping succeeds in emulating the human semantic judgment. Furthermore, we observe a significant difference between the relative $\overline{hit}$ gain corresponding to spurious words from close topics and spurious words from close topics. When the spurious words originates from distant topics, it is easier to identify, regardless of the quality of the topic at hand. Whereas, when the spurious words originates from a close topic, it is harder to spot out. It is this measure that actually shows the fact that our concept-topic mapping emulates the human judgment: when topics are coherent, humans tend to better identify the spurious word and our system achieves to identify these coherent topics.

Table 6.4 – Spurious Word Detection Rates.

Dataset	Type	$\overline{hit}_+$	$\overline{hit}_-$	Gain $\overline{hit}$
AP	Close	0.37	0.27	39.33%
	Distant	**0.69**	0.65	6.93%
	Gain close-distant	86.49%	140.74%	
Suall	Close	0.51	0.3	66.76%
	Distant	**0.75**	0.59	28.55%
	Gain close-distant	47.06%	96.67%	

In conclusion, these results show that (i) non-related words from distant topics (from the Kullback-Leibler divergence point of view) are easier to detect than spurious words from close topics and (ii) spurious words inserted into topics with high fitness score (from the concept-based evaluation point if view) are easier to detect. This proves that (i) the Kullback-Leibler divergence can be used as a measure of the semantic distance between two topics expressed as a probability distribution over words and (ii) the proposed topic-concept mapping and evaluation measures succeed in capturing and quantifying the semantic cohesion of a topic.

6.5 Applications

In this section, we discuss how the topic extraction and evaluation techniques, presented in this chapter, can be used in other applications. More precisely, the reader will find, in Section 6.5.1, how the topic-concept mapping, alongside with the measures defined in Section 6.4.2, can be used for improving topic. We examine a method for automatically detecting and removing spurious words from a topic's description. In Section 6.5.2, we present a very brief presentation of the automatic ontology learning field as well as a couple

of hints of how topics extracted from a collection of documents can be used in ontology learning.

6.5.1 Improving topics by Removing Topical Outliers

After presenting how topics can be created (in Sections 6.3.1, 6.3.2 and 6.4.1) and evaluated (in Section 6.3.4 and 6.4.2), a natural follow-up is topic improvement. We use the same general framework which is described in Section 6.4.2 (p. 148). A semantic resource under the form of a concept hierarchy is used and a topic-concept mapping is created, passing through words. Topics are associated with topical subtrees, which contain all the generalizations of the senses of their most relevant words.

The idea of the topic improvement system is to create a projection of the topic as a whole onto the given ontology and decide which part of the topic, if any, is separated from the others. The underlying assumption is that the understandability of the given topic can be improved by removing the parts which are unrelated from a human perspective. The presented system is designed to improve individual topics according to their semantic cohesion. We use an established simplified representation of topics, used in most of the literature, which is the list of most relevant words. To improve topic readability and meaningfulness, we prune the outliers from the related word set, using the concept hierarchy. Topical outliers are the words that are unrelated to the other words from a semantic perspective. After the pruning, the remaining relevant words are more inter-related as a set and confer more meaning to the user. The core of this work is establishing the conceptual context of a single given topic, as shown in Section 6.4.2. We detect which concepts from the used ontology are relevant to the topic as a whole and choose the topical words unrelated to those concepts as the outliers to be eliminated.

Methodology In order to detect the topical words that are unrelated to conceptual context created by the others, we must first identify the related concepts. In Section 6.4.2, we have shown that for any given topical, multiple topical subtrees of a specific concepts can be constructed. Each of topical subtrees were assigned a general topical subtree score, given by Equation 6.10 (p. 156). In the evaluation process, we have selected the "optimum" topical subtree as the one having the highest topical subtree score. In order to detect topical outliers, we do not choose one "optimum" subtree (belonging to just one concept), but we choose $l \in \mathbb{N}$ topical subtrees that have the highest scores. If we are to make a parallel, the highest ranking topical subtrees are like the axes in of Principal Component Analysis (PCA) [Dunteman 1989]. Just like PCA's axes, the topical subtrees contain decreasing amounts of semantic information. Selecting the first few highest ranking subtrees permits to map most of the semantic information contained in the topic. We define the topical outliers as the words not covered by the reunion of the l highest scoring topical subtrees for a given the topic μ. l is a parameter of the system.

In Figure 6.11, we present a simplified version of the topical subtree for the topic μ("mining", "data", "error", "ontology", "learning"). For $l = 2$, the highest two scoring topical subtrees are those of concepts "knowledge management" and "semantic web" (highlighted in orange). The topical subtree of the concept "knowledge management" contains senses for the words "mining" and "data", whereas the topical subtree of the concept "semantic web"

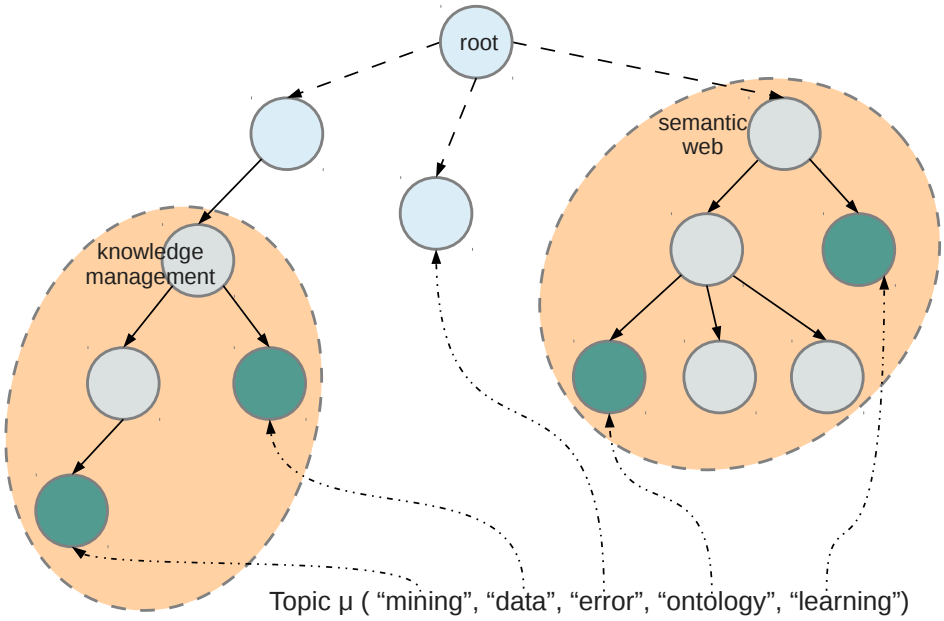


Figure 6.11 – Removing topical outliers using a topic-concept mapping.

contains the senses of “ontology ” and “learning”. The only word that does not have at least one sense in the reunion of the two highest scoring topical trees is “error”, which is therefore declared an outlier and eliminated.

We have, therefore, designed a method for detecting topical outliers, which are words semantically unrelated to the rest of the topical relevant words. This method can be used to improve the topics, by providing the user with a more semantically coherent list of relevant words. But the usages are not limited to topics. In fact, this method can be used to filter outliers from any kind of word list, as long as most of the words are semantically related among themselves. Such an approach can be used to filter the terms which are semantically unrelated with a specific domain concept from a list of automatically extracted terms. As we show in the next section, the manual term filtering is one of the bottlenecks of Ontology Learning.

6.5.2 Concept Ontology Learning

Ontologies are collections of concepts linked together through a set of relations. Ontology construction is a complex and time consuming process requiring the knowledge of highly specialized experts. To overcome this knowledge acquisition bottleneck, Ontology Learning techniques for the automatic construction of ontologies have been proposed. The process of Ontology Learning from Text involves Natural Language Processing techniques and, as shown in Figure 6.12, can be divided into five main steps, also known as the **Ontology Learning Layer Cake** [Buitelaar *et al.* 2005]:

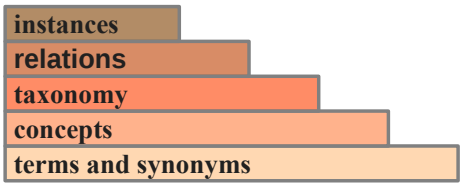


Figure 6.12 – Schema of the Ontology Learning Layer Cache.

1. extraction of domain terminology and synonyms from a corpus of documents (the *terms and synonyms layer*);
2. definition of main concepts on the basis of the detected relevant terms and the classes of synonyms (the *concepts layer*);
3. structuring of concepts into taxonomy (the *taxonomy layer*);
4. definition of non-taxonomic relations between concepts (the *relations layer*);
5. population of the ontology with concepts and relations instances (the *instances layer*).

An overview of the state-of-the-art of ontology learning and the proposed solutions for each step can be found in [Cimiano *et al.* 2006]. The main approach toward learning concepts and their taxonomy (the hierarchical relations between concepts) is **Conceptual clustering** [Michalski & Stepp 1983], an unsupervised machine learning technique closely connected to unsupervised hierarchical clustering. Examples of algorithms developed for this purpose are the well-known **COBWEB** [Fisher 1987] and the more recent **WebDCC** [Godoy & Amandi 2006]. We take a look into alternative methods and discuss the usage of topic extraction at the two bottom layers of the ontology cache: the *terms and synonyms layer* and the *concepts layer*.

Work at the *terms and synonyms layer* Topic extraction systems have already been used at the *terms and synonyms layer*, where the challenges are (i) extracting relevant terms that unambiguously refer to a domain-specific concept and (ii) dealing with disambiguation. The literature provides many examples of term extraction methods [Wong *et al.* 2008, Wong *et al.* 2009] that could be used as a first step in ontology learning from text, but the resulted list of relevant terms needs to be filtered by a domain expert [Cimiano *et al.* 2006]. The topic-concept mapping presented in Section 6.4.2 and the topic improvement approach presented in the Section 6.5.1 allow further automatization of this process. Semantic outliers can be detected by using a general purpose semantic resource, such as WordNet, therefore simplifying or completely eliminating the supervision of the field expert.

Disambiguation deals with choosing, considering the property of polysemy, the right meaning for a word in a given context. Most of today's word sense disambiguation algorithms, like the one in [Lesk 1986], rely on usage of synonym sets. The literature presents two main approaches towards finding synonyms [Buitelaar *et al.* 2005]:

- algorithms that rely on readily available synonym sets such as *WordNet* [Miller 1995] or *EuroWordNet*[11] [Turcato *et al.* 2000, Kietz *et al.* 2000];
- algorithms that directly discover synonyms by means of clustering. These algorithms are based on statistical measures mainly used in Information Retrieval and start from the hypothesis that terms are similar in meaning to the extent in which they share syntactic contexts [Harris 1968]. text clustering places documents which share the same context into the same group, which in turn leads to synonyms being placed under the same topic.

As terms have different meaning depending on the context, it is only natural to allow them to be part of more than one group. In this case, the clustering algorithm can be used to find synonyms, but also for term disambiguation (choosing between the different meanings).

11. http://www.illc.uva.nl/EuroWordNet/

The crisp text clustering solutions presented in Section 6.3.1 have the inconvenient that they output a crisp partition, where each document belong to only one group. These approaches can be used for regrouping synonyms, but they cannot be used for disambiguation. An overlapping solution, as the one presented in Section 6.4.1, would address the disambiguation problem, allowing terms to be in more than one cluster. Terms with multiple meanings can be regrouped together with their synonyms, for each of their meanings.

The *concepts layer*: evolving topics to concepts A topic is not a concept since it represents the abstraction of the idea behind a group of texts rather than a notion in itself. While the difference between the two is subtle, evolving a topic into a fully fledged concept is still to be achieved. The literature does not provide any largely accepted solution. Of course, the simplest way to do it is to have a human expert manually evolving the topics into concepts by adding relations and building the structure of the ontology. But in the long term, the objective is to completely automatize the ontology building process. That is why relations need to be found in a human-independent way.

Some of the recent topic extraction algorithms already made the first steps towards this objective. **hLDA** [Blei *et al.* 2004] outputs an hierarchy of topics, which can provide, to a certain extent, the hierarchical relation between concepts. Other algorithms, like **cLDA** [Blei & Lafferty 2006a], obtain a correlation structure between topics by using the logistic normal distribution instead of the Dirichlet. Some authors consider that a hierarchy of topics can already be considered an ontology. [Yeh & Yang 2008] extract the topics from the text, using LSA, LDA or pLSA. Then they regroup them into super-topics, using a hierarchical agglomerative clustering using the cosine distance. They consider that "because the latent topics contain semantics, the clustering process is regarded as some kind of semantic clustering". In the end, they obtain an ontology in OWL. Topic to concept passage is also related to other perspectives, such as reconciling the similarity-based dendrograms built by traditional Hierarchical Agglomerative Clustering and the concept hierarchies used in Format Concept Analysis [Ganter & Wille 1999]. The recent work of [Estruch *et al.* 2008] proposes in this line an original framework to fill the gap between statistics and logic. In Section 6.4.2 (p. 148), we presented a method for mapping topics to a concept hierarchy. Such a mapping can be used as intermediary step from a general purpose semantic resource, such as WordNet, to a domain specific ontology by passing though topics.

6.6 Conclusion and future work

In this chapter, we have focused on one of the core research challenges of this thesis: leveraging semantics, and applied it to dealing with textual data. More precisely, we are interested in the tasks of (i) topic extraction, (ii) topic labeling and (iii) topic evaluation. For the topic extraction task, we have proposed an overlapping text clustering-based solution, that authorizes textual documents to be associated with more than one topic, depending on their approached subjects. For the topic labeling task, we associate to topics humanly-comprehensible labels, which are chosen from a candidate set of frequent complete phrases. For the topic evaluation task, we have proposed a system that aims at emulating the human judgment of topics by using an external concept hierarchy (*e.g.*, WordNet).

Conclusion Topics are the central point of our work with the textual dimension. A topic is a general idea behind a group of similar documents. We have shown our reader an overview of the different approaches present in the literature which extract and evaluate topics. We argument that the property of polysemy of words and the fact that a single textual document can approach multiple topics are crucial in topic extraction. Therefore, we present two novel systems, one for extracting topics and the other for evaluating topics. The topic extraction system is based on an overlapping clustering algorithm and assigns for each of the extracted topics a humanly readable name using a suffix array-based keyphrase extraction algorithm. The topic evaluation system is based on an external semantic resource under the form of the concept hierarchy. The core of the system is the topic-concept mapping, in which each topic is associated with a topical subtree of a certain concept c. This structure is a concept subtree having (i) c as its root and (ii) the senses of the most relevant words of the given topic as leaves. It contains, therefore, all the generalizations of the senses of relevant words which are simultaneously specializations of the concept c. We search in the concept hierarchy for the topical subtree which has the largest coverage (contains at least one sense for as many as the possible of the topic's relevant words) and is as specific as possible. We propose an evaluation measure which quantifies the semantic coherence of a topic. Towards the end of the chapter, we have presented two applications of the proposed systems: the detection of outliers from a list of words (outliers are the words semantically unrelated with the others) and the usage of topics in automatic Ontology Learning.

Practical and applied context Some of the work concerning topic extraction was initially developed during my Masters research internship. The further extensions were published in the proceedings of a French national conference [Rizoiu *et al.* 2010], while the application of topic extraction to ontology learning was published in a book chapter [Rizoiu & Velcin 2011]. The topic evaluation methodology was developed in collaboration with the Computer Science department of the Polytechnic University of Bucharest, and more precisely, the PhD research internship of Claudiu Cristian Muşat at the ERIC laboratory. It was proposed in the proceedings of an international conference [Musat *et al.* 2011b]. The application of outlier detection was also published in the proceedings of an international conference [Musat *et al.* 2011a].

The work presented in this chapter has the particularity of being in close connection with the different research projects in which I was involved (as discussed in Section 6.1) and the applied part of my work. The proposals discussed in Section 6.4 either are or they will soon be integrated into `CommentWatcher`, our open-source web-based platform for analyzing online discussions on forums. `CommentWatcher` will be described in detail next, in Chapter 7.

Future work We identify for our work with the textual dimension some development tracks. We consider applying the proposed topic-concept mapping to other fields of natural language processing, such as (i) integrating semantic knowledge into the topic extraction algorithm by using the proposed topic-concept mapping, (ii) automatic filtering of unrelated terms at the *terms and synonyms layer* or (iii) word sense disambiguation in the Ontology Learning Layer Cache.

Given the intimate link between our work concerning text and our applied work (most notably `CommentWatcher`), many of our future perspectives are linked to practical aspects.

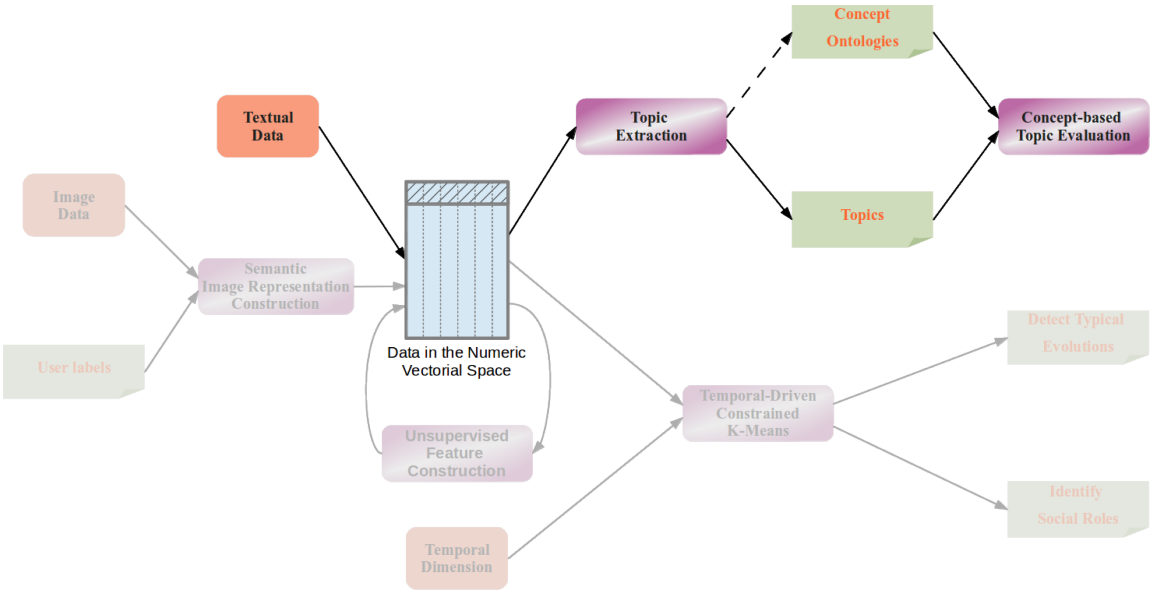


Figure 6.13 – Streamlined schema showing how the contributions in this chapter can be combined with those in previous chapters.

We have currently undergoing work to integrate the semantic-aware topic evaluation into `CommentWatcher`. Once this is achieved, we plan to perform a thorough comparison of our topic extraction algorithm with other state of the art extraction algorithms, from a semantic evaluation point of view. Furthermore, given `CommentWatcher`'s versatile parser architecture, we plan to test the proposed concept-based evaluation method on other, larger and more diverse textual datasets. We also plan to (i) complete the study of the influence of parameters (*e.g.*, the different weights defined in Table 6.3, p. 150), (ii) try other distances to compare the semantic similarity of concepts and (iii) compare with existing Word Sense Disambiguation systems.

Articulation with the previous work Conceptually, the work presented in this chapter articulates with the work of previous chapters as shown in Figure 6.13. The work presented in previous chapters is presented with faded colors. The joining point is, as in the case of the previous chapters, the passing of the data though a semantic-aware numeric description space. Furthermore, while in this chapter we presented topics and concepts from a textual point of view, none of the two is actually limited to words. For example, the pLSA topic extraction technique has been used [Kandasamy & Rodrigo 2010] with visual words when dealing with images. Concept ontologies are represented using words, but concepts represent abstract notions. Therefore, it is foreseeable to use such knowledge repositories to treat other types of complex data. For example, in Section 2.1.2 (p. 13) we give examples of algorithms dealing with images and text, therefore, images could be linked to concepts by passing through text.

Chapter 7

Produced Prototypes and Software

Contents

7.1 Introduction

As previously hinted, my theoretical research work was constantly doubled by an applied research work. Scripts were developed for the experimental part of each of the approaches presented in previous chapters. These are usually just *proofs-of-concept*, written in scripting languages (like *Octave*[1]). `CommentWatcher`, the most prominent produced software and the one presented in this chapter, is an open source tool aimed at analyzing discussions on web forums. Constructed as a web platform, `CommentWatcher` features automatic fetching of the forums using a versatile parser architecture, topic extraction from a selection of texts and a temporal visualization of extracted topics and the underlying social network of users. It is aimed at both the media watchers (it allows quick identification of important subjects in the forums and user interest) and the researches in social media (who can use to constitute temporal textual datasets). `CommentWatcher` is currently used in the Crtt-Eric research project to study the evolution of specialized discourse in the domain of nuclear medicine, while taking into account the temporal evolution and the different involved populations (*e.g.*, doctors, nurses, patients). It is also planned to be used to construct a dataset for studying the detection of behavioral roles by using temporal-driven constrained clustering (as seen in Section 3.6 (p. 52)).

1. http://www.gnu.org/software/octave/

Features of CommentWatcher `CommentWatcher` answers to a series of limitations concerning online forum discussions (some of which are detailed in Section 7.2): (a) concerning discussion forum benchmark datasets and (b) concerning existing software solutions. (a) We identify the following limitations concerning discussion forum benchmark datasets: (i) scarcity of forum benchmarks, (ii) the existing forum benchmarks are issued from a single forum website, therefore, it is not possible to study inter-website user behavior, (iii) the structure of forum websites is continuously changing, rendering forum parsers useless and (iv) the copyright of forum data is unclear, therefore hindering the sharing in the research community. (b) The limitations concerning existing software solutions are that (i) they are proprietary solutions, unusable for research purposes and (ii) they do not deal with crawling the forum sources and need to be supplied directly with formatted data.

`CommentWatcher` addresses these limitations by featuring a modular parser architecture, capable of handling the ever-changing structure of websites. Furthermore, it is open source, meaning that it can be freely distributed. It can be used to solve the problem of content copyright, since only the tool is distributed and the forum benchmark can be easily reconstructed locally by each researcher. To our knowledge, `CommentWatcher` is the only solution dedicated to online discussion forums that integrates (a) forum parsing, (b) topic extraction and visualization and (c) online social network inference and visualization.

Experimenting with `CommentWatcher` A public demo installation of `CommentWatcher` is available [2]. The reader is able to interact with `CommentWatcher` in a normal browser window, through the tools web interface. The tool itself is hosted and executed on its dedicated machine, located at the ERIC laboratory. The reader can, first-hand, experience the tools capabilities by (a) seeing how multiple discussion forums can be fetched by searching the web using keywords, (b) applying topics extraction algorithms and tweaking their parameters, (c) visualizing the extracted topic as a expression cloud and their temporal evolution and (d) visualizing the social network constructed starting from the initial forums. A short presentation movie is also available [3], showing the main features of `CommentWatcher`.

CommentWatcher's history The platform evolved from a simple prototype into a fully-fledged academic software, due to the needs and purposes of the different research projects (detailed in Appendix A) in which I was involved. It started in the context of the applied research project Conversession, in which the creation of a start-up enterprise [4] was involved. The purpose of the project was *Online Media Watching*, and more precisely focused on news discussion forums. The resulted prototype is called `Discussion Analysis` [5], a Java desktop application that features (i) fetching of discussion forums from 2 French websites (`www.liberation.fr` and `www.rue89.fr`) and 2 English sites (`www.huffingtonpost.com` and `forums.sun.com`), (ii) textual preprocessing (as shown in Section 6.2.1 (p. 128)) and (iii) topic extraction using the system described in Section 6.4.1 (p. 141). The development of the prototype was continued in the context of projects Eric-Elico, Crtt-Eric and ImagiWeb. The interest for each of these projects was to constitute discussion forum

2. Online here: `http://mediamining.univ-lyon2.fr:8080/CommentWatcher/`
3. Presentation website: `http://mediamining.univ-lyon2.fr/commentwatcher`
4. `http://www.conversationnel.fr/`
5. Download here: `http://mediamining.univ-lyon2.fr/rizoiu/files/discussion-analysis.jar`

datasets and to analyze the discussion topics. The platform evolved accordingly. The interface was migrated to a web-based interface to allow simultaneous work of multiple users and a unified discussion database. A new discussion forum fetching module, described in Section 7.3.3 was implemented, which eliminates the dependency of the code base on the website structure. Support for new topic extraction algorithms (details in Section 7.3.4) was added. The visualization module (described in Section 7.3.5) was added for temporal topic visualization and social network visualization. The development of the platform continues at this date: a force-directed graph drawing is being implemented, as well as a temporal topic model system.

Planning of the chapter The remainder of this chapter is structured as follows. In Section 7.2, we present an overview of applied discussion forum analysis: the current limitations, some of the solutions existing in the literature and an overview of our proposed approach. In Section 7.3, we present the general design and detail the different components of `CommentWatcher`. Section 7.4 gives the license under which `CommentWatcher` is released and describes how to obtain the software. We conclude in Section 7.5 and we present the work we are currently undertaking, and we plan future developments.

7.2 Discussion Forums

`CommentWatcher`'s vocation is to analyze online discussions. The Web 2.0 has changed the way users discuss with other users. One of the preferred online discussion environments are the web forums. Users can react, post their opinions, discuss and debate any kind of subjects. The forums are usually thematic (*e.g.* Java programming forums[6]) and new users have access to the past discussion (*e.g.* solutions posted by other users to a specific problem). Therefore the users become full collaborative participants in the information creation process. The subjects of discussion between readers are very dynamic and the overall sum of reactions gives a snapshot of the general trends that emerge in the user population. At the same time, the way users reply one to another suggests an underlying social network structure. The forum's "reply-to" structural relations can be used to add links between users. Other types of relations can be added, like the name and textual citations [Forestier *et al.* 2011]. Furthermore, based on such social networks constructed from web forums, adapted graph measures can be used to detect user social roles [Anokhin *et al.* 2012]. In Section 3.6 (p. 52), we have shown how a temporal-driven constrained clustering technique, that we proposed in Section 3.4 (p. 34), can be applied to detect behavioral roles in such a social network constructed from web forums.

7.2.1 Current limitations

These forum data are still ill explored, even if they represent an important source of knowledge. News articles analysis and micro blogging (*e.g.* Twitter) analysis receive a lot of attention from the community. There are available tools that perform the analysis of news media [Amer-Yahia *et al.* 2012], but without treating the social network aspect. Other tools

6. http://www.javaprogrammingforums.com/

concentrate on analyzing and visualizing the social dynamics [Guille *et al.* 2013] or detect events [Marcus *et al.* 2011] based on twitter data. To the best of our knowledge, there are no publicly available tools that treat forums, while inferring a social network structure.

Another limitation concerns the forum benchmarks. There are a multitude of general purpose information retrieval datasets (*e.g.* the `ClueWeb12 dataset`[7] of project Lemure) and of Twitter datasets (*e.g.* the `infochimps collections`[8]). But dedicated web forum benchmark datasets are scarce. Those that exist are usually issued from a single forum website (*e.g.* the `boards.ie Forums Dataset`[9] based on boards.ie website or the `Ancestry.com Forum Dataset`[10], based on ancestry.com website). This is due to the diverse and ever changing structure of the websites hosting the discussions and copyright problems. Each host website has its own license on the user-produced data, which is not always clearly stipulated. This leads researchers to develop their own house-bred parsers and create their own datasets. These datasets are rarely shared with the community, which poses problems when testing new proposals and comparing to existing approaches.

7.2.2 Related works

Several tools intending to extract knowledge from on-line discussions have been proposed in the recent years.

MAQSA [Amer-Yahia *et al.* 2012] is a system for social analytics on news that allows its users to define their own topic of interest, in order to gather related articles, identify related topics, and extract the time-line and network of comments that show who commented which article and when.

Eddi [Bernstein *et al.* 2010] offers visualizations such as time-lines and tag clouds of topics extracted from tweets using a simple topic detection algorithm that uses a search engine as an external knowledge base.

OpinionCrawl[11] is an on-line service that crawls various web-sources – such as blogs, news, forums and Twitter – searching for a user-defined topic and then presents key concepts as a tag cloud, provides a visualization of the temporal dynamics of the topic and performs a sentiment analysis.

SONDY [Guille *et al.* 2013] is an open-source plateform for analyzing on-line social network data. It features a data import and pre-processing service, a topic detection and trends analysis service, as well as a service for the interactive exploration of the corresponding networks (*i.e.*, active authors for the considered topic(s)).

The aforementioned tools are limited for various reasons. They are either proprietary softwares and thus can't be extended for scientific purposes or can't directly crawl web sources and can only be used to analyze formatted datasets provided by the user. `CommentWatcher` intends to provide researchers with an open-source extendable tool that permits to crawl the web and build datasets that suit their needs.

7. http://lemurproject.org/clueweb12/specs.php
8. http://www.infochimps.com/collections/twitter-census
9. http://www.icwsm.org/2012/submitting/datasets/
10. http://www.cs.cmu.edu/~jelsas/data/ancestry.com/
11. http://opinioncrawl.com

7.2.3 Introducing CommentWatcher

We address these issues by introducing `CommentWatcher`, an open source web-based platform for analyzing discussion on web forums. `CommentWatcher` was designed having in mind two types of users: the forum analyst, who seeks to understand the main topics of discussion and the social interactions between users, and the researcher who needs a benchmark to test his/her proposed approaches. Using `CommentWatcher`, the researcher can create forum discussions benchmarks without worrying for copyright issues, since the platform is open source and the text itself is not distributed (each researcher can locally recreate the benchmark dataset).

When building `CommentWatcher` we address the challenges that arise from retrieving forums from multiple web sources. Not only these sources are profoundly heterogeneous in structure, but they tend to change often and render parsers obsolete. We implement a parser architecture which is independent from the website structure and allows simple on-the-fly adding of new sources and updating the existing ones. `CommentWatcher` also supports mass fetching of forums from supported sources by using keyword search on the internet, extracting discussion topics, creating the underlying social network structure of users and visualizing it in relation with the extracted topics.

7.3 Platform Design

In this section, we describe the software technologies used in developing `CommentWatcher`, the general architecture and the different components to highlight their aim and the way they interact.

7.3.1 Software technologies

`CommentWatcher` is written using Java Servlets for server-side computing and Java Server Pages for the dynamic webpage generation. The support for fetching forums discussions from websites is implemented using the XLS Transformation technology. New websites can be added dynamically, without changing the source code. A MySql database is used for storing forum structure, user characteristics and the text. The visualization is performed client-side into a Java Applet.

7.3.2 Platform architecture

The application has three main modules, interconnected as shown in Figure 7.1. The *fetching module* deals with downloading the forums, parsing the web pages and storing the data into the database. Optionally, it can perform a keyword web search to find forums that can be fetched. The *topic extraction module* performs topic extraction using an algorithm implemented as a library on a selection of forums. The *visualization module* has two views: (i) topic visualization as an expression cloud and as a temporal evolution graphic and (ii) social network visualization.

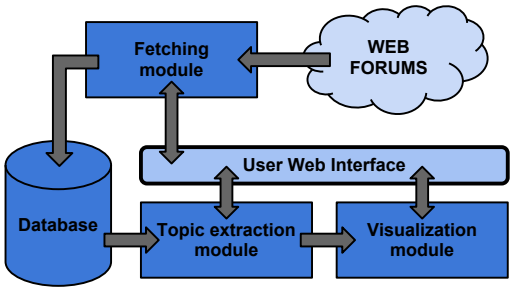


Figure 7.1 – `CommentWatcher`: overview of the platform's architecture.

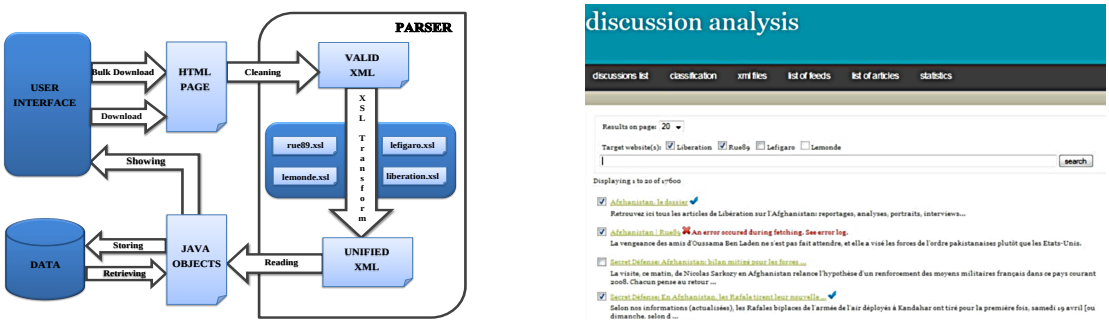


Figure 7.2 – The design of the fetching module (a) and a screenshot of the keyword mass fetching process (b)

7.3.3 The fetching module

This module deals with downloading, parsing and importing the forum data into the application. The main difficulty when parsing web pages is that the structure of each page is different. What is more, the structure of a certain web page tends to change over time. With `CommentWatcher`, we have designed and implemented a meta-parser, which is independent on the website structure. The actual adaptation of the parser to a specific page is done using an external definition file, implemented in XSLT, a standardized and well documented language. Therefore, adding support for new websites or modifying existing ones boils down to just adding or modifying definition files, without any change in the parser's source code.

The design schema of the fetching module, as well as its interactions with the user interface and the database, are given in Figure 7.2a. The download action specifies the URL of a forum to be downloaded. The bulk download follows the same idea, but a keyword web search is performed using the Bing API and all results from supported websites are downloaded. A screenshot of the keyword web search and mass fetching is given in Figure 7.2b. The specified page will be downloaded in raw HTML format which will undergo cleaning, XSL transformation and deserialization. The process of cleaning implies transforming the HTML document into a well formed XML. In the following step, the XSL transformation is applied to the valid XML document using one of the XSLT definition files of the supported websites. The result of the transformation is an XML document, which uses the same XML

schema for all supported websites. The required data is then deserialized into Java objects, which can be further on stored in and retrieved from the database.

The advantages of implementing such a parsing process are that it is simple, reliable, easy to understand and modify. Furthermore, it does not hard-code the website's structure and it allows adding new supported websites on-the-fly.

7.3.4 Topic extraction and textual classification

This module allows extracting topics from texts from a selection of forums, already fetched in the database. The design is modular, the extraction itself being performed by external libraries. The text from selected forums is prepared and packaged in the format required by the topic extraction library and then passed to the library. The user interface allows setting the parameters for each library. Once the extraction is finished, the results are saved into an XML document, which has the same format for all topic extraction libraries. The XML document contains the expressions associated to each topic and their scores.

At the present, `CommentWatcher` supports two topic extraction algorithms, provided by two libraries: Topical N-Grams [Wang *et al.* 2007] provided by the Mallet Toolkit library [McCallum 2002] and CKP [Rizoiu *et al.* 2010], provided by the CKP library. Topical NGrams is a graphical model algorithms, which models topics as distributions of probabilities over n-grams. CKP, which has already been presented in Section 6.4.1 (p. 141), uses overlapping textual clustering (one text can belong to multiple clusters) and considers each cluster of the partition as a topic. The expressions stored in the XML result document are either (i) the resulted n-grams (for Topical NGrams) or (ii) the frequent expressions (for CKP). Their score is (i) the probability to which an n-gram is associated to a topic (for Topical NGrams) or (ii) $1 - d(e_i, \mu)$, where $d(e_i, \mu)$ is the normalized cosine distance between the frequent expression e_i and the topic's centroid μ (for CKP). Support for new algorithms and libraries can be added easily, but it requires writing adapters for the inputs and outputs.

7.3.5 Visualization

Temporal topic visualization The visualization module is designed to help the user to quickly understand the extracted topics and visualize their temporal evolution. It is the only module that is executed client-side, in a Java Applet. After the XML object resulting from the topic extraction is loaded by the applet, two visualizations are available: the expression cloud and the temporal evolution graphic. Figure 7.3 shows a screenshot with the two visualizations. The expression cloud visualization is similar to the word cloud visualization, which the exception that it uses the expressions generated at the topic extraction module and their sizes are proportional with their score. The temporal evolution graphic portrays the popularity of each topic over the period of time. The time is discretized in a configurable number of intervals, the user posts associated to each topic in each interval are counted and graphics are generated for each forum or for each hosting website.

Social network visualization To facilitate the exploration of the interactions between the members of the forum, we compute a visualization of the underlying social network. The

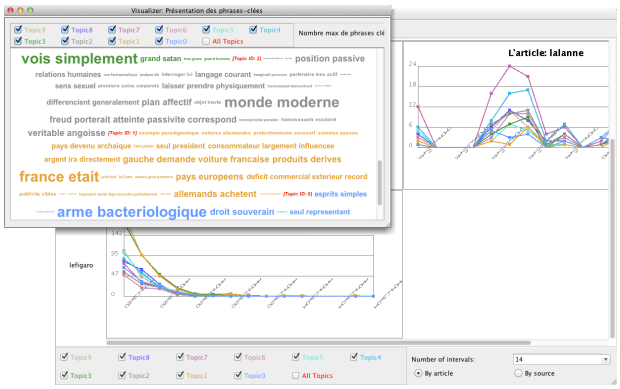


Figure 7.3 – The expression cloud visualization of topics and their temporal evolution.

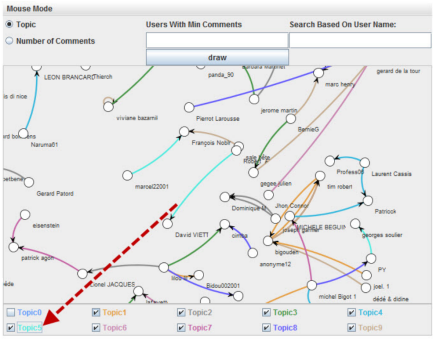


Figure 7.4 – Visualizing the constructed social network, enriched with topical an user features. One can see, *inter alia*, that the reply of "Robert" to "David VIETI" is associated to topic #5.

network is colored according to the topics on which the users are interacting. We construct the social network as a labeled multidigraph, as shown in [Forestier *et al.* 2011]. We map the network nodes on the authors of messages. We add an arc labeled with the topic between two nodes when there is, between the two users, at least one direct reply belonging to the respective topic. We further enrich the network with user's features as the number of posts, the number of topics a user participates in, the number of threads a user participates in, *etc.* Further measures are calculated on the graph, such as the weighted in- and out-degree, the betweenness centrality and the closeness centrality.

Figure 7.4 shows how `CommentWatcher` displays the induced social network. The visualization is created with the Jung Graph Library [12] and is interactive, so nodes can be selected in order to see their features. Relations can also be filtered in order to show only the network corresponding to certain topics.

12. http://jung.sourceforge.net

7.4 License and source code

`CommentWatcher` is released under the opensource license GNU General Public License version 3 (GNU GPL v3) [13]. The individual topic extraction and textual clustering software packages are the objects of their respective licenses. The present version of `CommentWatcher` comes with two Natural Language Processing toolkits: the Mallet Toolkit [McCallum 2002] v2.0.7, released under the open source Common Public License, and CKP [Rizoiu *et al.* 2010] v0.2, released under the GNU GPL v3. The install files and the source code of `CommentWatcher` is available through a public Mercurial repository [14].

7.5 Conclusion and future work

Conclusion This chapter presented `CommentWatcher`, an open source web-based platform for analyzing discussions on web forums. Our tool is designed for both end-users, as well as for researchers. End-users have at their disposal an easy to use, integrated tool that allows retrieving forum discussion from multiple websites, performs topic extraction to identify the main discussion topics and provides an expression cloud visualization to identify the most important expressions associated to each topic. The temporal popularity of topics can be evaluated using an evolution graphic. `CommentWatcher` also features extracting the underlying social network by using the direct citation links between users. The visualization of the social network is interactive, features of nodes can be visualized and relations can be filtered to show only the network corresponding to a certain topic. For researchers, `CommentWatcher` tackles the problem of creating multi source web forum datasets, thanks to its versatile parser which is independent of the structure of webpages. Support for new websites can be added on-the-fly. It can also solve the problem of copyright when sharing forum datasets, since no text is distributed and each researcher can easily recreate the dataset.

Current and future development With the beginning of the CRTT-ERIC project, `CommentWatcher` has officially become one of the academic software supported by the DMD (*Data Mining and Decision*) team of the ERIC laboratory. It is currently the center point of multiple student research and development internships and has recently acquired a dedicated machine. The objects of the ongoing development are (i) a better plotting of the social network, by using force-directed graph drawing, (ii) integrating a temporal topic model algorithm and (iii) importing external data into the discussion database, in order to be able to treat other types of discussions (*e.g.*, chat discussions). As future work, we intend to add a credential mechanism and transform `CommentWatcher` into a multiuser tool and we consider implementing a topic evaluation based on ontologies of concepts, as presented in Section 6.4.2 (p. 148).

13. http://www.gnu.org/licenses/
14. http://eric.univ-lyon2.fr/~commentwatcher/cgi-bin/CommentWatcher.cgi/CommentWatcher/

Chapter 8

Conclusion and Perspectives

Contents

This final chapter of my thesis is dedicated to drawing some general conclusions, briefly presenting the current work and outlining directions for future work. This chapter is structured into four parts: summary of the work, original ideas, conclusions, and outline of future work. In Section 8.1 we present, for each chapter, an outline of the original contributions and their positioning relative to existing methods. The three most important original ideas in our work are singled out in Section 8.2 and shown in the context of the publications they generated. The conclusions follow in Section 8.3 and present a meta-view of our work, underlining how the directive guidelines manifested in our work, the different transverse links that appear between the different parts and their conceptual articulation. Finally, Section 8.4 closes the chapter by detailing the ongoing work as well as the planned extensions of the research presented in this thesis.

8.1 Thesis outline

The purpose of this section is to present an overview of the work presented in this thesis and, most notably, to position our contributions relative to the current state of the art. It also follows the logical flow of ideas throughout our work and creates a summary of the tasks and accomplishments.

All the work presented in this thesis lies at the intersection of **Complex Data Analysis** and **Semi-Supervised Clustering**. More specifically, we address two research challenges: (i) embedding semantics into data representation and machine learning algorithms and (ii) leveraging the temporal dimension. We investigate on how data of different natures can be analyzed, while considering the temporal dimension and the additional information that may come with the data. Given the great heterogeneity of complex data (*e.g.*, different natures, temporal dimension) our work touches to many different aspects of Machine Learning. Our research ranges from introducing partial supervision into clustering, to using ontologies for topic evaluation and to extracting visual features from images. After a chapter in which we present an overview of the domain, we present our original research in four chapters, (a) one

for the temporal dimension, (b) a second one dedicated to reconstructing a feature set, with direct application in re-organizing user-supplied additional information under the form of labels, and the last two of them corresponding to the considered natures of complex data ((c) text and (d) images). A fifth chapter is dedicated to `CommentWatcher`, an academic software, result of the applied research and in close connection with the work on the textual dimension. Given the great diversity of the approached subjects, each chapter contains dedicated sections presenting the related work in the given field, the proposed contributions and partial conclusions. Therefore, each of the five chapters can be seen as autonomous, while remaining connected by the general research topics and the transverse links between them (some detailed in Section 8.3).

Chapter 1 starts by positioning our work and presenting the general context of the thesis. It is followed, in **Chapter 2**, by an overview of the two domains around which our work revolves: Complex Data Analysis and Semi-Supervised Clustering.

Chapter 3 addresses the task of *leveraging the temporal dimension of complex data* into clustering. Relative to existing methods, the work proposed in this chapter introduces (a) a new temporal-aware dissimilarity measure, which combines the descriptive dimension with the temporal dimension and allows the fine-tuning of their ratio; (b) a new penalty function, calculated using a function inspired from the Normal Distribution function, to ensure a contiguous segmentation of the observations belonging to an entity. Unlike existing solutions, which are basically a threshold function, our proposal inflict a high penalty for breaking the constraints for observations close in time and a low penalty for distant observations; (c) a novel time-driven constrained clustering algorithm, called TDCK-Means, which creates a partition of coherent clusters, both in the multidimensional space and in the temporal space; (d) a new measure (*ShaP*), to evaluate the contiguity of the segmentation of the series of observations belonging to an entity and (e) a new method to infer social roles in a social network as a mixture of temporal behavioral roles. As far as we know, this is the first proposal to infer the social roles as a succession of temporal states, previous solutions concentrate on calculating a set of measures on the network's graph.

Chapter 4 regroups our research concerning the task of *semantic data representation reconstruction*. In this chapter we address the problem of improving representation space of the data by using the underlying semantics of the dataset. Relative to existing methods, the work proposed in this chapter introduces (a) two algorithms that construct the new features as conjunctions of the initial features and their negations. The constructed features are more appropriate for describing the dataset and, at the same time, are comprehensible for a human user. The methods present so far in the literature either construct non-comprehensible features (*e.g.*, PCA, the kernel of SVM) or construct comprehensible features in a supervised way. As far as we know, this is the first solution for an unsupervised construction of comprehensible features. We also propose (b) a measure to quantify the total co-occurrence of a feature set; (c) a method, based on statistical testing, for setting the value of parameters; (d) a method, based on statistical considerations, for pruning the candidate pairs of correlated features.

Chapter 5 presents our research concerning image data, and more specifically the task of *improving image representation using semi-supervised visual vocabulary construction*. We are interested in using expert knowledge, under the form of non-positional labels attached to the images, in the process of creating the image numerical representation. Relative to

existing methods, the work proposed in this chapter introduces (a) a method for constructing a dedicated visual vocabulary starting from features sampled only from a subset of labeled images. This ensures that the generated visual vocabulary has words adapted to describing each of the objects appearing in the image collection; (b) a novel method for filtering features irrelevant to a given object, using non-positional labels. We show that that filtering approach consistently improves the accuracy of a content-based image classification task.

Chapter 6 presents in detail our research concerning textual data, and more precisely, we interest in the task of *topic extraction and evaluation*. We make an in-depth review of topic extraction and evaluation literature, while referencing methods related to our general domain of interest (*e.g.*, incorporating the temporal dimension or external semantic knowledge). Relative to existing methods, the work proposed in this chapter introduces (a) a textual clustering-based topic extraction system; (b) a topic evaluation system. Most of the solutions present in the literature use statistical measures (*e.g.*, the perplexity) to asses the fitness of topics. Our solution uses an external semantic resource, such as WordNet, to evaluate the semantic cohesion of topics.

Chapter 7 presents the practical prototype production, most notably `CommentWatcher`, an open source tool for analyzing discussions on web forums. Most of the solutions present in literature have the inconvenience of being either (a) proprietary (*i.e.*, cannot be used in scientific purposes) or (b) unable to crawl web sources and can be only used to analyze formatted datasets provided by the user. `CommentWatcher` has the advantage of being constructed as a web platform, giving the possibility of being used collaboratively. It also allows to by-pass the problem of copyright issues: the platform is open-source and the forum benchmark needs not to the distributed (each researcher can locally recreate the benchmark). `CommentWatcher` features (i) automatic fetching of forums, using a versatile parser architecture, (ii) topic extraction from a selection of texts and (iii) a temporal visualization of extracted topics and the underlying social network of users.

8.2 Original contributions

We summarize our work by presenting three of the most important ideas of our research. Most of the original proposals presented in this thesis are related to these ideas:

- *taking into account both the temporal dimension and the descriptive dimension into a clustering framework* (in Chapter 3). The resulted clusters are coherent from both the temporal and the descriptive point of view. Constraints are added to ensure the entity segmentation contiguity. This idea is presented in a paper in the proceedings of the **International Conference on Tools with Artificial Intelligence (ICTAI '12)**, paper which won the **best student paper award** [Rizoiu *et al.* 2012];
- *unsupervised construction of a feature set based on the co-occurrences issued from the dataset* (in Chapter 4). This allows adapting a feature set to the dataset's semantics. The new features are constructed as conjunctions of the initial features and their negations, which renders the result comprehensible for the human reader. This idea was published in an article with the international **Journal of Intelligent Information Systems (JIIS)** [Rizoiu *et al.* 2013a];
- *using non-positional user labels (denoting objects) to filter irrelevant visual features*

and to construct a semantically aware visual vocabulary for a "bag-of-feature" image representation (in Chapter 5). Even if the position of objects is unknown, we use the information about the presence of objects in images to detect and remove features unlikely to belong to the given object. Dedicated visual vocabularies are constructed, resulting in a numerical description which yields higher object categorization accuracy. This idea is presented in an article under review with the **International Journal of Artificial Intelligence Tools (IJAIT)** [Rizoiu *et al.* 2013b].

Throughout this manuscript, multiple contributions were proposed, some of which are issued from the major ideas presented before. Others are related to the textual nature of complex data and the prototype production, and were published in the proceedings of the **International Joint Conference on Artificial Intelligence (IJCAI '11)** [Musat *et al.* 2011b], the **International Symposium on Methodologies for Intelligent Systems (ISMIS '11)** [Musat *et al.* 2011a], the French national conference **Extraction et Gestion des Connaissances (EGC '10)** [Rizoiu *et al.* 2010] and a chapter in the book **Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances** [Rizoiu & Velcin 2011]. A complete list of the publications issued from the research presented in this thesis can be found in Annex B.

8.3 General conclusions

There are two main research challenges that our work addresses: (i) embedding semantics into data representation and machine learning algorithms and (ii) leveraging the temporal dimension. The two challenges are linked to (a) the applications concerning the structuring of the immense quantities of complex data produced by the *Web 2.0* and (b) the embedding of semantics into webpages, which is the object of the *Semantic Web*. The general context my thesis lies at the intersection of **Complex Data Analysis** and **Semi-Supervised Clustering**. In our work concerning the semantic representation, we concentrate on leveraging additional external information, under the form of expert-provided information (*e.g.*, image labels in Chapter 5) and external semantic resources (*e.g.*, concept ontologies in Chapter 6), while our work concerning the temporal dimension of data is presented in Chapter 3.

As mentioned earlier, our work is composed of four distinct, yet complementary parts. The four parts deal, respectively, with (a) the temporal dimension, (b) semantic data representation and the different natures of complex data, *i.e.*, (c) image and (d) text. Each part is dealt with in an individual chapter, which contains an overview of the state of the art of the domain, the proposals, conclusions about the work and some plans for future work. A fifth chapter is dedicated to the my applied work. Therefore, each of the five chapters can be seen as autonomous, while remaining connected the **directive guidelines**, the **transverse links** between them and the **conceptual articulation**. We detail in the following paragraphs how each of the above were taken into account in our work.

Directive guidelines While the different aspects of our work are distinct, they are not independent one from the other. Each deals with the different particularities of complex data, while respecting the same directive guidelines: (i) human comprehension, (ii) translating data of different natures into a semantic-aware description space and (iii) devising

algorithms and methods that embed semantics and the temporal component.

We consider crucial to generate **human comprehensible outputs** and we have stressed this aspect all along this thesis. Our proposal for feature construction, in Chapter 4, was partly motivated by this need. Black-box approaches (*i.e.*, the feature extraction and the feature selection algorithms presented in Section 4.2, p. 63) exist and they can achieve well the reduction of co-occurrences in the feature set. But the new features are completely synthetic and make results difficult to interpret. Human comprehension is also central for topic labeling, presented in Chapter 6. We argue that a complete expression is more meaningful for a human being than a probability distribution over words. Similarly, when generating the typical evolution phases in Chapter 3, human comprehension motivates the choice to segment contiguously the observations corresponding to an entity.

As shown in Section 2.1 (p. 9), some of today's challenges concerning complex data lie in (a) rendering the data into a common usable numeric format, which succeeds in capturing the information present in the native format, and in (b) efficiently using external information for improving the numeric representation. The second directive guideline throughout our work is **translating data of different natures into a semantic-aware description space**, which we call the Numeric Vectorial Space. Our work concerning images in Chapter 5 is specifically targeted at embedding semantic information in the image numerical description. Section 6.2 (p. 128) in Chapter 6 is dedicated to translating textual data into the "bag-of-words" numerical representation. Finally, the purpose of the feature construction algorithm in Chapter 4 is to improve the representation of data, by adapting the features to the dataset they describe.

Finally, a central axis of our research is **devising algorithms and methods that embed semantics and the temporal component**, based on unsupervised and semi-supervised techniques. Often additional information and knowledge is attached to the data, under the form of (a) user labels, (b) structure of interconnected documents or (c) external knowledge bases. We use this additional knowledge at multiple instances, usually using semi-supervised clustering techniques. We use such an approach when constructing a semantically improved image numeric representation, in Chapter 5. Dedicated visual vocabularies are constructed starting from the provided labeled subset of images. But the entire image collection (labeled and unlabeled) is used to generate the actual image representation. Furthermore, our work with the text also deals with leveraging external semantic knowledge into the topic evaluation process. Similarly, in Chapter 3, we use semi-supervised soft pair-wise constraints to model the temporal dependencies in the data.

Transverse links There are multiple **transverse links** between the different parts of our work. (a) The feature construction algorithm, **uFC**, was initially motivated by the need to re-organize the user user label set we use to create the semantic-enabled image representation. In Section 5.5 (p. 121), we discuss how the proposed semantic visual vocabulary construction can be adapted to scene classification (where the main problem is label co-occurrence) by using the proposed feature construction algorithm. (b) Our work with textual data is intimately linked with the software `CommentWatcher`. The text from online discussion forums is retrieved, we extract topics from it and infer a social network using the forum's reply-to relation. The social network is modeled and visualized as a multidigraph, in which links between nodes are associated to topics. (c) Furthermore, the temporal-driven

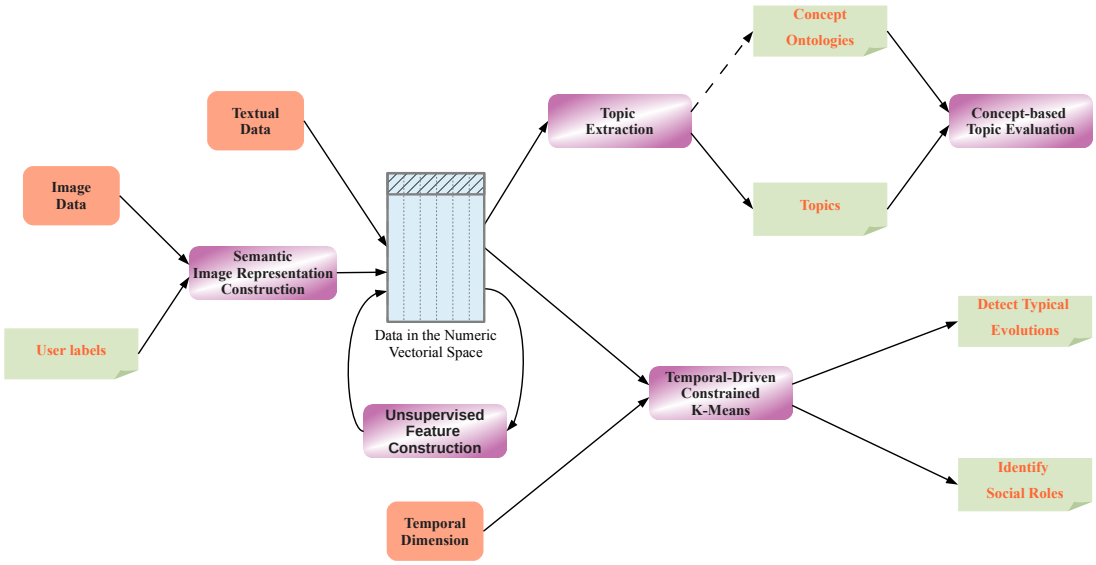


Figure 8.1 – Conceptual integration of the work in the thesis.

clustering algorithm TDCK-Means is applied to detect social roles in the social network. Behavioral roles are first identified, as shown in Section 3.6 (p. 52), and social roles are inferred as a succession of behavioral roles. (d) Finally, we have ongoing work which deals with embedding the temporal dimension into the feature construction algorithm. The idea is to detect if features are correlated with a certain time lag. We give more details about this current work in Section 8.4.1.

Conceptual articulation of the different parts The semantic-enabled numeric representation space is also the conceptual joining point of the different parts of our research, as shown in Figure 8.1. Given the extent of the approached subjects, the schema is not a blueprint of an integrated system. It rather has the vocation of giving the reader an overview of the articulation between the different parts of our work: data of different natures is translated into a semantic-aware numeric format, which is afterwards further used together with the temporal dimension or with external knowledge bases. This space is not common for all types of considered data and, therefore, we treat, in our work, each data type individually. One of the long term future plans of our work is a broader integration of all the information provided by complex data (*e.g.*, dealing simultaneously with data of different natures, the temporal component and expert knowledge).

The schema presented in Figure 8.1 was incrementally constructed in Figures 3.1 (p. 30), 4.17 (p. 93), 6.13 (p. 165), and 5.13 (p. 122). At the end of each chapter, the reader was shown how the work presented in the given chapter can be conceptually integrated with the work in previous chapters.

8.4 Current and Future work

The domain of this thesis is vast and plenty of work still remains. Future work for each of the parts has already been planned in the corresponding chapters. The remainder of this section will revisit some of them and present them in a broader context. We divide our research plans into current work and future work, depending on their status of ongoing or planned work. The order of presentation roughly represents the advancement of each of the ideas, mostly for current work.

8.4.1 Current work

Detecting behavioral roles At present, we are simultaneously working on multiple extensions and applications of our work. One of the directions of our current work is to apply TDCK-Means to the detection of user social roles in social networks. This work is performed in collaboration with the Technicolor Laboratories in Rennes, France, and it is in an advanced state: the submission of an article is already planned. The reader has already seen a more detailed description of this current work in Section 3.6 (p. 52).

Our underlying hypothesis is that, when interacting in an online community, a user plays, during a given period of time, multiple roles. We assume that these roles are temporally coherent (*i.e.*, a user's activity is uniformly similar when in a role) and he/she can change between roles. We call these roles *behavioral roles* and we construct a global *social role* as a mixture of different behavioral roles, which incorporates the dynamics of behavioral transitions. Therefore, we define the user social role as a succession of behavioral roles.

The social roles of users are constructed based on the social network inferred from online discussion forums. The used `TWOP` [Anokhin *et al.* 2012] dataset is constructed based on *Television Without Pity*[1] forum website. We have briefly shown, in Chapter 7, how such a social network can be constructed (when we presented `CommentWatcher`, our tool designed to analyze online discussion forums). The nodes of the graph are the users posting in the forums. A directed arc is added between users A and B when A replies to B. *Social roles* are identified in a three-phase framework: (a) behavioral features are identified based on the structure of the inferred social network, (b) *behavioral roles* are created using TDCK-Means and (c) the user *social roles* are determined based in the transitions between behavioral roles. Section 3.6 (p. 52) describes more in detail this process, alongside with some preliminary results.

Inferring a graph structure for temporal clusters In Chapter 3, when we constructed our temporal clustering algorithm, TDCK-Means, we concentrated on the temporal and descriptive coherence of clusters, as well as on the contiguous segmentation of observations belonging to an entity. We are currently working on an extension of TDCK-Means, which also organizes the resulted temporal clusters in a graph structure. This would be very useful for the human comprehension of the constructed evolution phases (see the discussion on the human comprehensibility guideline of our work, in Section 8.3). The trajectory of an individual through an evolution graph is easier to follow and more informative towards the relations between phases.

1. http://www.televisionwithoutpity.com/

We are working on (i) a temporal distance between clusters and (ii) a function that quantifies the intersection of two clusters. We plan to extend the Objective Function defined in Equation 3.7 (p. 39) with these two functions, which quantify the relation between two clusters. With these modifications, the graph's adjacency matrix could be inferred simultaneously with the temporal clusters.

Temporal feature construction Another research direction that we already started undergoing is incorporating temporal information into the feature construction algorithm, presented in Chapter 4. From a research challenge point of view, this work will allow the integration of our two research challenges, with which, for now, we deal individually.

This work is motivated by the fact that *the introduction of the temporal information changes the problem definition.* The datasets used in Chapter 4 have no temporal evolution. The building block of the **uFC** algorithm is feature co-occurrence. We have motivated that this co-occurrence is not the fruit of hazard, but has a semantic meaning. For example, *"manifestation"* co-occurs with *"urban"* because usually manifestations take place in cities. With the introduction of the temporal information, new questions arise and new semantic information can be induced. Such a question would be *what means co-occurrence in a temporal context?* Some features might co-occur, but not simultaneously. For example, the arrival to power of a socialist government and the increase of the country's public deficit might be correlated, but with a time lag, as the macro-economic indicators have a big inertia. The purpose of this work is to detect such "correlations at a time lag" and create new features like *"socialist"* and *"public_ deficit"* co-occur at a time lag δ.

We have extended the correlation coefficient defined in Equation 4.5 (p. 70) to calculate the correlation with a given fixed lag δ. The experiments we performed so far show that an "optimum" lag δ can be determined, that maximizes the temporal correlation. We are currently working on an extension of the *and* operator to the temporal case. The new features are no longer constructed as boolean expressions, but as temporal chains like $f_i \xrightarrow{\delta_1} f_j \xrightarrow{\delta_2} f_k$, meaning that f_i precedes f_j at a time distance δ_1, which f_j precedes f_k at a time distance δ_2.

This approach allows us to improve our feature construction algorithm, so that it uses the temporal dimension in addition to data semantics to improve data representation. In addition, the newly constructed temporal features can be used as easily comprehensible labels for the temporal clusters extracted using TDCK-Means.

8.4.2 Future work

One of our short term plans is devising a method for **automatically setting the values of TDCK-Means's parameters** (α, β and δ), by using an approach inspired from multi-objective optimization using evolutionary algorithms [Zhang & Li 2007]. The idea is to transform the learning process into a multi-objective learning problem. Each of the additive components of the objective function defined in Equation 3.7 (p. 39) becomes an objective (in the sense of multi-objective optimization). Multiple instances of TDCK-Means, with multiple initial combinations of values for α, β and δ, will be launched simultaneously and a small seed set of solutions will be obtained. Using evolutionary algorithms, the seed set can be evolved to approximate the Pareto front (in the space defined by the different objectives).

In the end, it suffices to chose, on the generated Pareto front, a compromise solution, for example using a technique similar to the one employed in Section 4.5.2 (p. 74). Such a technique would eliminate the need to arbitrarily set the parameters and would guarantee an "optimum" (meaning Pareto non-dominated), given certain criteria.

Another planned extension venue is adapting the **image representation construction proposal to incomplete labeling**. This boils down to adapting the feature construction algorithm to data issued from the internet (*e.g.*, labels on an image sharing platform). We have made several assumptions about the feature set (in Chapter 4) and the label set (in Chapter 5). For comprehensibility reasons, in the remainder of the discussion we consider the two as being the same problem in two different contexts and we denote them as labels. Throughout Chapters 4 and 5, we considered the labeling to be complete: if the label was not present, it therefore means that the object is absent. This assumption supposes binary labeling, where **true** means presence and **false** means absence. In the case of real world labeling, the absence of a label might also mean that the user forgot/chose not to label the given image/document. Therefore, a value of **false** is no longer a sure indicator for the absence of the given object. For example, when a user is labeling an image depicting a cascade and has a choice between *water*, *cascade* or both, he/she might choose only *cascade* as it is the most specific. This adds new challenges for both (i) the feature construction algorithm (*i.e.*, the co-occurrence of *water* and *cascade* is no longer present) and (ii) the filtering algorithm for image representation construction (*i.e.*, a keypoint belonging to a *cascade* is not different from a keypoint belonging to *water*).

Our short term plans for the applied part of our work are closely related to the textual data. We intend to **implement the proposed topic evaluation using a concept hierarchy in `CommentWatcher`**. This, alongside integrating other topic extraction algorithms, will allow performing a thorough comparative evaluation of topic models in the context of internet issued texts. Most of the extensions for `CommentWatcher` concern the visualization module, and more specifically the generation of the social network based on the discussion forums. Other, longer term plans include extending (a) the natures of complex data that can be processed and (b) the knowledge that can be used as additional information (*e.g.* processing video, using knowledge from the semantic web *etc.*). In a foreseeable future, the treatment of structured data of multiple natures will be integrated into `CommentWatcher` and it will become a veritable "media mining" platform, capable to retrieve and analyze text and images from discussion forums, online news media and other imaginable online sources.

The long term goal, for both the theoretical and the applied parts of our work, is a broader integration of all the information provided by complex data: text, image, video, audio, numerical measurements, temporal dimension, user labels and ontologies. Throughout this manuscript, we have shown our reader how to solve a number of different tasks (*e.g.*, how to take time into account, how to evaluate topics using external resources), but the real objective is devising algorithms and data representation that are capable of taking profit from every available piece of information and construct a complete knowledge inference system.

Appendix A

Participation in Research Projects

A.1 Participation in projects

Some of the learning tasks presented in Chapter 1 were partially motivated by the specific problems and applications needed by the different research projects in which I was involved. I present, in Table A.1, the list with these projects alongside with my contributions to them.

A.2 The ImagiWeb project

On a daily basis, millions of people post their opinions on Web 2.0 and discuss about various topics such as the news, politics, the latest results of athletics, *etc.* These kinds of postings contribute to the production and dissemination of the image of different entities, such as that of politicians or companies. The image, as we specify it here, is a structured and dynamic representation which can be seen in at least two ways: the representation that an entity wishes to assign to itself, and the view that a person or a group of persons has of this entity. Thus, Internet seems to be privileged in its role as a contributor to disseminate, strengthen and impose representations and opinions, and as a place where the logic of influence is present.

In this framework, the ImagiWeb project aims precisely at studying the image of entities of various kinds (companies, politicians, *etc.*) as this is diffused and viewed on the Internet. The study of these representations and their dynamics is considered today to be a real challenge which, if it is resolved (even partially), will not only allow to respond to specific needs, especially in the field of the press watching, but it will also answer to important nowadays issues in the field of political sciences and sociology in general.

The project proposes two major novelties. The first is to address together a set of issues treated so far separately (*i.e.*, study of the opinions, taking context into account, the topic evolution, social network analysis, study of the topology of the Web) around a common object that is the image of entities (in the sense of the representation) that populate the Web. Emphasis is given to the fact that one entity can be associated with several different images, and also to the underlying temporal aspect of the dynamics of the images.

The second novelty concerns the involvement in the project of Social Sciences and Humanities researchers. This is still quite rare in such computer science projects. Thus, the

1. https://research.technicolor.com/rennes/
2. http://eric.univ-lyon2.fr/~jvelcin/imagiweb/
3. http://recherche.univ-lyon2.fr/crtt/
4. http://www.elico-recherche.eu/?lang=en
5. http://www.conversationnel.fr/
6. Download here: http://eric.univ-lyon2.fr/~arizoiu/files/discussion-analysis.jar

Table A.1 – List of research projects in which I was involved during my PhD thesis.

Period	Name and short description
2012 - present:	**Research project with Technicolor R&I Labs [1], Rennes** Apply and develop the research in the domains of temporal clustering to social role identification in social networks. **Contribution:** Research concerning the learning task of *social role identification*, detailed in Section 3.6 (p. 52).
2012 - present:	**The ImagiWeb [2] project** (financed by the **ANR**) Analyze the evolution of the image of politicians and enterprises through social media and Twitter. ImagiWeb is further detailed in Section A.2. **Contribution:** Research concerning the task of *detecting typical evolutions* and *temporal data clustering*, applied to a political science dataset. More details in Chapter 3.
2012 - present:	**The Crtt [3]-Eric project** (financed by the **Lyon 2 University**) Study the evolution of specialized discourse in the domain of nuclear medicine, while taking into account the temporal evolution and the different involved populations (*e.g.*, doctors, nurses, patients). **Contribution:** Research concerning the tasks of *topic extraction, labeling and evaluation* (details in Chapter 6). Applied work and student internship supervision towards developing `CommentWatcher`.
2010 - 2011:	**The Eric-Elico [4] project** (financed by the **Lyon 2 University**) Analyze the information extracted, either by usage of machine learning algorithms or manually, by the experts in Communication Sciences. **Contribution:** Research concerning the tasks of *topic extraction, labeling and evaluation* and *content-based image classification*. Applied work towards developing `CommentWatcher`.
2009:	**The Conversession project** (financed by the **Rhône-Alpes region**) Develop a new platform for organizing and analyzing online forum discussions. The project was associated with the creation of a start-up enterprise [5]. **Contribution:** Research concerning the tasks of *topic extraction, labeling and evaluation* and applied work: I have developed the prototype [6], dealing with the parsing web forums, topic extraction and topic labeling.

case study about the EDF (French Electricity Company) company will be carried out with the assistance of a semiologist who will be able to enlighten the automated analysis provided by computer tools produced during the project. In addition, the involvement of the social scientists of CEPEL will allow not only to conduct a relevant study on the image of the politicians, but also to provide answers relative to the issues of representing the data extracted from the Web and characterizing the panels of Internet users.

The project ImagiWeb involves six partners: (a) three academics: the ERIC Laboratory [7], the CEPEL laboratory [8] and the LIA laboratory [9] and (b) three companies: AMI

7. http://eric.univ-lyon2.fr/
8. http://www.cepel.univ-montp1.fr/
9. http://lia.univ-avignon.fr/

Software [10], EDF [11] and XEROX [12]. The ImagiWeb project is financed by the French National Research Agency (ANR).

10. http://www.amisw.com/en/
11. http://innovation.edf.com/innovation-et-recherche-20.html
12. http://www.xrce.xerox.com/

Appendix B

List of Publications

International journals

- Marian-Andrei Rizoiu, Julien Velcin and Stéphane Lallich. *Unsupervised Feature Construction for Improving Data Representation and Semantics.* Journal of Intelligent Information Systems, vol. 40, no. 3, pages 501–527, 2013.

Proceedings of international conferences

- Marian-Andrei Rizoiu. *Semi-Supervised Structuring of Complex Data.* In Doctoral Consortium of International Joint Conference on Artificial Intelligence, Proceedings of the Twenty-Third, IJCAI 2013. AAAI Press, November 2013.
- Marian-Andrei Rizoiu, Julien Velcin and Stéphane Lallich. *Structuring typical evolutions using Temporal-Driven Constrained Clustering.* In International Conference on Tools with Artificial Intelligence, Proceedings of the Twenty-Forth, ICTAI 2012, pages 610–617. IEEE, November 2012. **Best Student Paper Award.**
- Claudiu Musat, Julien Velcin, Stefan Trausan-Matu and Marian-Andrei Rizoiu. *Improving topic evaluation using conceptual knowledge.* In International Joint Conference on Artificial Intelligence, Proceedings of the Twenty-Second, volume 3 of *IJCAI 2011*, pages 1866–1871. AAAI Press, 2011.
- Claudiu Musat, Julien Velcin, Marian-Andrei Rizoiu and Stefan Trausan-Matu. *Concept-based Topic Model Improvement.* In International Symposium on Methodologies for Intelligent Systems, volume 369 of *ISMIS 2011*, pages 133–142. Springer, June 2011.

National journals and proceedings of national conferences

- Marian-Andrei Rizoiu, Julien Velcin and Jean-Hugues Chauchat. *Regrouper les données textuelles et nommer les groupes à l'aide des classes recouvrantes.* In Extraction et Gestion des Connaissances, (EGC 10) 10ème Conférence, volume E-19 of *Revue des Nouvelles Technologies de l'Information*, pages 561–572. Cépaduès, January 2010.
- Claudiu Musat, Marian-Andrei Rizoiu and Stefan Trausan-Matu. *An Intra and Inter-Topic Evaluation and Cleansing Method.* Romanian Journal of Human-Computer Interaction, vol. 3, no. 2, pages 81–96, 2010.

Book chapters

– Marian-Andrei Rizoiu and Julien Velcin. *Topic Extraction for Ontology Learning.* In Wilson Wong, Wei Liu and Mohammed Bennamoun, editors, Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances, chapter 3, pages 38–61. Hershey, PA: Information Science Reference, 2011.

Under review and submitted

– Marian-Andrei Rizoiu, Julien Velcin and Stéphane Lallich. *Visual Vocabulary Construction for Image Classification in a Weakly Supervised Context.* International Journal of Artificial Intelligence Tools, 2012. **Under review.**
– Marian-Andrei Rizoiu, Julien Velcin and Stéphane Lallich. *How to use Temporal-Driven Constrained Clustering to detect typical evolutions.* International Journal of Artificial Intelligence Tools, 2013. **Submitted.**

Bibliography

[Abouelhoda *et al.* 2002] Mohamed Abouelhoda, Enno Ohlebusch and Stefan Kurtz. *Optimal exact string matching based on suffix arrays.* In String Processing and Information Retrieval, pages 175–180. Springer, 2002. (Cited on page 143.)

[Aggarwal *et al.* 2003] Charu C. Aggarwal, Jiawei Han, Jianyong Wang and Philip S. Yu. *A Framework for Clustering Evolving Data Streams.* In Very large data bases, Proceedings of the 29th International Conference On, pages 81–92, 2003. (Cited on page 18.)

[Agrawal & Srikant 1995] Rakesh Agrawal and Ramakrishnan Srikant. *Mining sequential patterns.* In Data Engineering, 1995. Proceedings of the Eleventh International Conference on, pages 3–14. IEEE, 1995. (Cited on page 17.)

[Agrawal *et al.* 1993] Rakesh Agrawal, Tomasz Imieliński and Arun Swami. *Mining association rules between sets of items in large databases.* In ACM SIGMOD Record, volume 22, pages 207–216. ACM, 1993. (Cited on page 17.)

[Agresti 2002] Alan Agresti. Categorical data analysis, volume 359. Wiley-interscience, 2002. (Cited on page 63.)

[Amer-Yahia *et al.* 2012] Sihem Amer-Yahia, Samreen Anjum, Amira Ghenai, Aysha Siddique, Sofiane Abbar, Sam Madden, Adam Marcus and Mohammed El-Haddad. *MAQSA: a system for social analytics on news.* In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12, pages 653–656, New York, NY, USA, 2012. ACM. (Cited on pages 169 and 170.)

[Anaya-Sánchez *et al.* 2008] Henry Anaya-Sánchez, Aurora Pons-Porrata and Rafael Berlanga-Llavori. *A New Document Clustering Algorithm for Topic Discovering and Labeling.* In Iberoamerican congress on Pattern Recognition, Proceedings of the 13th, CIARP '08, pages 161–168, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 138.)

[Anokhin *et al.* 2012] Nikolay Anokhin, James Lanagan and Julien Velcin. *Social Citation: Finding Roles in Social Networks. An Analysis of TV-Series Web Forums.* In Mining Communities and People Recommenders, The Second International Workshop on, pages 49–56, 2012. (Cited on pages 52, 53, 169 and 183.)

[Armingeon *et al.* 2011] Klaus Armingeon, David Weisstanner, Sarah Engler, Panajotis Potolidis, Marlène Gerber and Philipp Leimgruber. *Comparative Political Data Set 1960-2009.* Institute of Political Science, University of Berne., 2011. (Cited on pages 31, 32 and 43.)

[Athanasiadis *et al.* 2005] Thanos Athanasiadis, Vassilis Tzouvaras, Kosmas Petridis, Frederic Precioso, Yannis Avrithis and Yiannis Kompatsiaris. *Using a multimedia ontology infrastructure for semantic annotation of multimedia content.* In International Workshop on Knowledge Markup and Semantic Annotation, collocated with International Semantic Web Conference (ISWC 2005), SemAnnot '05, Galway, Ireland, November 2005. (Cited on page 104.)

[Bar-Hillel *et al.* 2003] Aharon Bar-Hillel, Tomer Hertz, Noam Shental and Daphna Weinshall. *Learning distance functions using equivalence relations.* In Machine Learning, International Workshop then conference, volume 20, page 11, 2003. (Cited on page 25.)

[Barnard *et al.* 2003] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando De Freitas, David M. Blei and Michael I. Jordan. *Matching words and pictures.* The Journal of Machine Learning Research, vol. 3, pages 1107–1135, 2003. (Cited on page 16.)

[Basu *et al.* 2002] Sugato Basu, Arindam Banerjee and Raymond J. Mooney. *Semi-supervised clustering by seeding.* In International Conference on Machine Learning, pages 19–26, 2002. (Cited on page 27.)

[Basu *et al.* 2003] Sugato Basu, Mikhail Bilenko and Raymond J. Mooney. *Comparing and Unifying Search-Based and Similarity-Based Approaches to Semi-Supervised Clustering.* In Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, Collocated with ICML '03, ICML '03, pages 42–49, 2003. (Cited on pages 23, 24, 26 and 27.)

[Bay *et al.* 2006] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. *Surf: Speeded up robust features.* Computer Vision–ECCV 2006, pages 404–417, 2006. (Cited on pages 100 and 103.)

[Bekkerman & Jeon 2007] Ron Bekkerman and Jiwoon Jeon. *Multi-modal clustering for multimedia collections.* In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007. (Cited on pages 14 and 15.)

[Bellman & Kalaba 1959] Richard Bellman and Robert Kalaba. *On adaptive control processes.* Automatic Control, IRE Transactions on, vol. 4, no. 2, pages 1–9, 1959. (Cited on page 19.)

[Benjamini & Liu 1999] Yoav Benjamini and Wei Liu. *A step-down multiple hypotheses testing procedure that controls the false discovery rate under independence.* Journal of Statistical Planning and Inference, vol. 82, no. 1-2, pages 163–170, 1999. (Cited on page 83.)

[Bernstein *et al.* 2010] Michael S. Bernstein, Bongwon Suh, Lichan Hong, Jilin Chen, Sanjay Kairam and Ed H. Chi. *Eddi: interactive topic-based browsing of social status streams.* In UIST '10, pages 303–312, 2010. (Cited on page 170.)

[Berry *et al.* 1995] Michael W. Berry, Susan T. Dumais and Gavin W. O'Brien. *Using Linear Algebra for Intelligent Information Retrieval.* SIAM Review, vol. 37, no. 4, pages 573–595, 1995. (Cited on page 133.)

[Beyer *et al.* 1999] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan and Uri Shaft. *When is "nearest neighbor" meaningful?* Database Theory, Proceedings of the Internatioanal Conference on, vol. ICDT '99, pages 217–235, 1999. (Cited on page 19.)

[Bilenko & Mooney 2003] Mikhail Bilenko and Raymond J. Mooney. *Adaptive duplicate detection using learnable string similarity measures.* In Knowledge Discovery and Data Mining, Proceedings of the ninth ACM SIGKDD international conference on, pages 39–48. ACM New York, NY, USA, 2003. (Cited on page 25.)

[Biskri *et al.* 2004] Ismaïl Biskri, Jean-Guy Meunier and Sylvain Joyal. *L'extraction des termes complexes : une approche modulaire semiautomatique.* In Gérard Purnelle, Cédrick Fairon and Anne Dister, editors, Analyse Statistique des Données Textuelles, Actes des 7èmes Journées Internationales de, volume 1, pages 192–201. Presses Universitaires de Louvain, 2004. (Cited on pages 137 and 138.)

[Bizer *et al.* 2009] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören. Auer, Christian Becker, Richard Cyganiak and Sebastian Hellmann. *DBpedia-A crystallization point for the Web of Data.* Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, no. 3, pages 154–165, 2009. (Cited on pages 2 and 10.)

[Blei & Lafferty 2006a] David M. Blei and John D. Lafferty. *Correlated Topic Models.* In Advances in Neural Information Processing Systems, Proceedings of the 2005 Conference on, volume 18, page 147. MIT, 2006. (Cited on pages 136 and 162.)

[Blei & Lafferty 2006b] David M. Blei and John D. Lafferty. *Dynamic topic models.* In International Conference on Machine Learning, Proceedings of the 23rd, pages 113–120. ACM, 2006. (Cited on page 136.)

[Blei & McAuliffe 2008] David M. Blei and Jon D. McAuliffe. *Supervised topic models.* Advances in Neural Information Processing Systems, vol. 20, pages 121–128, 2008. (Cited on page 140.)

[Blei *et al.* 2003] David M. Blei, Andrew Y. Ng and Michael I. Jordan. *Latent dirichlet allocation.* The Journal of Machine Learning Research, vol. 3, pages 993–1022, 2003. (Cited on pages 104, 131, 134 and 157.)

[Blei *et al.* 2004] David M. Blei, Thomas L. Griffiths, Michael I. Jordan and Joshua B. Tenenbaum. *Hierarchical topic models and the nested Chinese restaurant process.* In Advances in Neural Information Processing Systems, Proceedings of the 18th Annual Conference on, volume 16, pages 106–123. MIT Press, 2004. (Cited on pages 136 and 162.)

[Blockeel *et al.* 1998] Hendrik Blockeel, Luc De Raedt and Jan Ramon. *Top-down induction of clustering trees.* In Proceedings of the 15th International Conference on Machine Learning, pages 55–63, 1998. (Cited on page 66.)

[Bloedorn & Michalski 1998] Eric Bloedorn and Ryszard S. Michalski. *Data-driven constructive induction.* Intelligent Systems and their Applications, vol. 13, no. 2, pages 30–37, 1998. (Cited on page 64.)

[Blum & Mitchell 1998] Avrim Blum and Tom Mitchell. *Combining labeled and unlabeled data with co-training.* In Computational Learning Theory, Proceedings of the eleventh annual conference on, COLT ́98, pages 92–100. ACM, 1998. (Cited on page 104.)

[Bourigault 1992] Didier Bourigault. *Surface grammatical analysis for the extraction of terminological noun phrases.* In Computational Linguistics, Proceedings of the International Conference on, volume 92, pages 977–981, 1992. (Cited on page 137.)

[Boyd-Graber *et al.* 2007] Jordan Boyd-Graber, David M. Blei and Xiaojin Zhu. *A topic model for word sense disambiguation.* In Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,

Proceedings of the 2007, EMNLP-CoNLL 2007, pages 1024–1033, 2007. (Cited on page 140.)

[Brillinger 2001] David R. Brillinger. Time series: data analysis and theory, volume 36. Society for Industrial Mathematics, 2001. (Cited on page 17.)

[Buitelaar *et al.* 2005] Paul Buitelaar, Philipp Cimiano and Bernardo Magnini. *Ontology Learning from Texts: An Overview*. In Paul Buitelaar, Philipp Cimiano and Bernardo Magnini, editors, Ontology Learning from Text: Methods, Evaluation and Applications, volume 123 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2005. (Cited on pages 137, 161 and 162.)

[Cai *et al.* 2004] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma and Ji-Rong Wen. *Hierarchical clustering of WWW image search results using visual, textual and link information*. In International Conference on Multimedia, Proceedings of the 12th annual ACM, pages 952–959. ACM, 2004. (Cited on page 15.)

[Cattell 1966] Raymond B. Cattell. *The Scree Test For The Number Of Factors*. Multivariate Behavioral Research, vol. 1, no. 2, pages 245–276, 1966. (Cited on page 134.)

[Chang *et al.* 2009a] Jonathan Chang, Jonathan Boyd-Graber, Sean Gerrish, Chong Wang and David M. Blei. *Reading Tea Leaves: How Humans Interpret Topic Models*. In Advances in Neural Information Processing Systems, Proceedings of the 23rd Annual Conference on, volume 31 of *NIPS 2009*, 2009. (Cited on pages 140, 148, 149, 150 and 157.)

[Chang *et al.* 2009b] Jonathan Chang, Jordan Boyd-Graber and David M. Blei. *Connections between the lines: augmenting social networks with text*. In International Conference on Knowledge Discovery and Data Mining, Proceedings of the 15th ACM SIGKDD, pages 169–178. ACM, 2009. (Cited on page 136.)

[Chang *et al.* 2010] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard and Chih-Jen Lin. *Training and testing low-degree polynomial data mappings via linear SVM*. The Journal of Machine Learning Research, vol. 99, pages 1471–1490, 2010. (Cited on page 20.)

[Chapelle *et al.* 2006] Olivier Chapelle, Bernhard Schölkopf and Alexander Zien. Semi-supervised learning, volume 2 of *Adaptive Computation and Machine Learning*. The MIT Press, September 2006. (Cited on pages 2 and 21.)

[Chavez *et al.* 2008] Camara G. Chavez, Frederic Precioso, Matthieu Cord, Sylvie Phillip-Foliguet and A. de A. Araújo. *An interactive video content-based retrieval system*. In Systems, Signals and Image Processing, 15th International Conference on, IWSSIP '08, pages 133–136. IEEE, 2008. (Cited on page 100.)

[Chen *et al.* 2009] Shixi Chen, Haixun Wang and Shuigeng Zhou. *Concept clustering of evolving data*. In Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on, pages 1327–1330. IEEE, 2009. (Cited on page 34.)

[Christoudias *et al.* 2006] C. Mario Christoudias, Kate Saenko, Louis-Philippe Morency and Trevor Darrell. *Co-adaptation of audio-visual speech and gesture classifiers*. In Proceedings of the 8th international conference on Multimodal interfaces, pages 84–91. ACM, 2006. (Cited on page 16.)

[Cimiano *et al.* 2006] Philipp Cimiano, Johanna Völker and Rudi Studer. *Ontologies on Demand? -A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text Information.* Information, Wissenschaft und Praxis, vol. 57, no. 6-7, pages 315–320, 2006. (Cited on pages 137, 161 and 162.)

[Clare & King 2001] Amanda Clare and Ross D. King. *Knowledge discovery in multi-label phenotype data.* In Principles of Data Mining and Knowledge Discovery, pages 42–53. Springer, 2001. (Cited on page 94.)

[Cleuziou *et al.* 2004] Guillaume Cleuziou, Lionel Martin and Christel Vrain. *PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classification and Textual Data.* In R. López de Mántaras and L. Saitta, editors, European Conference on Artificial Intelligence, Proceedings of the 16th, ECAI '04, pages 440–444, Valencia, Spain, August 2004. (Cited on page 133.)

[Cleuziou 2008] Guillaume Cleuziou. *An extended version of the k-means method for overlapping clustering.* In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pages 1–4. IEEE, 2008. (Cited on pages 131, 133, 139, 140, 142 and 143.)

[Cleuziou 2009] Guillaume Cleuziou. *OKMed et WOKM : deux variantes de OKM pour la classification recouvrante.* In Jean-Gabriel Ganascia and Pierre Gançarski, editors, Extraction et Gestion des Connaisances, volume RNTI-E-15 of *EGC 2009*, pages 31–42. Cépaduès-Éditions, January 2009. (Cited on page 133.)

[Cohn *et al.* 2003] David Cohn, Rich Caruana and Andrew McCallum. *Semi-supervised clustering with user feedback.* In Constrained Clustering: Advances in Algorithms, Theory, and Applications, volume 4, pages 17–32. Cornell University, 2003. (Cited on pages 22, 23 and 25.)

[Comaniciu & Meer 2002] Dorin Comaniciu and Peter Meer. *Mean shift: A robust approach toward feature space analysis.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 5, pages 603–619, 2002. (Cited on page 103.)

[Cortes & Vapnik 1995] Corrina Cortes and Vladimir Vapnik. *Support-vector networks.* Machine learning, vol. 20, no. 3, pages 273–297, 1995. (Cited on pages 20, 21, 62, 64 and 112.)

[Csurka *et al.* 2004] Gabriela Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski and Cédric Bray. *Visual categorization with bags of keypoints.* In Workshop on statistical learning in computer vision, ECCV, volume 1, pages 1–22, 2004. (Cited on pages 100 and 103.)

[da Silva *et al.* 1999] Joaquim da Silva, Gaël Dias, Sylvie Guilloré and José Pereira Lopes. *Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units.* Progress in Artificial Intelligence, page 849, 1999. (Cited on page 138.)

[Darrell & Pentland 1993] Trevor Darrell and Alexander Pentland. *Space-time gestures.* In Computer Vision and Pattern Recognition, Proceedings of the IEEE Computer Society Conference on, CVPR '93, pages 335–340. IEEE, 1993. (Cited on page 17.)

[Davidson & Basu 2007] Ian Davidson and Sugato Basu. *A survey of clustering with instance level constraints.* ACM Transactions on Knowledge Discovery from Data, pages 1–41, 2007. (Cited on pages 2, 22 and 26.)

[Davidson & Ravi 2005] Ian Davidson and SS Ravi. *Clustering with constraints: Feasibility issues and the fc-means algorithm.* In International Conference on Data Mining, Proceedings of the Fifth SIAM, volume 119, page 138. Society for Industrial Mathematics, 2005. (Cited on page 23.)

[de Andrade Bresolin *et al.* 2008] Adriano de Andrade Bresolin, Adrião Duarte Dória Neto and Pablo Javier Alsina. *Digit recognition using wavelet and SVM in Brazilian Portuguese.* In Acoustics, Speech and Signal Processing, IEEE International Conference on, ICASSP '08, pages 1545–1548, 2008. (Cited on page 13.)

[De la Torre & Agell 2007] Fernando De la Torre and Carlos Agell. *Multimodal Diaries.* In Multimedia and Expo, 2007 IEEE International Conference on, pages 839–842. IEEE, 2007. (Cited on page 34.)

[de Medeiros Martins *et al.* 2002] Allan de Medeiros Martins, Wedson Torres de Almeida Filho, Agostinho Medeiros Brito Júnior and Adrião Duarte Dória Neto. *A new method for multi-texture segmentation using neural networks.* In Neural Networks, Proceedings of the 2002 International Joint Conference on, volume 3 of *IJCNN'02*, pages 2064–2069. IEEE, 2002. (Cited on page 102.)

[Demiriz *et al.* 1999] Ayhan Demiriz, Kristin Bennett and Mark J. Embrechts. *Semi-Supervised Clustering Using Genetic Algorithms.* In Artificial Neural Networks in Engineering, pages 809–814. ASME Press, 1999. (Cited on page 26.)

[Dermouche *et al.* 2013] Mohamed Dermouche, Julien Velcin, Sabine Loudcher and Leila Khouas. *Une nouvelle mesure pour l'évaluation des méthodes d'extraction de thématiques: la Vraisemblance Généralisée.* In Extraction et la Gestion des Connaissances, La 13ème Conférence Francophone sur, EGC '13, pages 317–328. Revue des Nouvelles Technologies de l'Information, 2013. (Cited on page 132.)

[Dias *et al.* 2000] Gaël Dias, Sylvie Guilloré and José Pereira Lopes. *Extraction automatique d'associations textuelles à partir de corpora non traités.* In M. Rajman and J. C. Chapelier, editors, Statistical Analysis of Textual Data, Proceedings of 5th International Conference on the, volume 2 of *JADT 2000*, pages 213–221, Lausanne, March 2000. Ecole Polytechnique Fédérale de Lausanne. (Cited on page 138.)

[Dietterich & Michalski 1985] Thomas G. Dietterich and Ryszard S. Michalski. *Discovering patterns in sequences of events.* Artificial Intelligence, vol. 25, no. 2, pages 187–232, 1985. (Cited on page 17.)

[Diplaris *et al.* 2005] Sotiris Diplaris, Grigorios Tsoumakas, Pericles A. Mitkas and Ioannis Vlahavas. *Protein classification with multiple algorithms.* Advances in Informatics, pages 448–456, 2005. (Cited on page 94.)

[Dormont 1989] Brigitte Dormont. *Petite apologie des données de panel.* Économie & prévision, vol. 87, no. 1, pages 19–32, 1989. (Cited on page 43.)

[Dumais *et al.* 1998] Susan Dumais, John Platt, David Heckerman and Mehran Sahami. *Inductive learning algorithms and representations for text categorization.* In Con-

ference on Information and Knowledge Management, Proceedings of the Seventh International, CIKM '98, pages 148–155. ACM, 1998. (Cited on page 130.)

[Dunn 1973] Joseph C. Dunn. *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters.* Journal of Cybernetics, vol. 3, no. 3, pages 32–57, 1973. (Cited on page 133.)

[Dunteman 1989] G. H. Dunteman. Principal components analysis, volume 69. SAGE publications, Inc, 1989. (Cited on pages 20, 62, 64 and 160.)

[Egghe 2006] Leo Egghe. *Theory and practise of the g-index.* Scientometrics, vol. 69, no. 1, pages 131–152, 2006. (Cited on page 53.)

[Elomaa & Rousu 2004] Tapio Elomaa and Juho Rousu. *Efficient multisplitting revisited: Optima-preserving elimination of partition candidates.* Data Mining and Knowledge Discovery, vol. 8, no. 2, pages 97–126, 2004. (Cited on page 63.)

[Estruch *et al.* 2008] Vicent Estruch, J. H. Orallo and M. J. R. Quintana. *Bridging the gap between distance and generalisation: Symbolic learning in metric spaces.* PhD thesis, Universitat Politècnica de València, 2008. (Cited on page 163.)

[Farnstrom *et al.* 2000] Fredrik Farnstrom, James Lewis and Charles Elkan. *Scalability for clustering algorithms revisited.* ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pages 51–57, 2000. (Cited on page 18.)

[Fawcett 2006] Tom Fawcett. *An introduction to ROC analysis.* Pattern recognition letters, vol. 27, no. 8, pages 861–874, 2006. (Cited on page 118.)

[Fayyad & Irani 1993] Usama M. Fayyad and Keki B Irani. *Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning.* In Ruzena Bajcsy, editor, International Joint Conference on Uncertainty in AI, Proceedings of the, volume 2, pages 1022–1027. Morgan Kaufmann, 1993. (Cited on page 63.)

[Fei-Fei & Perona 2005] Li Fei-Fei and Pietro Perona. *A bayesian hierarchical model for learning natural scene categories.* In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, volume 2 of *CVPR 2005*, pages 524–531. IEEE, 2005. (Cited on pages 100 and 103.)

[Fei-Fei *et al.* 2007] Li Fei-Fei, Rob Fergus and Pietro Perona. *Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories.* Computer Vision and Image Understanding, vol. 106, no. 1, pages 59–70, 2007. (Cited on page 113.)

[Feller 1950] W. Feller. An introduction to probability theory and its applications. vol. i. Wiley, 1950. (Cited on page 71.)

[Ferret 2006] Olivier Ferret. *Approches endogène et exogène pour améliorer la segmentation thématique de documents.* Traitement Automatique des Langues, vol. 47, no. 2, pages 111–135, 2006. (Cited on page 132.)

[Fisher 1987] Douglas H. Fisher. *Knowledge Acquisition Via Incremental Conceptual Clustering.* Machine Learning, vol. 2, no. 2, pages 139–172, 1987. (Cited on page 161.)

[Forestier *et al.* 2011] Mathilde Forestier, Julien Velcin and Djamel A. Zighed. *Extracting social networks to understand interaction.* In Advances in Social Networks Analysis

and Mining, International Conference on, ASONAM '11, pages 213–219. IEEE, 2011. (Cited on pages 169 and 174.)

[Fournier-Viger *et al.* 2011] Philippe Fournier-Viger, Roger Nkambou and Vincent Shin-Mu Tseng. *RuleGrowth: mining sequential rules common to several sequences by pattern-growth.* In Proceedings of the 2011 ACM Symposium on Applied Computing, pages 956–961. ACM, 2011. (Cited on page 17.)

[Francois *et al.* 2007] Damien Francois, Vincent Wertz and Michel Verleysen. *The concentration of fractional distances.* Knowledge and Data Engineering, IEEE Transactions on, vol. 19, no. 7, pages 873–886, 2007. (Cited on page 19.)

[Freund & Schapire 1997] Yoav Freund and Robert E. Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of computer and system sciences, vol. 55, no. 1, pages 119–139, 1997. (Cited on page 105.)

[Frintrop *et al.* 2010] Simone Frintrop, Erich Rome and Henrik I. Christensen. *Computational visual attention systems and their cognitive foundations: A survey.* ACM Transactions on Applied Perception (TAP), vol. 7, no. 1, page 6, 2010. (Cited on page 102.)

[Fulkerson *et al.* 2008] Brian Fulkerson, Andrea Vedaldi and Stefano Soatto. *Localizing objects with smart dictionaries.* Computer Vision–ECCV 2008, pages 179–192, 2008. (Cited on page 105.)

[Gangemi *et al.* 2003] Aldo Gangemi, Roberto Navigli and Paola Velardi. *The OntoWordNet Project: extension and axiomatization of conceptual relations in WordNet.* In International Conference on Ontologies, Databases and Applications of SEmantics, Proceedings of the, ODBASE'03, pages 820–838. Springer, 2003. (Cited on page 149.)

[Ganter & Wille 1999] Bernhard Ganter and Rudolf Wille. Formal concept analysis. Springer Berlin, 1999. (Cited on page 163.)

[Gao *et al.* 2006] Jing Gao, Pang-Ning Tan and Haibin Cheng. *Semi-supervised clustering with partial background information.* In In Proceedings of the Sixth SIAM International Conference on Data Mining, 2006. (Cited on page 26.)

[Ge *et al.* 2003] Yongchao Ge, Sandrine Dudoit and Terence P. Speed. *Resampling-based multiple testing for microarray data analysis.* Test, vol. 12, no. 1, pages 1–77, 2003. (Cited on page 83.)

[Geraci *et al.* 2006] Filippo Geraci, Marco Pellegrini, Marco Maggini and Fabrizio Sebastiani. *Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution.* In String Processing and Information Retrieval, pages 25–36. Springer, 2006. (Cited on page 137.)

[Godoy & Amandi 2006] Daniela Godoy and Analía Amandi. *Modeling user interests by conceptual clustering.* Information System, vol. 31, no. 4-5, pages 247–265, 2006. (Cited on page 161.)

[Goemans *et al.* 2009] Hein E. Goemans, Kristian S. Gleditsch and Giacomo Chiozza. *Introducing Archigos: A dataset of political leaders.* Journal of Peace Research, vol. 46, no. 2, pages 269–283, 2009. (Cited on page 31.)

[Gold *et al.* 2011] Bernard Gold, Nelson Morgan and Dan Ellis. Speech and audio signal processing. Wiley Online Library, 2011. (Cited on page 17.)

[Gomez & Morales 2002] Giovani Gomez and Eduardo Morales. *Automatic feature construction and a simple rule induction algorithm for skin detection.* In Proc. of the ICML workshop on Machine Learning in Computer Vision, pages 31–38, 2002. (Cited on page 64.)

[Grira *et al.* 2005] Nizar Grira, Michel Crucianu and Nozha Boujemaa. *Unsupervised and Semi-supervised Clustering: a Brief Survey.* Technical report, A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (FP6), 2005. (Cited on page 24.)

[Guille *et al.* 2013] Adrien Guille, Cécile Favre, Hakim Hacid and Djamel A. Zighed. *SONDY: An Open Source Platform for Social Dynamics Mining and Analysis.* In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13, 2013. (Cited on page 170.)

[Halkidi *et al.* 2001] Maria Halkidi, Yannis Batistakis and Michalis Vazirgiannis. *On clustering validation techniques.* Journal of Intelligent Information Systems, vol. 17, no. 2, pages 107–145, 2001. (Cited on pages 45 and 139.)

[Hammouda *et al.* 2005] Khaled M. Hammouda, Diego N. Matute and Mohamed S. Kamel. *CorePhrase: Keyphrase Extraction for Document Clustering.* Machine Learning and Data Mining in Pattern Recognition, pages 265–274, 2005. (Cited on pages 136, 137 and 138.)

[Haralick & Shanmugam 1973] R. M. Haralick and K. Shanmugam. *Computer classification of reservoir sandstones.* Geoscience Electronics, IEEE Transactions on, vol. 11, no. 4, pages 171–177, 1973. (Cited on page 102.)

[Harris 1954] Zellig S. Harris. *Distributional structure.* Word, vol. 10, pages 146–162, 1954. (Cited on page 128.)

[Harris 1968] Zellig S. Harris. Mathematical structures of language. Wiley, 1968. (Cited on pages 128 and 162.)

[Hastie *et al.* 2005] Trevor Hastie, Robert Tibshirani, Jerome Friedman and James Franklin. *The elements of statistical learning: data mining, inference and prediction.* The Mathematical Intelligencer, vol. 27, no. 2, pages 83–85, 2005. (Cited on page 17.)

[Hinneburg *et al.* 2000] Alexander Hinneburg, Charu C. Aggarwal and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? Bibliothek der Universität Konstanz, 2000. (Cited on page 19.)

[Hirsch 2005] Jorge E. Hirsch. *An index to quantify an individual's scientific research output.* Proceedings of the National Academy of Sciences of the United states of America, vol. 102, no. 46, page 16569, 2005. (Cited on page 53.)

[Hoffman *et al.* 2008] Forrest M. Hoffman, William W. Hargrove, Richard T. Mills, Salil Mahajan, David J. Erickson and Robert J. Oglesby. *Multivariate Spatio-Temporal Clustering (MSTC) as a data mining tool for environmental applications.* In International Congress on Environmental Modelling and Software, Proceedings of the iEMSs Fourth Biennial Meeting, 2008. (Cited on page 17.)

[Hofmann 1999] Thomas Hofmann. *Probabilistic latent semantic indexing.* In Research and Development in Information Retrieval, Proceedings of the 22nd annual international ACM SIGIR conference on, RDIR '99, pages 50–57. ACM, 1999. (Cited on page 134.)

[Holm 1979] Sture Holm. *A simple sequentially rejective multiple test procedure.* Scandinavian Journal of Statistics, vol. 6, no. 2, pages 65–70, 1979. (Cited on page 83.)

[Hong & Kwong 2009] Yi Hong and Sam Kwong. *Learning assignment order of instances for the constrained k-means clustering algorithm.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39, no. 2, pages 568–574, 2009. (Cited on page 26.)

[Houle *et al.* 2010] Michael E. Houle, Hans-Peter Kriegel, Peer Kröger, Erich Schubert and Arthur Zimek. *Can shared-neighbor distances defeat the curse of dimensionality?* In Scientific and Statistical Database Management, pages 482–500. Springer, 2010. (Cited on page 19.)

[Hsu & Chang 2005] Winston Hsu and Shih-Fu Chang. *Visual cue cluster construction via information bottleneck principle and kernel density estimation.* Image and Video Retrieval, pages 82–91, 2005. (Cited on page 105.)

[Huo *et al.* 2006] Xiaoming Huo, Xuelei S. Ni and Andrew K. Smith. *A survey of manifold-based learning methods.* In Mining of Enterprise Data, emerging nonparametric methodology, chapter 1, pages 06–40. Springer, New York, 2006. (Cited on pages 20 and 64.)

[Ibekwe-SanJuan & SanJuan 2004] Fidelia Ibekwe-SanJuan and Eric SanJuan. *Mining textual data through term variant clustering: the TermWatch system.* Coupling approaches, coupling media and coupling languages for information retrieval., pages 487–503, 2004. (Cited on page 137.)

[Jacquemin 1999] Christian Jacquemin. *Syntagmatic and paradigmatic representations of term variation.* In Meeting of the Association for Computational Linguistics on Computational Linguistics, Proceedings of the 37th annual, pages 341–348. Association for Computational Linguistics, 1999. (Cited on page 137.)

[Jain & Dubes 1988] Anil K Jain and Richard C Dubes. Algorithms for clustering data. Prentice-Hall, Inc., 1988. (Cited on page 132.)

[Ji *et al.* 2010] Rongrong Ji, Hongxun Yao, Xiaoshuai Sun, Bineng Zhong and Wen Gao. *Towards semantic embedding in visual vocabulary.* In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 918–925, 2010. (Cited on pages 15 and 105.)

[Jiang *et al.* 2007] Yu-Gang Jiang, Chong-Wah Ngo and Jun Yang. *Towards optimal bag-of-features for object categorization and semantic video retrieval.* In Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pages 494–501. ACM, 2007. (Cited on pages 103 and 119.)

[Jianjia & Limin 2011] Zhang Jianjia and Luo Limin. *Combined Category Visual Vocabulary: A new approach to visual vocabulary construction.* In Image and Signal Processing, 4th International Congress on, volume 3 of *CISP 2011*, pages 1409–1415, October 2011. (Cited on page 106.)

[Jones 1972] Karen Sparck Jones. *A statistical interpretation of term specificity and its application in retrieval.* Journal of documentation, vol. 28, no. 1, pages 11–21, 1972. (Cited on page 131.)

[Jurie & Triggs 2005] Frederic Jurie and Bill Triggs. *Creating efficient codebooks for visual recognition.* In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 1, pages 604–610. IEEE, 2005. (Cited on page 103.)

[Kadir & Brady 2001] Timor Kadir and Michael Brady. *Saliency, scale and image description.* International Journal of Computer Vision, vol. 45, no. 2, pages 83–105, 2001. (Cited on page 102.)

[Kandasamy & Rodrigo 2010] Kirthevasan Kandasamy and Ranga Rodrigo. *Use of a visual word dictionary for topic discovery in images.* In Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on, pages 510–515, 2010. (Cited on page 164.)

[Karkali *et al.* 2012] Margarita Karkali, Vassilis Plachouras, Constantinos Stefanatos and Michalis Vazirgiannis. *Keeping keywords fresh: a BM25 variation for personalized keyword extraction.* In Proceedings of the 2nd Temporal Web Analytics Workshop, collocated with WWW'12, TempWeb '12, pages 17–24, New York, NY, USA, 2012. ACM. (Cited on page 131.)

[Kietz *et al.* 2000] Joerg-Uwe Kietz, Alexander Maedche and Raphael Volz. *A method for semi-automatic ontology acquisition from a corporate intranet.* In Workshop on Ontologies and Text, Collocated with EKAW '2000, October 2000. (Cited on page 162.)

[Kim *et al.* 2003] Dong Kim, Jeong Sim, Heejin Park and Kunsoo Park. *Linear-time construction of suffix arrays.* In Combinatorial Pattern Matching, pages 186–199. Springer, 2003. (Cited on page 143.)

[Kinnunen *et al.* 2010] Teemu Kinnunen, Joni Kristian Kamarainen, Lasse Lensu, Jukka Lankinen and Heikki Kälviäinen. *Making Visual Object Categorization More Challenging: Randomized Caltech-101 Data Set.* In 2010 International Conference on Pattern Recognition, pages 476–479. IEEE, 2010. (Cited on page 113.)

[Kisilevich *et al.* 2010] Slava Kisilevich, Florian Mansmann, Mirco Nanni and Salvatore Rinzivillo. *Spatio-temporal clustering.* Data mining and knowledge discovery handbook, pages 855–874, 2010. (Cited on page 17.)

[Klein *et al.* 2002] Dan Klein, Sepandar D. Kamvar and Christopher D. Manning. *From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering.* In International Conference on Machine Learning, pages 307–314, 2002. (Cited on page 25.)

[Kotsiantis & Kanellopoulos 2006] Sotiris Kotsiantis and Dimitris Kanellopoulos. *Discretization techniques: A recent survey.* GESTS International Transactions on Computer Science and Engineering, vol. 32, no. 1, pages 47–58, 2006. (Cited on page 63.)

[Kriegel *et al.* 2011] Hans-Peter Kriegel, Irene Ntoutsi, Myra Spiliopoulou, Grigoris Tsoumakas and Arthur Zimek. *Mining complex dynamic data.* Tutorial at ECML/PKDD 2011, September 2011. (Cited on page 18.)

[Kullback & Leibler 1951] Solomon Kullback and Richard A. Leibler. *On information and sufficiency.* The Annals of Mathematical Statistics, vol. 22, no. 1, pages 79–86, 1951. (Cited on pages 25, 136 and 158.)

[Lallich & Rakotomalala 2000] Stéphane Lallich and Ricco Rakotomalala. *Fast feature selection using partial correlation for multi-valued attributes.* In Djamel A. Zighed, J. Komorowski and J. M. Zytkow, editors, Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, pages 221–231. LNAI Springer-Verlag, 2000. (Cited on pages 19 and 64.)

[Lallich *et al.* 2006] Stéphane Lallich, Olivier Teytaud and Elie Prudhomme. *Statistical inference and data mining: false discoveries control.* In COMPSTAT: proceedings in computational statistics: 17th symposium, page 325. Springer, 2006. (Cited on page 83.)

[Larsson 1998] N. Jesper Larsson. *Notes on Suffix Sorting.* Technical Report LU-CS-TR:98-199, LUNDFD6/(NFCS-3130)/1–43/(1998), Department of Computer Science, Lund University, Sweden, June 1998. (Cited on page 144.)

[Lauriston 1994] Andy Lauriston. *Automatic recognition of complex terms: Problems and the TERMINO solution.* Terminology, vol. 1, no. 1, pages 147–170, 1994. (Cited on page 137.)

[Laxman & Sastry 2006] Srivatsan Laxman and P. Shanti Sastry. *A survey of temporal data mining.* Sadhana, vol. 31, no. 2, pages 173–198, 2006. (Cited on page 17.)

[Lazebnik & Raginsky 2009] Svetlana Lazebnik and Maxim Raginsky. *Supervised learning of quantizer codebooks by information loss minimization.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 7, pages 1294–1309, 2009. (Cited on page 105.)

[Lazebnik *et al.* 2003a] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *Affine-invariant local descriptors and neighborhood statistics for texture recognition.* In Computer Vision, 2003. Proceedings of the Ninth IEEE International Conference on, ICCV 2003, pages 649–655. IEEE, 2003. (Cited on page 102.)

[Lazebnik *et al.* 2003b] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *A sparse texture representation using affine-invariant regions.* In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 2, pages II–319. IEEE, 2003. (Cited on page 102.)

[Lazebnik *et al.* 2006] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories.* In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 2169–2178. IEEE, 2006. (Cited on pages 102 and 103.)

[Lee & Seung 1999] Daniel D. Lee and H. Sebastian Seung. *Learning the parts of objects by non-negative matrix factorization.* Nature, vol. 401, no. 6755, pages 788–791, 1999. (Cited on page 134.)

[Lesk 1986] Michael Lesk. *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone.* In SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation, pages 24–26, New York, NY, USA, 1986. ACM. (Cited on page 162.)

[Lesot *et al.* 2009] Marie-Jeanne Lesot, Maria Rifqi and Hamid Benhadda. *Similarity measures for binary and numerical data: a survey.* International Journal of Knowledge Engineering and Soft Data Paradigms, vol. 1, no. 1, pages 63–84, 2009. (Cited on page 24.)

[Lin & Hauptmann 2006] Wei-Hao Lin and Er Hauptmann. *Structuring continuous video recordings of everyday life using time-constrained clustering.* In IS&T/SPIE Symposium on Electronic Imaging, 2006. (Cited on pages 26, 34, 37, 46 and 50.)

[Lindeberg 1993] Tony Lindeberg. *Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention.* International Journal of Computer Vision, vol. 11, no. 3, pages 283–318, 1993. (Cited on page 102.)

[Liu & Motoda 1998] Huan Liu and Hiroshi Motoda. Feature extraction, construction and selection: A data mining perspective. Springer, 1998. (Cited on page 63.)

[Liu *et al.* 2007] Ying Liu, Dengsheng Zhang, Guojun Lu and Wei-Ying Ma. *A survey of content-based image retrieval with high-level semantics.* Pattern Recognition, vol. 40, no. 1, pages 262–282, 2007. (Cited on page 15.)

[Liu *et al.* 2009] Jingen Liu, Yang Yang and Mubarak Shah. *Learning semantic visual vocabularies using diffusion distance.* In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pages 461–468. IEEE, 2009. (Cited on page 105.)

[Loeff *et al.* 2006] Nicolas Loeff, Cecilia Ovesdotter Alm and David A. Forsyth. *Discriminating image senses by clustering with multimodal features.* In Proceedings of the COLING/ACL on Main conference poster sessions, pages 547–554. Association for Computational Linguistics, 2006. (Cited on page 15.)

[López-Sastre *et al.* 2010] RJ López-Sastre, T. Tuytelaars, FJ Acevedo-Rodríguez and S. Maldonado-Bascón. *Towards a more discriminative and semantic visual vocabulary.* Computer Vision and Image Understanding, vol. 115, no. 3, pages 415–425, November 2010. (Cited on page 103.)

[Lowe 1999] David G. Lowe. *Object recognition from local scale-invariant features.* In Computer Vision, The Proceedings of the Seventh IEEE International Conference on, volume 2 of *ICCV 1999*, pages 1150–1157. IEEE, 1999. (Cited on page 102.)

[Lowe 2004] David G. Lowe. *Distinctive image features from scale-invariant keypoints.* International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, 2004. (Cited on pages 100, 103, 106 and 111.)

[Lu *et al.* 2009] Zhiwu Lu, Horace H. S. Ip and Qizhen He. *Context-based multi-label image annotation.* In Conference on Image and Video Retrieval, Proceeding of the ACM International, CIVR '09, pages 1–7. ACM, 2009. (Cited on page 16.)

[MacQueen 1967] James MacQueen. *Some methods for classification and analysis of multivariate observations.* In L. M. Cam and J. Neyman, editors, Berkeley Symposium on Mathematical Statistics and Probability, Proceedings of the Fifth, volume 1, pages 281–297. University of California Press, 1967. (Cited on pages 34 and 132.)

[Manber & Myers 1993] Udi Manber and Gene Myers. *Suffix arrays: A new method for on-line string searches.* SIAM Journal on Computing, vol. 22, no. 5, pages 935–948, 1993. (Cited on pages 138 and 143.)

[Mannila *et al.* 1997] Heikki Mannila, Hannu Toivonen and Inkeri A. Verkamo. *Discovery of frequent episodes in event sequences.* Data Mining and Knowledge Discovery, vol. 1, no. 3, pages 259–289, 1997. (Cited on page 17.)

[Marcus *et al.* 2011] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden and Robert C. Miller. *Twitinfo: aggregating and visualizing microblogs for event exploration.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pages 227–236, New York, NY, USA, 2011. ACM. (Cited on page 170.)

[Maree *et al.* 2005] Raphael Maree, Pierre Geurts, Justus Piater and Louis Wehenkel. *Random subwindows for robust image classification.* In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, volume 1 of *CVPR 2005*, pages 34–40. IEEE, 2005. (Cited on page 102.)

[Maron & Kuhns 1960] Melvin Earl Maron and John L. Kuhns. *On relevance, probabilistic indexing and information retrieval.* Journal of the ACM (JACM), vol. 7, no. 3, pages 216–244, 1960. (Cited on page 129.)

[Marszałek & Schmid 2006] Marcin Marszałek and Cordelia Schmid. *Spatial weighting for bag-of-features.* In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 2118–2125. IEEE, 2006. (Cited on page 105.)

[Matheus 1990] Christopher John Matheus. *Adding domain knowledge to SBL through feature construction.* In Proceedings of the Eighth National Conference on Artificial Intelligence, pages 803–808, 1990. (Cited on page 65.)

[McCallum 2002] Andrew McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002. (Cited on pages 157, 173 and 175.)

[Mei *et al.* 2007a] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su and ChengXiang Zhai. *Topic sentiment mixture: modeling facets and opinions in weblogs.* In International Conference on World Wide Web, Proceedings of the 16th, pages 171–180. ACM, 2007. (Cited on page 136.)

[Mei *et al.* 2007b] Qiaozhu Mei, Xuehua Shen and ChengXiang Zhai. *Automatic labeling of multinomial topic models.* In International Conference on Knowledge Discovery and Data Mining, Proceedings of the 13th ACM SIGKDD, ICKDDM 2007, pages 490–499. ACM, 2007. (Cited on page 136.)

[Meyerson 2001] Adam Meyerson. *Online facility location.* In Foundations of Computer Science, Proceedings of the 42nd IEEE Symposium on, pages 426–431. IEEE, 2001. (Cited on page 103.)

[Michalski & Stepp 1983] Ryszard S. Michalski and Robert E. Stepp. *Learning from observation: Conceptual clustering.* Machine Learning: An artificial intelligence approach, vol. 1, pages 331–363, 1983. (Cited on page 161.)

[Michalski 1983] Ryszard S. Michalski. *A theory and methodology of inductive learning.* Artificial Intelligence, vol. 20, no. 2, pages 111–161, 1983. (Cited on page 64.)

[Mikolajczyk & Schmid 2004] Krystian Mikolajczyk and Cordelia Schmid. *Scale & affine invariant interest point detectors.* International Journal of Computer Vision, vol. 60, no. 1, pages 63–86, 2004. (Cited on page 102.)

[Miller 1995] George A. Miller. *WordNet: a lexical database for English.* Communications of the ACM, vol. 38, no. 11, pages 39–41, 1995. (Cited on pages 126, 140, 149 and 162.)

[Mo & Huang 2011] D. Mo and S. H. Huang. *Feature selection based on inference correlation.* Intelligent Data Analysis, vol. 15, no. 3, pages 375–398, 2011. (Cited on pages 19 and 64.)

[Mooney *et al.* 2008] Raymond J. Mooney, Sonal Gupta, Joohyun Kim and Kristen Grauman. *Watch, Listen & Learn: Co-training on Captioned Images and Videos.* Machine Learning and Knowledge Discovery in Databases, pages 457–472, September 2008. (Cited on pages 15 and 104.)

[Moosmann *et al.* 2007] Frank Moosmann, Bill Triggs and Frederic Jurie. *Fast discriminative visual codebooks using randomized clustering forests.* Advances in neural information processing systems, vol. 19, page 985, 2007. (Cited on page 103.)

[Morsillo *et al.* 2009] Nicholas Morsillo, Christopher Pal and Randal Nelson. *Semi-supervised learning of visual classifiers from web images and text.* In International Joint Conference on Artificial Intelligence, Proceedings of the 21st, IJCAI 2009, pages 1169–1174. Morgan Kaufmann Publishers Inc., 2009. (Cited on page 104.)

[Motoda & Liu 2002] Hiroshi Motoda and Huan Liu. *Feature selection, extraction and construction.* Communication of IICM (Institute of Information and Computing Machinery), vol. 5, pages 67–72, 2002. (Cited on page 64.)

[Murphy & Pazzani 1991] Patrick M. Murphy and Michael J. Pazzani. *ID2-of-3: Constructive induction of M-of-N concepts for discriminators in decision trees.* In Proceedings of the Eighth International Workshop on Machine Learning, pages 183–187, 1991. (Cited on page 66.)

[Musat *et al.* 2011a] Claudiu Musat, Julien Velcin, Marian-Andrei Rizoiu and Stefan Trausan-Matu. *Concept-based Topic Model Improvement.* In International Symposium on Methodologies for Intelligent Systems, volume 369 of *ISMIS 2011*, pages 133–142. Springer, June 2011. (Cited on pages 127, 140, 164 and 180.)

[Musat *et al.* 2011b] Claudiu Musat, Julien Velcin, Stefan Trausan-Matu and Marian-Andrei Rizoiu. *Improving topic evaluation using conceptual knowledge.* In International Joint Conference on Artificial Intelligence, Proceedings of the Twenty-Second, volume 3 of *IJCAI 2011*, pages 1866–1871. AAAI Press, 2011. (Cited on pages 127, 140, 157, 164 and 180.)

[Musat 2011] Claudiu Musat. *The Analysis of Implicit Opinions in Economic Texts and Relations between Them.* PhD thesis, Polytechnic University of Bucharest, Bucharest, Romania, October 2011. (Cited on page 140.)

[Nanni & Pedreschi 2006] Mirco Nanni and Dino Pedreschi. *Time-focused clustering of trajectories of moving objects.* Journal of Intelligent Information Systems, vol. 27, no. 3, pages 267–289, 2006. (Cited on page 17.)

[Navarro 2001] Gonzalo Navarro. *A guided tour to approximate string matching.* ACM computing surveys (CSUR), vol. 33, no. 1, pages 31–88, 2001. (Cited on page 17.)

[Newman *et al.* 2010] David Newman, Jey Han Lau, Karl Grieser and Timothy Baldwin. *Automatic evaluation of topic coherence.* In North American Chapter of the Association for Computational Linguistics, Human Language Technologies: The 2010

Annual Conference of the, pages 100–108. Association for Computational Linguistics, 2010. (Cited on pages 140 and 157.)

[Ng *et al.* 2002] Andrew Y. Ng, Michael I. Jordan, Yair Weiss*et al. On spectral clustering: Analysis and an algorithm.* Advances in Neural Information Processing Systems, vol. 2, pages 849–856, 2002. (Cited on page 132.)

[Norris 2008] Pippa Norris. Driving democracy: do power-sharing institutions work? Cambridge University Press New York, 2008. (Cited on page 31.)

[Nowak *et al.* 2006] Eric Nowak, Frederic Jurie and Bill Triggs. *Sampling strategies for bag-of-features image classification.* Computer Vision–ECCV 2006, pages 490–503, 2006. (Cited on pages 102 and 103.)

[O'Hara & Draper 2011] Stephen O'Hara and Bruce A Draper. *Introduction to the bag of features paradigm for image classification and retrieval.* Technical report, Cornell University Library, 2011. arXiv preprint arXiv:1101.3354. (Cited on page 102.)

[Oliva & Torralba 2001] Aude Oliva and Antonio Torralba. *Modeling the shape of the scene: A holistic representation of the spatial envelope.* International Journal of Computer Vision, vol. 42, no. 3, pages 145–175, 2001. (Cited on page 104.)

[Osato *et al.* 2002] Naoki Osato, Masayoshi Itoh, Hideaki Konno, Shinji Kondo, Kazuhiro Shibata, Piero Carninci, Toshiyuki Shiraki, Akira Shinagawa, Takahiro Arakawa and Shoshi Kikuchi. *A computer-based method of selecting clones for a full-length cDNA project: simultaneous collection of negligibly redundant and variant cDNAs.* Genome research, vol. 12, no. 7, pages 1127–1134, 2002. (Cited on page 17.)

[O'Shaughnessy 2000] Douglas O'Shaughnessy. Speech communication: Human and machine. IEEE Press, Piscataway, USA, 2nd ed. édition, 2000. (Cited on page 17.)

[Osinski & Weiss 2004] Stanislaw Osinski and Dawid Weiss. *Conceptual Clustering Using Lingo Algorithm: Evaluation on Open Directory Project Data.* In Intelligent Information Systems, pages 369–377, 2004. (Cited on page 138.)

[Osinski *et al.* 2004] Stanislaw Osinski, Jerzy Stefanowski and Dawid Weiss. *Lingo: Search results clustering algorithm based on singular value decomposition.* In Intelligent information processing and web mining, Proceedings of the International IIS, IIPWM'04, page 359, 2004. (Cited on page 138.)

[Osinski 2003] Stanislaw Osinski. An algorithm for clustering of web search results. Master's thesis, Poznań University of Technology, Poland, June 2003. (Cited on pages 131, 134, 136, 137, 143, 144 and 148.)

[Pagallo & Haussler 1990] Giulia Pagallo and David Haussler. *Boolean feature discovery in empirical learning.* Machine learning, vol. 5, no. 1, pages 71–99, 1990. (Cited on pages 20, 65 and 66.)

[Parsons *et al.* 2004] Lance Parsons, Ehtesham Haque and Huan Liu. *Subspace clustering for high dimensional data: a review.* ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pages 90–105, 2004. (Cited on page 133.)

[Patwardhan & Pedersen 2006] Siddharth Patwardhan and Ted Pedersen. *Using WordNet-based context vectors to estimate the semantic relatedness of concepts.* In Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics

Together, Collocated with EACL 2006, volume 1501 of *EACL '06*, pages 1–8, 2006. (Cited on page 151.)

[Pedersen *et al.* 2004] Ted Pedersen, Siddharth Patwardhan and Jason Michelizzi. *WordNet:: Similarity: measuring the relatedness of concepts.* In Demonstration Track at North American Chapter of the Association for Computational Linguistics: Human Language Technologies, HLT-NAACL '04, pages 38–41. Association for Computational Linguistics, 2004. (Cited on page 151.)

[Perronnin *et al.* 2006] Florent Perronnin, Christopher R. Dance, Gabriela Csurka and Marco Bressan. *Adapted vocabularies for generic visual categorization.* Computer Vision–ECCV 2006, pages 464–475, 2006. (Cited on pages 105 and 106.)

[Plamondon & Srihari 2000] Réjean Plamondon and Sargur N. Srihari. *Online and off-line handwriting recognition: a comprehensive survey.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 1, pages 63–84, 2000. (Cited on page 17.)

[Porter 1980] Martin F. Porter. *An algorithm for suffix stripping.* Program, vol. 14, no. 3, 1980. (Cited on page 129.)

[Qamra *et al.* 2006] Arun Qamra, Belle Tseng and Edward Y. Chang. *Mining blog stories using community-based and temporal clustering.* In Information and Knowledge Management, Proceedings of the 15th ACM international conference on, pages 58–67, New York, NY, USA, 2006. ACM. (Cited on page 34.)

[Quattoni *et al.* 2007] Ariadna Quattoni, Michael Collins and Trevor Darrell. *Learning visual representations using images with captions.* In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. (Cited on page 15.)

[Quelhas *et al.* 2005] Pedro Quelhas, Florent Monay, J-M Odobez, Daniel Gatica-Perez, Tinne Tuytelaars and Luc Van Gool. *Modeling scenes with local descriptors and latent aspects.* In Computer Vision, Tenth IEEE International Conference on, volume 1 of *ICCV 2005*, pages 883–890. IEEE, 2005. (Cited on page 100.)

[Quinlan 1986] John R. Quinlan. *Induction of decision trees.* Machine learning, vol. 1, no. 1, pages 81–106, 1986. (Cited on page 66.)

[Quinlan 1993] John R. Quinlan. C4.5: programs for machine learning. Morgan Kaufmann, 1993. (Cited on page 66.)

[Rizoiu & Velcin 2011] Marian-Andrei Rizoiu and Julien Velcin. *Topic Extraction for Ontology Learning.* In Wilson Wong, Wei Liu and Mohammed Bennamoun, editors, Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances, chapter 3, pages 38–61. Hershey, PA: Information Science Reference, 2011. (Cited on pages 127, 164 and 180.)

[Rizoiu *et al.* 2010] Marian-Andrei Rizoiu, Julien Velcin and Jean-Hugues Chauchat. *Regrouper les données textuelles et nommer les groupes à l'aide des classes recouvrantes.* In Extraction et Gestion des Connaissances, (EGC 10) 10ème Conférence, volume E-19 of *Revue des Nouvelles Technologies de l'Information*, pages 561–572. Cépaduès, January 2010. (Cited on pages 127, 139, 164, 173, 175 and 180.)

[Rizoiu *et al.* 2012] Marian-Andrei Rizoiu, Julien Velcin and Stéphane Lallich. *Structuring typical evolutions using Temporal-Driven Constrained Clustering.* In International Conference on Tools with Artificial Intelligence, Proceedings of the Twenty-Forth, ICTAI 2012, pages 610–617. IEEE, November 2012. (Cited on pages 57 and 179.)

[Rizoiu *et al.* 2013a] Marian-Andrei Rizoiu, Julien Velcin and Stéphane Lallich. *Unsupervised Feature Construction for Improving Data Representation and Semantics.* Journal of Intelligent Information Systems, vol. 40, no. 3, pages 501–527, 2013. (Cited on pages 95 and 179.)

[Rizoiu *et al.* 2013b] Marian-Andrei Rizoiu, Julien Velcin and Stéphane Lallich. *Visual Vocabulary Construction for Image Classification in a Weakly Supervised Context.* International Journal of Artificial Intelligence Tools, 2013. (Cited on pages 123 and 180.)

[Robertson *et al.* 1995] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu and Mike Gatford. *Okapi at TREC-3.* NIST Special Publication SP, pages 109–109, 1995. (Cited on page 131.)

[Roche 2004] Mathieu Roche. *Intégration de la construction de la terminologie de domaines spécialisés dans un processus global de fouille de textes.* Thèse de doctorat université de paris 11, Université de Paris 11, December 2004. (Cited on pages 131, 137, 138 and 139.)

[Rodríguez 2005] Carlos C. Rodríguez. *The ABC of Model Selection: AIC, BIC and the New CIC.* In Bayesian Inference and Maximum Entropy Methods in Science and Engineering, volume 803, pages 80–87, 2005. (Cited on page 135.)

[Russell *et al.* 2008] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy and William T. Freeman. *LabelMe: a database and web-based tool for image annotation.* International Journal of Computer Vision, vol. 77, no. 1, pages 157–173, 2008. (Cited on page 76.)

[Salton & Buckley 1988] Gerard Salton and Christopher Buckley. *Term-weighting approaches in automatic text retrieval.* Information Processing & Management, vol. 24, no. 5, pages 513–523, 1988. (Cited on pages 130 and 137.)

[Salton *et al.* 1975] Gerard Salton, Anita Wong and Chung-Shu Yang. *A vector space model for automatic indexing.* Communications of the ACM, vol. 18, no. 11, pages 613–620, November 1975. (Cited on page 129.)

[Sanders & Sukthankar 2001] Brandon C. S. Sanders and Rahul Sukthankar. *Unsupervised discovery of objects using temporal coherence.* Technical report, CVPR Technical Sketch, 2001. (Cited on page 34.)

[Sawaragi *et al.* 1985] Y. Sawaragi, H. Nakayama and T. Tanino. Theory of multiobjective optimization, volume 176. Academic Press New York, 1985. (Cited on pages 63 and 74.)

[Sethi *et al.* 2001] Ishwar K. Sethi, Ioana L. Coman and Daniela Stan. *Mining association rules between low-level image features and high-level concepts.* In Aerospace/Defense Sensing, Simulation, and Controls, pages 279–290. International Society for Optics and Photonics, 2001. (Cited on page 15.)

[Seung & Lee 2001] H. Sebastian Seung and Daniel D. Lee. *Algorithms for non-negative matrix factorization.* Advances in Neural Information Processing Systems, vol. 13, pages 556–562, 2001. (Cited on page 134.)

[Silberztein 1994] Max D. Silberztein. *INTEX: a corpus processing system.* In Computational Linguistics, Proceedings of the 15th conference on, volume 1, pages 579–583. Association for Computational Linguistics, 1994. (Cited on page 137.)

[Singhal 2001] Amit Singhal. *Modern information retrieval: A brief overview.* IEEE Data Engineering Bulletin, vol. 24, no. 4, pages 35–43, 2001. (Cited on page 129.)

[Sivic & Zisserman 2003] Josef Sivic and Andrew Zisserman. *Video Google: A text retrieval approach to object matching in videos.* In Computer Vision, Proceedings of the Ninth IEEE International Conference on, ICCV 2003, pages 1470–1477. IEEE, 2003. (Cited on pages 100, 102 and 103.)

[Sivic *et al.* 2005] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman and William T. Freeman. *Discovering objects and their location in images.* In Computer Vision, Tenth IEEE International Conference on, volume 1 of *ICCV 2005*, pages 370–377. IEEE, 2005. (Cited on page 100.)

[Smadja 1991] Frank A. Smadja. *From N-grams to collocations: an evaluation of Xtract.* In Association for Computational Linguistics, Proceedings of the 29th annual meeting on, pages 279–284, Morristown, NJ, USA, 1991. Association for Computational Linguistics. (Cited on pages 137 and 139.)

[Steinbach *et al.* 2000] Michael Steinbach, George Karypis and Vipin Kumar. *A Comparison of Document Clustering Techniques.* In KDD Workshop on Text Mining, volume 400, pages 525–526, 2000. (Cited on page 132.)

[Storey 2002] John D. Storey. *A direct approach to false discovery rates.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 64, no. 3, pages 479–498, 2002. (Cited on page 83.)

[Swain & Ballard 1991] Michael J. Swain and Dana H. Ballard. *Color indexing.* International Journal of Computer Vision, vol. 7, no. 1, pages 11–32, 1991. (Cited on page 102.)

[Talukdar *et al.* 2012] Partha Pratim Talukdar, Derry Wijaya and Tom Mitchell. *Coupled temporal scoping of relational facts.* In Web Search and Data Mining, Proceedings of the fifth ACM international conference on, pages 73–82. ACM, 2012. (Cited on page 34.)

[Thanopoulos *et al.* 2002] Aristomenis Nikos Thanopoulos, Nikos Fakotakis and George Kokkinakis. *Comparative Evaluation of Collocation Extraction Metrics.* In Language Resources Evaluation Conference, Proceedings of the 3rd, volume 2, pages 620–625, 2002. (Cited on page 138.)

[Tsoumakas & Katakis 2007] Grigorios Tsoumakas and Ioannis Katakis. *Multi-label classification: An overview.* International Journal of Data Warehousing and Mining (IJDWM), vol. 3, no. 3, pages 1–13, 2007. (Cited on page 94.)

[Turcato *et al.* 2000] Davide Turcato, Fred Popowich, Janine Toole, Dan Fass, Devlan Nicholson and Gordon Tisher. *Adapting a synonym database to specific domains.*

In Workshop on Recent Advances in Natural Language Processing and Information Retrieval, collocated with ACL 2000, pages 1–11, Morristown, NJ, USA, 2000. Association for Computational Linguistics. (Cited on page 162.)

[Turtle & Croft 1989] Howard Turtle and W. Bruce Croft. *Inference networks for document retrieval.* In Research and Development in Information Retrieval, Proceedings of the 13th annual international ACM SIGIR conference on, pages 1–24. ACM, 1989. (Cited on page 129.)

[Varma & Zisserman 2003] Manik Varma and Andrew Zisserman. *Texture classification: Are filter banks necessary?* In Computer Vision and Pattern Recognition, Proceedings of the IEEE computer society conference on, volume 2 of *CVPR 2003*, pages II–691. IEEE, 2003. (Cited on page 102.)

[Vogel & Schiele 2007] Julia Vogel and Bernt Schiele. *Semantic modeling of natural scenes for content-based image retrieval.* International Journal of Computer Vision, vol. 72, no. 2, pages 133–157, 2007. (Cited on page 100.)

[Wagstaff & Cardie 2000] Kiri Wagstaff and Claire Cardie. *Clustering with Instance-level Constraints.* In International Conference on Machine Learning, Proceedings of the Seventeenth, pages 1103–1110, 2000. (Cited on pages 22 and 26.)

[Wagstaff *et al.* 2001] Kiri Wagstaff, Claire Cardie, Seth Rogers and Stefan Schroedl. *Constrained K-means Clustering with Background Knowledge.* In International Conference on Machine Learning, Proceedings of the Eighteenth, pages 577–584. Morgan Kaufmann, 2001. (Cited on page 26.)

[Wallach *et al.* 2009] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov and David Mimno. *Evaluation methods for topic models.* In International Conference on Machine Learning, Proceedings of the 26th Annual, pages 1105–1112. ACM, 2009. (Cited on page 139.)

[Wang & McCallum 2006] Xuerui Wang and Andrew McCallum. *Topics over time: a non-Markov continuous-time model of topical trends.* In Knowledge discovery and data mining, Proceedings of the 12th ACM SIGKDD international conference on, pages 424–433. ACM, 2006. (Cited on page 157.)

[Wang *et al.* 2007] Xuerui Wang, Andrew McCallum and Xing Wei. *Topical n-grams: Phrase and topic discovery, with an application to information retrieval.* In International Conference on Data Mining, Proceedings of the 7th IEEE, ICDM 2007, pages 697–702. IEEE, 2007. (Cited on pages 135, 136 and 173.)

[Wang *et al.* 2008] Chong Wang, David M. Blei and David Heckerman. *Continuous time dynamic topic models.* In Conference on Uncertainty in Artificial Intelligence, Proceedings of the 23rd, 2008. (Cited on page 136.)

[Wang *et al.* 2009] Gang Wang, Derek Hoiem and David Forsyth. *Building text features for object image classification.* In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1367–1374. IEEE, 2009. (Cited on pages 15 and 104.)

[Webster & Burrough 1972] R. Webster and P. A. Burrough. *Computer-based Soil Mapping of Small Areas from Sample Data: II. Classification Smoothing.* Journal of Soil Science, vol. 23, no. 2, pages 222–234, 1972. (Cited on page 23.)

[Widmer & Ritschard 2009] Eric D. Widmer and Gilbert Ritschard. *The de-standardization of the life course: Are men and women equal?* Advances in Life Course Research, vol. 14, no. 1-2, pages 28–39, 2009. (Cited on page 34.)

[Willamowski *et al.* 2004] Jutta Willamowski, Damian Arregui, Gabriela Csurka, Christopher R. Dance and Lixin Fan. *Categorizing nine visual classes using local appearance descriptors.* In ICPR Workshop on Learning for Adaptable Visual Systems, 2004. (Cited on page 100.)

[Winn *et al.* 2005] J. Winn, A. Criminisi and T. Minka. *Object categorization by learned universal visual dictionary.* In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 2, pages 1800–1807. IEEE, 2005. (Cited on page 105.)

[Wong *et al.* 2008] Wilson Wong, Wei Liu and Mohammed Bennamoun. *Determination of unithood and termhood for term recognition.* Handbook of Research on Text and Web Mining Technologies. IGI Global, 2008. (Cited on page 161.)

[Wong *et al.* 2009] Wilson Wong, Wei Liu and Mohammed Bennamoun. *A probabilistic framework for automatic term recognition.* Intelligent Data Analysis, vol. 13, no. 4, pages 499–539, 2009. (Cited on page 161.)

[Xing *et al.* 2002] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan and Stuart Russell. *Distance Metric Learning with Application to Clustering with Side-Information.* Advances in Neural Information Processing Systems, vol. 15, pages 505–512, 2002. (Cited on page 25.)

[Xu *et al.* 2003] Wei Xu, Xin Liu and Yihong Gong. *Document clustering based on non-negative matrix factorization.* In Research and Development in Informaion Retrieval, Proceedings of the 26th international SIGIR conference on, pages 267–273. ACM, 2003. (Cited on page 134.)

[Yamamoto & Church 2001] Mikio Yamamoto and Kenneth W. Church. *Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus.* Computational Linguistics, vol. 27, no. 1, pages 1–30, 2001. (Cited on page 143.)

[Yang *et al.* 1991] Der-Shung Yang, Larry Rendell and Gunnar Blix. *A scheme for feature construction and a comparison of empirical methods.* In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, pages 699–704, 1991. (Cited on page 64.)

[Yang *et al.* 2010] Shuang Hong Yang, Jiang Bian and Hongyuan Zha. *Hybrid Generative/Discriminative Learning for Automatic Image Annotation.* In Uncertainty in Artificial Intelligence, Proceedings of the 26th Conference on, UAI 2010, Catalina Island, California, 2010. Association for Uncertainty in Artificial Intelligence (AUAI)., AUAI Press. (Cited on page 16.)

[Yeh & Yang 2008] J. Yeh and N. Yang. *Ontology Construction Based on Latent Topic Extraction in a Digital Library.* Digital Libraries: Universal and Ubiquitous Access to Information, pages 93–103, 2008. (Cited on page 162.)

[Zaiane *et al.* 2003] Osmar R. Zaiane, Simeon Simoff and Chabane Djeraba, editors. Mining multimedia and complex data, volume 2797 of *Lecture Notes in Computer Science.* Springer, 2003. (Cited on pages 13 and 14.)

[Zhang & Li 2007] Qingfu Zhang and Hui Li. *MOEA/D: A multiobjective evolutionary algorithm based on decomposition.* Evolutionary Computation, IEEE Transactions on, vol. 11, no. 6, pages 712–731, 2007. (Cited on pages 51, 57 and 184.)

[Zhang *et al.* 2007] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik and Cordelia Schmid. *Local features and kernels for classification of texture and object categories: A comprehensive study.* International Journal of Computer Vision, vol. 73, no. 2, pages 213–238, 2007. (Cited on pages 100 and 106.)

[Zhang *et al.* 2009] Wei Zhang, Akshat Surve, Xiaoli Fern and Thomas G. Dietterich. *Learning non-redundant codebooks for classifying complex objects.* In Proceedings of the 26th Annual International Conference on Machine Learning, pages 1241–1248. ACM, 2009. (Cited on page 105.)

[Zheng 1995] Zijian Zheng. *Constructing nominal X-of-N attributes.* In Proceedings of International Joint Conference On Artificial Intelligence, volume 14, pages 1064–1070, 1995. (Cited on pages 64 and 66.)

[Zheng 1996] Zijian Zheng. *A comparison of constructive induction with different types of new attribute.* Technical report, School of Computing and Mathematics, Deakin University, Geelong, 1996. (Cited on page 66.)

[Zheng 1998] Zijian Zheng. *Constructing conjunctions using systematic search on decision trees.* Knowledge-Based Systems, vol. 10, no. 7, pages 421–430, 1998. (Cited on pages 20 and 65.)

[Zhu 2005] Xiaojin Zhu. *Semi-Supervised Learning Literature Survey.* Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. (Cited on page 21.)

[Zhuang *et al.* 1999] Yueting Zhuang, Xiaoming Liu and Yunhe Pan. *Apply semantic template to support content-based image retrieval.* In Proceedings of the SPIE, Storage and Retrieval for Media Databases, volume 3972, pages 442–449, 1999. (Cited on page 15.)

[Zighed *et al.* 2009] Djamel A. Zighed, Shusaku Tsumoto, Zbigniew W. Ras and Hakim Hacid, editors. Mining complex data, volume 165 of *Studies in Computational Intelligence.* Springer, 2009. (Cited on page 10.)