Structuring typical evolutions using Temporal-Driven Constrained Clustering

Marian-Andrei Rizoiu
ERIC laboratory
University Lumière Lyon 2.
Email: Marian-Andrei.Rizoiu@univ-lyon2.fr

Julien Velcin
ERIC laboratory
University Lumière Lyon 2.
Email: Julien.Velcin@univ-lyon2.fr

Stéphane Lallich
ERIC laboratory
University Lumière Lyon 2.
Email: Stephane.Lallich@univ-lyon2.fr

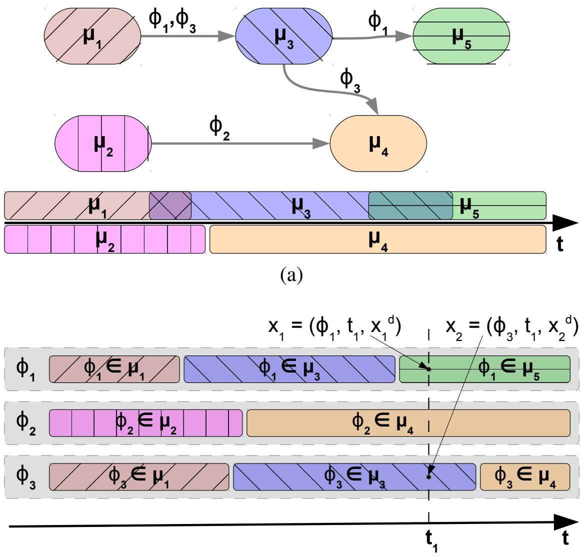
***Abstract*—In this paper, we propose a new time-aware dissimilarity measure that takes into account the temporal dimension. Observations that are close in the description space, but distant in time are considered as dissimilar. We also propose a method to enforce the segmentation contiguity, by introducing, in the objective function, a penalty term inspired from the Normal Distribution Function. We combine the two propositions into 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. This algorithm uses soft semi-supervised constraints, to encourage adjacent observations belonging to the same entity to be assigned to the same cluster. We apply our algorithm to a Political Studies dataset in order to detect typical evolution phases. We adapt the Shannon entropy in order to measure the entity contiguity, and we show that our proposition consistently improves temporal cohesion of clusters, without any significant loss in the multidimensional variance.**

***Keywords*-semi-supervised clustering, temporal clustering, temporal-aware dissimilarity measure, contiguity penalty function.**

I. Introduction

Researchers in Social Sciences and Humanities (like Political Studies) have always gathered data and compiled databases of knowledge. This information often has a temporal component, the evolution of a certain number of entities is recorded over a period of time. Each entity is described using multiple attributes, which form the multidimensional description space. Therefore, an entry in such a database would be an observation, a triple $(entity, timestamp, description)$. An observation $x_i = (\phi_l, t_m, x_i^d)$ signifies that the entity ϕ_l is described by the vector x_i^d at the moment of time t_m. Each observation belongs to an entity and, consequently, each entity is associated with multiple observations, for different moments of time. For example, a database studying the evolution of democratic states [1] will store, 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.



(a)

(b)

Fig. 1. Desired output: (a) the evolution phases and the entity trajectories, (b) the observations of 3 entities contiguously partitioned into 5 clusters.

We address these issues by proposing a novel temporal-driven constrained clustering algorithm. The proposed algorithm partitions the observations into clusters, 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 1 shows the desired result of our clustering algorithm. The observations of three entities (ϕ_1, ϕ_2 and ϕ_3) are partitioned into 5 clusters ($\mu_j, j = 1, 2, ..., 5$). In Figure 1a) 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 1b) 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 like $\mu_1 \longrightarrow \mu_2 \longrightarrow \mu_1 \longrightarrow \mu_2$ hinder the comprehension.

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 this paper, 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 inspired from 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. The graph structure represented in Figure 1a) is going to be addressed in a future work, based on the clustering results obtained using TDCK-Means.

The paper is organized as follows. In Section II we present some previous related works and, in Section III, we introduce the temporal-aware dissimilarity function, the contiguity penalty function and the TDCK-Means algorithm. In Section IV, we present the dataset that we use, the proposed evaluation measures and the obtained results. Finally, in Section V, we draw the conclusion and plan some future extensions.

II. Related work

Leveraging partial expert knowledge into clustering represents the domain of semi-supervised clustering. The expert knowledge is under the form of either class labels, or pairwise constraints. Pairwise constraints [2] are either "must-link" (the observations must be placed in the same cluster) or "cannot-link" (the two observations cannot be placed in the same cluster). Depending on the method in which supervision is introduced into clustering, [3] divides the semi-supervised clustering methods into two classes: a) the similarity-adapting methods [4]–[7], which seek to learn new similarity measures in order to satisfy the constraints, and b) the search-based methods [2], [8], [9] in which the clustering algorithm itself is modified.

The literature presents some examples of techniques which are used to segment a series of observations into continuous chunks. In [10], the daily tasks of a user are detected by segmenting scenes from the recordings of his activities. 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 [11], 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 [12], 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 setting the value of the threshold, which is usually data-dependent. Optimal matching is used in [13] 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 [14], constrained clustering is used to scope temporal relational facts in the a knowledge bases, by exploiting temporal containment, alignment, succession, and mutual exclusion constraints among facts. In [15], clustering to segment temporal observations into continuous chunks, as a preprocessing phase. A graphical model is proposed in [16], 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.

III. 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 [17] is a clustering algorithm based on iterative relocation, that partitions a dataset into k clusters, locally minimizing the total distance between the data points x_i and the cluster centroids $\mu_j \in \mathcal{M}$ (the collection of centroids). At each iteration, the objective function $\mathcal{I} = \Sigma_{\mu_j \in \mathcal{M}} \Sigma_{x_i \in \mathcal{C}_j} ||x_i^d - \mu_j^d||^2$ 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:

$$\mathcal{J} = \sum_{\mu_j \in \mathcal{M}} \sum_{x_i \in \mathcal{C}_j} \left(||x_i - \mu_j||_{TE} + \sum_{\substack{x_k \notin \mathcal{C}_j \\ x_k^\phi = x_i^\phi}} w(x_i, x_k) \right) \qquad (1)$$

where $w(x_i, x_j)$ 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.

A. *The temporal-aware dissimilarity measure*

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

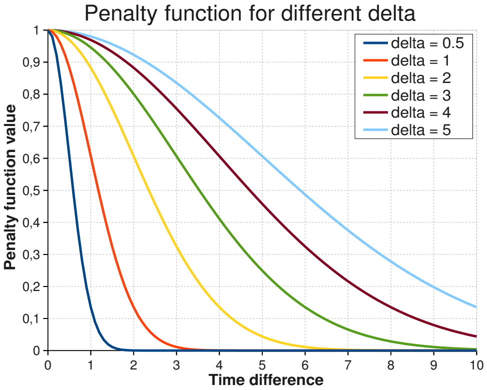


Fig. 2. Penalty function vs. time difference for multiple δ. ($\beta = 1$)

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

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:

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

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. Therefore, any algorithm that seeks to minimize an objective function based on the temporal-aware dissimilarity measure, will need to minimize both components. This makes it suitable for algorithms that search to minimize both the multidimensional and the temporal variance in clusters. Furthermore, both components that intervene in the measure follow a function like $1 - \epsilon^2, \epsilon \in [0, 1]$. 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 IV-D, 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 observations.

B. *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:

$$w(x_i, x_k) = \beta * e^{-\frac{1}{2}\left(\frac{||x_i^t - x_j^t||}{\delta}\right)^2} \mathbb{1}\left[x_i^\phi = x_j^\phi\right] \qquad (3)$$

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.

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 2. 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 2 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 [10]. The δ parameter permits the fine tuning of the penalty function. The influence of both β and δ will be studied in Section IV-E. In Section IV-D, we evaluate **Constrained K-Means**, which is an extension of K-Means, to which we add the proposed contiguity penalty function.

C. The TDCK-Means algorithm

The time dependent distance $||x_i - \mu_j||_{TE}$ 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 $\mu_j = (x_i^t, x_i^d)$. In the assignment phase, for every observation x_i, the **best_cluster** function chooses a cluster so that the temporal-aware dissimilarity measure from x_i to the clusters centroid μ_j, added to the cost of penalties possibly incurred by this cluster assignment, is minimized. It resumes to solving the following equation:

$$\underset{j=1,2,\ldots,k}{argmin} \left(||x_i - \mu_j^{(iter-1)}||_{TE}^2 + \sum_{x_k \notin \mathcal{C}_j^{(iter-1)}}^{x_k^\phi = x_i^\phi} w(x_i, x_k) \right)$$

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
    $\mu_j \leftarrow$ choose_random($\mathcal{X}$)
  end for
  $iter \leftarrow 0$
  $\mathcal{M}^{(iter)} \leftarrow \emptyset$      //set of centroids
  $\mathcal{P}^{(iter)} \leftarrow \emptyset$      //set of clusters
  repeat
    $iter \leftarrow iter + 1$
    for $j = 1, 2, ..., k$ do
      $\mathcal{C}_j^{(iter)} \leftarrow \emptyset$
    end for
    // assignment phase
    for $x_i \in \mathcal{X}$ do
      $\mathcal{C}_j^{(iter)} = \mathcal{C}_j^{(iter)} \cup x_i |$ where
                  $j =$ best_cluster($\mathcal{X}$, $\mathcal{M}^{(iter-1)}$, $\mathcal{P}^{(iter-1)}$)
    end for
    // centroids update phase
    for $j = 1, 2, ..., k$ do
      $(\mu_j^{\phi,(iter)}, \mu_j^{t,(iter)}) \leftarrow$ update_centroid
                  ($j$, $\mathcal{X}$, $\mathcal{M}^{(iter-1)}$, $\mathcal{P}^{(iter-1)}$)
    end for
    $\mathcal{M}^{(iter)} \leftarrow \{\mu_j^{(iter)} | j = 1, 2, ..., k\}$
    $\mathcal{P}^{(iter)} \leftarrow \{\mathcal{C}_j^{(iter)} | j = 1, 2, ..., k\}$
  until $\mathcal{C}_j^{(iter)} = \mathcal{C}_j^{(iter-1)}, \forall j \in [1, k]$
```

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).

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:

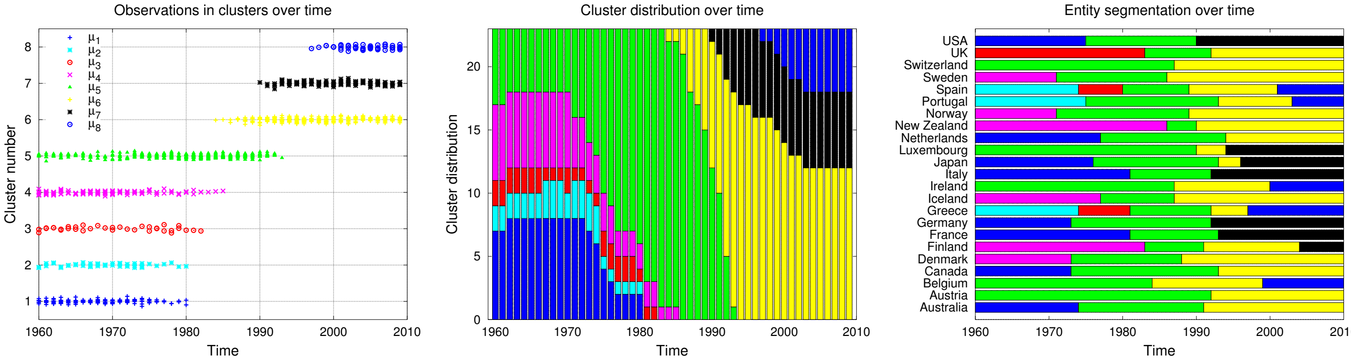
$$\frac{\partial \mathcal{J}}{\partial \mu_j^d} = 0; \frac{\partial \mathcal{J}}{\partial \mu_j^t} = 0 \qquad (4)$$

By replacing equations (2) and (3) in (1), we obtain the following formula for the objective function:

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

$$+ \sum_{x_i \in \mathcal{X}} \sum_{x_k \notin \mathcal{C}_j} \beta * e^{-\frac{1}{2}\left(\frac{||x_i^t - x_j^t||}{\delta}\right)^2} \mathbb{1}\left[x_i^\phi = x_j^\phi\right] \qquad (5)$$

From equations (4) and (5), we obtain the centroid update


Observations in clusters over time

Cluster distribution over time

Entity segmentation over time

(a) (b) (c)

Fig. 3. Typical evolution patterns constructed by TDCK-Means on *Comparative Political Data Set I* with 8 clusters.

formulas:

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

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

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.

Temporal complexity: equation (5) shows that the temporal complexity of the TDCK-Means 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 ($p > q$).

IV. Experiments

A. Dataset

Experimentations with Time-Driven Constrained K-Means are performed on a dataset issued from political sciences: *Comparative Political Data Set I* [1]. 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 [18], 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 $(country, year, description)$.

B. 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 shows the typical evolution patterns constructed by TDCK-Means (with $\beta = 0.003$ and $\delta = 3$), when asked for 8 clusters. The distribution over time of observations in each cluster is given in Figure 3a). 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 3b) 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

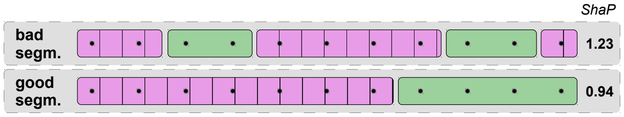


Fig. 4. Examples of a good and a bad segmentation in contiguous chunks and their related *ShaP* score.

only when studying the segmentation of countries over clusters. 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.

C. 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. Each of the three tasks proposed in Section I is evaluated separately. The mean cluster variance is traditionally used in clustering to assess how the observations in a cluster are dispersed. We use the variance to measure the dispersion of clusters both in the multidimensional space (*MDvar* measure) and in the temporal space (*Tvar* measure).

One of the initial demands was to segment the temporal series of observations of each entity into a relatively small number of contiguous segments. Each segment is a succession of observations belonging to the same cluster. We evaluate using an adapted mean Shannon entropy of clusters over entities (*ShaP* measure), which weights the entropy by a penalty factor depending on the number of continuous segments in the series of each entity. *ShaP* is calculated as:

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

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 observation for an entity. For example, in Figure 4, if the series of 11 observation 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.

D. Quantitative evaluation

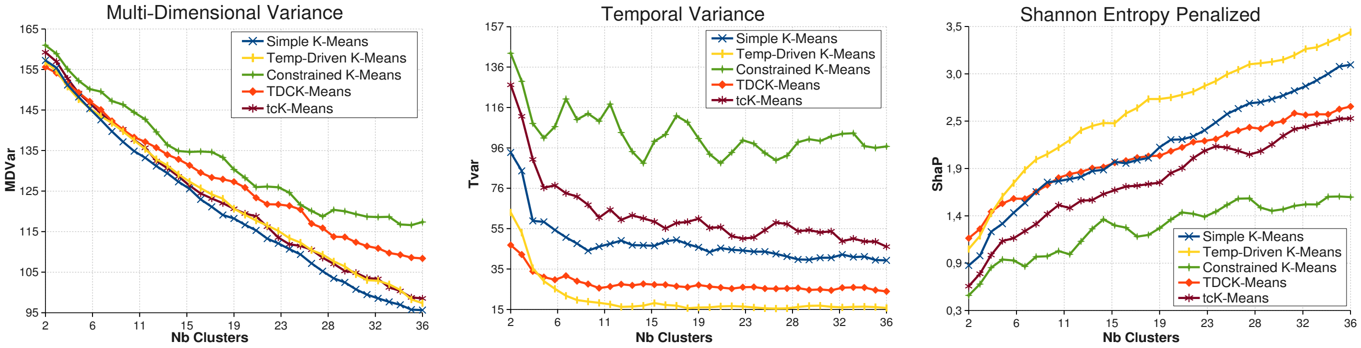
For each combination of algorithms and parameters, we execute 10 times and compute only the average and the standard deviation. We vary k, the number of clusters, from 2 to 36. The performances of five algorithms are compared from a quantitative point of view:

TABLE I
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%	

- 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 define in Section III-A;
- Constrained K-Means - uses the Euclidean distance together with the penalty component, as proposed in Section III-B. $\beta = 0.003$ and $\delta = 3$;
- TDCK-Means - the Temporal-Driven Constrained Clustering algorithm proposed in Section III-C. $\beta = 0.003$ and $\delta = 3$;
- tcK-Means - the temporal constrained clustering algorithm proposed in [10]. 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: $\alpha = 2, d = 4$.

All the parameters are determined as shown in Section IV-E. Table I 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 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 I 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 III-A, with a 58% gain of Temporal Variance and less than 2% loss of multidimensional



(a) (b) (c)

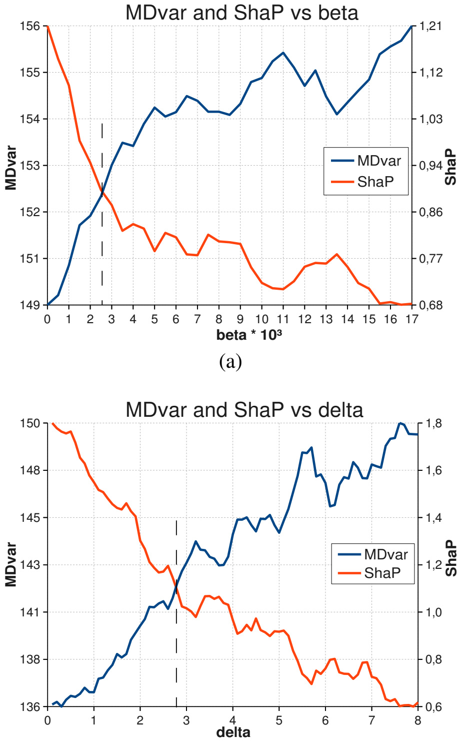
Fig. 5. *MDvar* (a), *Tvar* (b) and ShaP (c) values of the considered algorithms when varying the number of clusters.

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*.

Similar conclusions can be drawn when varying the number of clusters. *MDvar* (Figure 5a)) 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 5c), 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 5b) shows a very sharp decrease for a low number of clusters, and afterwards remains relatively constant.

E. Impact of parameters β and δ

The β parameter controls the impact of the contiguity constraints in equation (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 6a) 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



(a)

(b)

Fig. 6. *MDvar* and *ShaP* function of β (a) and of δ (b)

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). As Figure 2 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 6b) 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 variance *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 [10] 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 6a) and 6b)). 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.

V. Conclusion and future work

In this article we have studied the detection of typical evolutions from a collection of observations. We have proposed a novel way to introduce temporal information directly into the dissimilarity measure, weighting the Euclidean component in the description space by the temporal component. We have 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 are required, *e.g.* career evolution of politicians or abnormal disease evolution. In our current work, we have only detected the centroids that serve as the evolution phases. For future development, we consider generating the evolution graph (as shown in Figure 1a)), based on how the observations belonging to an entity get assigned over different clusters. This will allow an abstract succinct description of the evolution of an entity. 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 an unsupervised feature construction algorithm. We are also experimenting a method to fine tune of the ratio between the multidimensional component and the temporal component in the temporal-aware dissimilarity measure, based on the maximum and minimum values for *MDvar* and *Tvar*.

Acknowledgment

This work was partially funded by the project ImagiWeb ANR-2012-CORD-002-01.

References

[1] K. Armingeon, D. Weisstanner, S. Engler, P. Potolidis, M. Gerber, and P. Leimgruber, "Comparative political data set 1960-2009," Inst. Pol. Science, Univ. Berne., 2011. [Online]. Available: http://www.ipw.unibe.ch/content/team/klaus_armingeon/comparative_political_data_sets

[2] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *ICML*, 2001, p. 577.

[3] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," MUSCLE European Network of Excellence, Tech. Rep., 2005.

[4] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Knowledge Discovery and Data Mining*, 2003, pp. 39–48.

[5] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised clustering with user feedback," in *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, 2003, vol. 4, no. 1, pp. 17–32.

[6] D. Klein, S. D. Kamvar, and C. D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," in *ICML*, 2002, pp. 307–314.

[7] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," *Advances in Neural Information Processing Systems*, vol. 15, pp. 505–512, 2002.

[8] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised clustering by seeding," in *ICML*, 2002, pp. 19–26.

[9] A. Demiriz, K. Bennett, and M. J. Embrechts, "Semi-supervised clustering using genetic algorithms," in *Artificial Neural Networks in Engineering*, 1999, pp. 809–814.

[10] W.-H. Lin and E. Hauptmann, "Structuring continuous video recordings of everyday life using time-constrained clustering," in *IS&T/SPIE Symposium on Electronic Imaging*, 2006.

[11] F. De la Torre and C. Agell, "Multimodal diaries," in *Multimedia and Expo, IEEE International Conference*. IEEE, 2007, pp. 839–842.

[12] B. C. S. Sanders and R. Sukthankar, "Unsupervised discovery of objects using temporal coherence," CVPR Technical Sketch, Tech. Rep., 2001.

[13] E. D. Widmer and G. Ritschard, "The de-standardization of the life course: Are men and women equal?" *Advances in Life Course Research*, vol. 14, no. 1-2, pp. 28–39, 2009.

[14] P. P. Talukdar, D. Wijaya, and T. Mitchell, "Coupled temporal scoping of relational facts," in *Web Search and Data Mining*, 2012, pp. 73–82.

[15] S. Chen, H. Wang, and S. Zhou, "Concept clustering of evolving data," in *Data Engineering*, 2009, pp. 1327–1330.

[16] A. Qamra, B. Tseng, and E. Y. Chang, "Mining blog stories using community-based and temporal clustering," in *IKM*, 2006, pp. 58–67.

[17] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.

[18] B. Dormont, "Petite apologie des données de panel," *Économie & prévision*, vol. 87, no. 1, pp. 19–32, 1989.