Categorisation of Urban Water Consumptions

Joaquim Leitão*
*jpleitao@dei.uc.pt
CISUC, Department of Informatics Engineering, Univesity of Coimbra, Portugal

Abstract—Lol, here goes the abstract

Index Terms—Water Consumption Patterns, Clustering, Time series, Pattern Recognition.

1 Introduction

In most cities and urban areas investment in equipment related to water supply and drainage is substantial, being responsible for a considerable share of the budget of many municipalities.

Substantial attention has been given to the construction and installation of new pipe and sewage infrastructures, as well as to maintenance operations on existing infrastructures; however, relatively small efforts have been made to study water consumption patterns in these environments.

Indeed, understanding and accurately predicting such behaviours is extremely important to water companies, as it allows them to improve the management of their infrastructures, namely water storage facilities (such as water tanks and towers): Many water utilities repeatedly overestimate the volume of water needed to supply the population of a given region. Therefore, water companies tend to operate existing water storage tanks close to their full capacity, resulting in higher energy-related costs (since water has to be pumped into the tanks) and in larger volumes of retained water.

By being capable of characterising water consumption patterns in a given region water companies could adjust water volumes in existing storage tanks[1], improving the management of their finances and reducing excess water volumes retained in storage tanks.

The current work aims at categorising water consumptions in a given region of a median size city, contributing towards an improved understanding and knowledge of water consumption patterns. Therefore, in this work, historic water consumptions are intended to be processed and organised in different groups according to their similarity.

In the field of pattern recognition, tasks of this nature are referred to as *clustering* approaches. Since a time-ordered series of data (water consumptions, in this case) is intended to be processed and clustered, the proposed problem can be considered as *time series clustering*. In this sense, it is important to define this term. We present a definition proposed by Aghabozorgi *et al.* [1]:

Definition 1 (Time Series Clustering)**.** *"Given a dataset of n time series data $D = \{F_1, \cdots, F_n\}$, time series clustering can be defined as the process of unsupervised partitioning of D into $C = C_1, \cdots, C_k$ in such a way that homogeneous time series are grouped together based on a certain similarity measure."*

The remainder of this document is organised as follows: Section 2 presents the main objectives of this work. In section 3 water consumption data used in this work is briefly presented. The methodology adopted throughout this work is discussed in section 4. Sections 5 to 8 cover the main steps of the work, presented in the methodology section. Finally, section 9 concludes this document.

FIXME: Review outline

2 Objectives

The main objective of the proposed work is related with the categorisation of urban domestic water consumptions in a given region of a median size city.

That is, taking into account information about an entire civil year (from January 1st to December 31st) the system must be capable of identifying recurrent consumption patterns and determine the moments in time when such patterns were repeated: an intuitive (and somewhat expected) result is that water consumptions during summer months will have a similar shape, which substantially differs from those recorded during winter months. This could be justified by the fact that many people in that region usually spend summer months away, on vacation.

In this sense, the appropriate steps of a pattern recognition system will be considered, namely: (i) data pre-processing; (ii) pattern recognition techniques - in this case clustering techniques; (iii) analysis of the experimental results.

3 Dataset Description

The dataset to be used in the proposed project contains domestic water consumptions in different regions of a median size city. The available data corresponds to the entire 2016 year and was obtained as a result of a collaboration with *Águas de Coimbra*[2], the public water company in the city of Coimbra, Portgual.

Water consumption's measurement is carried out in strategic locations of the city, through measurement and control zones - *ZMC*[3]. These infrastructures provide a more efficient management of the water distribution systems, enabling the network's management in logical zones of analysis and allowing the adoption of measures to control water losses.

Collected data corresponds to a time series of total water volume distributed to the region in question - recorded as a floating point number whose units are m^3/h. Therefore, pairs of values $(time, volume)$ are provided. The collection time is provided in the format *dd/mm/yyyy HH:MM:SS*.

During the data acquisition equipment errors and malfunctions can occur, resulting in the failure to record the total

[1] For obvious reasons this volume must always be overestimated; however, such overestimation could be better controlled.

[2] http://www.aguasdecoimbra.pt/

[3] Or in portuguese, *Zonas de Medição e Controlo*.

consumptions. In such moments, a value of *"n/a"* is presented for the total distributed water volume, signalling the missing value. The developed pattern recognition system must adopt proper techniques to deal with these missing values.

4 Methodology

Extensive research has been made in the field of time series clustering, resulting in the adoption and proposal of a wide variety of clustering approaches and techniques. In a 2015 survey on time series clustering, Aghabozorgi *et al.* [1] highlights techniques such as *hierarchical clustering*, *k-means clustering* and *k-medoids clustering*. Other techniques have also been applied, such as neural networks and variations of the mentioned algorithms, namely of *k-means* ans *k-medoids*.

Determining the most appropriate technique to be applied in our problem is far from being an easy task. The same methods applied on data of distinct natures can produce completely different results, and the same can even occur with similar data, collected from distinct sources and (most probably) with a different process.

As a result, in the current work, different techniques are intended to be explored in an attempt to determine the ones that appear to be more suitable to the collected data. These techniques fall in four distinct categories that compose four main steps of time series clustering. In subsequent sections a more derailed overview of such categories will be provided.

1) **Time Series pre-processing**, comprising two major tasks: computing the unit of analysis for the time series; and imputing missing values
2) **Time Series Representation**, consisting in the application of dimensionality reduction techniques. As will be explained later in this section, both raw time series data and dimensionality reduced time series data is intended to be clustered, in order to compare the results obtained with both methods.
3) **Time Series Segmentation**, consisting in the application of clustering techniques to group similar samples together.
4) **Results analysis and assessment**

Regarding missing data imputation, three distinct techniques are considered to be applied to provide sounding estimations for missing water consumptions data: fitting a polynomial expression, fitting an ARIMA model or the application of Kalman Filters.

With respect to time series representation methods, two alternatives are intended to be considered: on one side, the popular *raw time series* representation - where time series are represented as an ordered list of values collected over time; as an alternative, the applications of dimensionality reduction methods via *Principal Components Analysis* and *Stacked Autoencoders* is also explored.

Several clustering algorithms have been proposed and studied in the literature; however, based on the studied and referenced works, more traditional algorithms such as *Hierarchical* and *K-Means* clustering still remain popular and valid choices among researchers in this field. In this sense, their performance when applied to our data will be assessed. Another algorithm applied in time series clustering problems will be implemented and evaluated: this is the case of the recently proposed *TADPole* density-based clustering algorithm, featuring a pruning strategy that softens the computational burden of the application of traditional clustering algorithms to time series.

5 Time Series Pre-Processing

As stated in section 4, time series pre-processing comprises two main iterations: computing the unit of analysis of the time series and imputing missing values in time series' readings. These steps will be covered in more detail in their own subsections.

5.1 Unit of Analysis

When identifying patterns of recurrent behaviour in time series data it is important to determine the periods of time with the higher contributions to the overall signal. In other words, the most intense and important periods of time are sought.

Similarly to the works of Abreu *et al.* [2] and Calabrese *et al.* [3], a *Fourier* analysis can be conducted on the collected time series data to determine the frequencies with higher contribution. In this sense, a *Fast Fourier Transform* (FFT) was performed on the time series data.

Analysing the results of such transformation, it was determined that the frequency with the largest contribution to the resulting signal was the frequency $0\ Hz$. Even though this may seem an unexpected result, a valid explanation can be found: by performing the FFT of a given signal, the frequency with higher contribution is usually an approximation of the fundamental frequency. As the signal being analysed (the time series data) is not periodic, its fundamental period is infinite, rendering its fundamental frequency to be $0\ Hz$. In such scenarios, the second frequency with higher contribution is usually considered as the unit of analysis. In this case such a frequency was around $24h$, suggesting this to be the unit of analysis.

With this finding, an aggregation of the time series values was conducted: the values of the original time series were recorded with a one-minute interval; since the unit of analysis is $24h$, each sample to be clustered contains readings of one day, with an interval of $1h^4$.

5.2 Missing Data Imputation

In almost all real-life applications, errors in the data collection procedures are prompted to occur, forcing data analysis and processing methods to properly deal with missing and invalid values. Substantial research has been performed on the topic of missing data imputation, resulting in different methods and techniques being proposed and tested in a variety of problems.

Nevertheless, choosing a valid missing data imputation technique to be applied in a given problem is far from being an easy task. The same method can have completely distinct performances when applied to different data, of diverse natures and/or collected in different ways. In the particular case of this work, no previous studies of the performance of missing data imputation methods had been conducted on the collected data. As a result, a choice was made to browse the literature on this topic and test the application of the most cited and used ones in the collected data.

Steffen *et al.* [4] studied the application of different missing data imputation methods on univariate time series. In their work, the authors obtained interesting results for approaches exploring linear interpolation, ARIMA and SARIMA models and Kalman Filters.

Alternative methods can also be considered. Usually more active in time series prediction and classification, techniques featuring artificial neural networks, decision tress or k-nearest neighbours [5], [6] can also be applied to compute estimations

[4] Therefore, each sample consists in a list of 24 values, one per hour. These individual hourly consumptions were obtained simply by adding all partial consumptions in the same hour.

for missing values. In addition, traditional time series analysis methods such as autocorrelation and trend analysis have also been applied [2], as well as expectation-maximization and multivariate algorithms on lagged data [4].

Inspired by the results reported on the literature, and after an initial study of the different methods, a choice was made to select the following three missing data imputation methods for further analysis: (i) linear interpolation; (ii) ARIMA; (iii) Kalman Filter.

At this point, the goal of this analysis was to studying the performance of these methods when applied to the collected data. This study was conducted as follows: Initially individual samples were divided into samples that contained missing values and samples without missing value[5]. For each complete sample missing values were artificially introduced. Each method was then applied to the individual samples and the correctness of their imputations was assessed using the *root mean squared error* (rmse) metric.

Overall the Kalman Filter produced better estimates than the competing alternatives: An average rmse of about 263.83 was registered (against a rmse of about 571.37 for the ARIMA model and 12403.17 for the linear interpolation), and the Kalman filter produced better daily estimates (smaller rmse on individual samples) in 189 days (against 72 for the ARIMA model and only 4 for the linear interpolation).

Based on such results the Kalman Filter was chosen as the imputation approach for our data. Therefore, for each day where missing data were registered, a Kalman Filter model was computed and estimations for the missing data in that day were obtained.

6 Time Series Representation

When working with high dimensional data, computational requirements grow both in terms of memory and execution time: if each time series contains more values then it will require more space in memory and, therefore, if a larger number of time series is intended to be clustered or processed in some way substantially more memory may be required to load all the time series data. In addition, specially in the case of clustering, distance computation between data samples is often performed; if a larger number of samples is considered, each comprising several dimensions, distance computation between pairs of samples will take longer to compute, as will the overall process.

Another motivation for seeking alternative time series representations, as pointed out by Aghabozorgi *et al.* [1], has to do with the fact that when measuring the distance between raw time series samples highly unintuitive results may be gathered resulting in the clustering of series similar in noise, instead of shape. It must be noticed that the adopted distance metric highly influences the clustering results.

In this sense, several time series representation methods have been proposed in the literature. Aghabozorgi *et al.* presents a taxonomy comprising four main categories: i) Data adaptive; (ii) Non-data adaptive; (iii) Model-based; and (iv) Data dictated. The choice of time series representation has a big impact in the outcome of the clustering exercise, in both quality and execution time. Furthermore, time series representation also affects the choice of distance metric (also called dissimilarity function).

In the specific clustering problem being tackled in this work, where time series containing similar behaviours are intended to be grouped together, one of the major challenges to be face has to due with the dissimilarity function (that is, how to determine that two time series are close together). Even though it is computationally expensive, the *Dynamic Time Warping* (DTW) [7] distance metric is quite popular, having been extensively adopted when dealing with raw time series data [1], [8]–[10] and being considered a more robust distance metric for time series [11].

As the DTW targets time sequences, it can be concluded that a substantial number of approaches adopt a raw time series representation. Dimensionality Reduction (DR) techniques such as *Principal Components Analysis* (PCA) have also been explored [2] in an attempt to represent time series samples in fewer dimensions, thus reducing the computational burden of distance computation. As the application of DR techniques loses the temporal sequence of the data, alternative distance metrics need to be considered.

In light of the discussion carried out in this section, in the scope of the current work, it is intended to explore clustering performance obtained using two distinct time series representation approaches: (i) **Raw time series representation**, featuring the use of the popular *Dynamic Time Warping* distance metric; (ii) **”Dimensionality-Reduced” time series representation**, featuring the use of the classical *Euclidean* distance.

6.1 Raw Time Series Representation

With respect to the *raw* time series representation not much needs to be added to what has already been mentioned. Initially, this representation can be seen as considering raw collected time series values.

Taking into account the pre-processing steps mentioned in section 5, imputed time series corresponding to daily lists of hourly water consumption readings should be considered at this point; however, an additional step must be considered: when working with raw time series data, performing a *Z-Normalization* of the data can significantly improve distance computation results with the DTW metric [12].

As a result, when working with raw time series representation, a *Z-Normalization* of the data was conducted. The *Z-Normalization* can be defined as:

$$X = \frac{X - MEAN(X)}{STD(X)}$$

6.2 Dimensionality Reduction for Time Series

Regarding the application of dimensionality reduction techniques to time series data, two popular algorithms of this nature were compared: *PCA* and *Stacked Autoencoders* (SAEs). The comparison was based on the rmse on reconstruction and on visual inspection of the reconstructed series.

In PCA the target number of features was selected as a function of the percentage of explained data variance. In this sense, principal components capable of representing 80% of the data variance were selected, yielding a reduced dimensionality of 3 features.

SAEs were then trained for dimensionality reduction, targeting the same number of features as PCA. Different network architectures were considered while training the SAEs (that is, different number of autoencoders with different numbers of neurons per hidden layer, different activation functions, etc); however, in the best scenarios, SAEs were only capable of getting close to PCA in terms of reconstruction error, producing reconstructed time series considerably more distant to the original ones than PCA.

[5] The reader must keep in mind that individual samples consist in a list with water consumptions for a given day, featuring hourly values.

It is also worth mentioning at this point that both PCA and SAEs approaches were tested with normalised data, using the *Min-Max* method. Figure 1 presents a reconstruction of water consumptions for one day using both a PCA decomposition and Stacked Autoencoders.

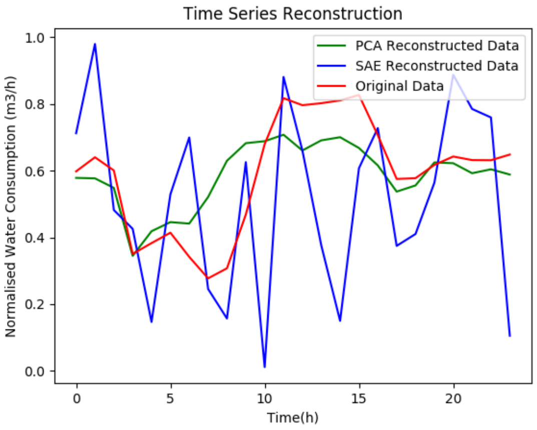

Figure 1. Reconstruction of water consumptions for one day using a PCA decomposition and a Stacked Autoencoder.

From the analysis of figure 1 it is clear that, even though SAEs may be able to reconstruct the original data with an error close to PCA, SAEs fail in capturing the shape and tendency of the series. As a proper learning could not be obtained with SAEs, PCA was the dimensionality reduction technique chosen to be applied in the experiments and exercises carried out in the remainder of this work.

7 Clustering

The choice of which clustering algorithm to apply in a given problem can strongly condition the obtained results. Several clustering algorithms have been proposed over the years and applied to a wide variety of problems.

It is common among authors to characterise and classify clustering approaches according to how the algorithms work. Aghabozorgi *et al.* [1] proposes a general categorisation of clustering algorithms applied to time series featuring six different groups: Hierarchical, Partitioning, Model-based, Density-based, Grid-based and Multi-step clustering.

As mentioned in section 4, from the conducted literature review on time series clustering *Hierarchical* and *K-Means* clustering algorithms were identified as the most popular choice among researchers.

One of the main drawbacks of most time series clustering algorithms is related with the computational burden of finding similarities in the data being clustered. As the most robust similarity measures for time series rely on a raw representation, continuously searching for patterns based on similarities/dissimilarities in the data becomes very expensive in computation time.

Motivated by the fact that robust and less demanding alternatives to metrics such as the *DTW* have not been achieved, some researchers studied ways of reducing this computational burden. This is the case of *TADPole* [13]. Proposed in 2015 by Begum *et al.*, TADPole exploits upper and lower bounds on DTW in a novel pruning strategy that avoids a large fraction of distance calculations, achieving results identical to brute force (but at least an order of magnitude faster).

In this sense, the mentioned cluster algorithms were selected for analysis at this point. Regarding HC and K-Means, as is often performed in the literature, both algorithms were combined [14]. In the remainder of this section, details regarding the implementation and experimental setup of the two sets of algorithms being compared are covered.

7.1 Combining Hierarchical and K-Means Clustering

Formulating K-Means clustering into an algorithm comprises quite simple steps. Nevertheless, this simplicity does not avoid some of the most well-known and documented drawbacks of this algorithm: K-means requires the specification of a number of clusters prior to its execution; resulting from the previous point, K-means always groups the data into the specified number of clusters, meaning even if the number of clusters is inadequate to the dataset at hand, K-means will find such a number of partitions; even on perfect datasets it can get stuck on local minimum; and lastly, the final clustering solution is very sensitive to the initial cluster selection.

To avoid (or at least soften) these problems, an hybrid approach is typically used by combining *hierarchical clustering* and *k-means* methods. The procedure is as follows:

1) Compute hierarchical clustering and cut the resulting dendrogram into k-clusters.
2) Compute the centroid of each cluster.
3) Compute K-means by using the previously computed centroids as the initial cluster selection.

7.2 Centroid Computation in K-Means

Besides specifying the number of clusters, K-Means also allows for different methods to be applied when computing the centroid representative of a given cluster. A widely used method that can be applied to many distance metrics involves computing, for each dimension of the data, the average of all the samples assigned to the cluster in question. For this reason this method is often referred to as the *average* method.

As mentioned by Aghabozorgi *et al.* [1], when working with time series data the average centroid computation method tends to only be applied when time series have equal length and a none-elastic distance metric (such as the Euclidean distance) is employed.

When time series are allowed to have different lengths, or when distance metrics of other natures are employed (namely the DTW) simply performing the mean of the time series at each point fails to capture the average shape of all the time series in the given cluster. As a result, alternative centroid computation methods more adequate to these distance metrics have been studied.

Probably the most common centroid computation method used with raw time series data is the *medoid* method. According to this method, the centroid is the sample that minimizes the sum of squared distance to all the other samples within the cluster. Examples of the application of this method can be found in [10], [15].

A more recent centroid computation method is the *DTW Barycenter Averaging* (DBA) [16], proposed in 2011 by Petitjean *et al.*. This method was specified for the DTW distance metric and, according to the authors of the cited work, is claimed to outperform all existing averaging techniques when applied to datasets of the UCR Archive [17].

When computing the cluster centroid, DBA tries to minimize the sum of squared DTW distances from the average sequence (that is, the centroid) to all the sequences assigned to the cluster in question. A local optimisation strategy is implemented, which strongly depends on an initial guess for the average sequence - the initial centroid guess. As a result, improved results are usually obtained by performing multiple random initial starts. An example study making use of the DBA to compute the centroids for K-Means clustering can be found in [9].

In short, taking into account that two distinct time series representation techniques are intended to be applied, different centroid computation methods were also implemented and the performances of the resulting clusterings compared:

- Regarding the *DTW* distance metric, the following centroid computation methods were considered: *Average*, *Medoid* and *DBA*.
- With respect to the *Euclidean* distance metric, only the *Average* centroid method was considered.

7.3 TADPole

In the beginning of the current section a brief introduction to the TADPole clustering algorithm was performed, highlighting its main characteristics and motivations leading to its development. Regarding implementation details of this algorithm, two parameters must be defined prior to its applications: the *window size* used in the DTW computations and a cutoff distance.

The TADPole algorithm makes use of a centred window when computing distances (using the DTW metric). For any observation x_i the algorithm considers the points in the range x_{i-w} and x_{i+w} in the DTW computation. As such taking into account that the DTW is only applied for raw time series representation, and that such time series have a dimensionality of 24, a window size of 23 was defined.

The second parameter that needs to be defined is the cutoff distance, often represented as d_c. In an initial step of the algorithm, upper and lower bound on the DTW are used to find time series with many close neighbours. The mentioned cutoff distance is used as a threshold when determining a time series' neighbours: Any other time series whose distance is below d_c is considered a neighbour. In the current work initial experiments were performed with different values for the d_c threshold, obtaining a value of 1.5 as the outcome. Nevertheless, further research on the most appropriate and adequate value for this parameter can still be performed.

Finally, regarding the cluster computation, TADPole takes the series that lie in dense areas (that is, the series that have many neighbours) as the cluster centroids.

8 Results Assessment

The current section covers the distinct experiments conducted in the course of this work, presenting and discussing its main findings. As acknowledged in the previous section, different configurations for the clustering process were defined. Overall, the experimental procedure and consequent results analysis can be summarised in four main steps:

1) Initially estimates for the number of clusters were obtained by performing *Hierarchical Clustering* on the time series data and analysing the obtain dendrogram.
2) Apply an hybrid clustering algorithm, where the centroids obtained for the different selected number of clusters in the previous point are used as initial centroids for the *K-Means Clustering* algorithm.
3) Apply cluster evaluation metrics to assess the quality of the clusters computed in the hybrid approach.
4) Based on the cluster quality computed in the previous point select a subset of clustering results to be further analysed. Analysis conducted at this point focuses on descriptive characteristics of the categorisation performed, namely percentage of weekdays and weekend-days assigned to each cluster, percentage of days of each month assigned to each cluster, among others.

In this sense, each of the four points mentioned in the previous listing will be covered in a separate subsection, within the remainder of the current section.

8.1 Initial Number of Cluster Estimates

Based on the experimental setup presented in table 1, and discussed throughout the document, time series' similarities were computed based on two distinct distance metrics: DTW and Euclidean. As such, hierarchical clustering using both metrics was computed to obtain the initial estimates for the number of clusters. Figures 2 and 3 present the dendrograms obtained:

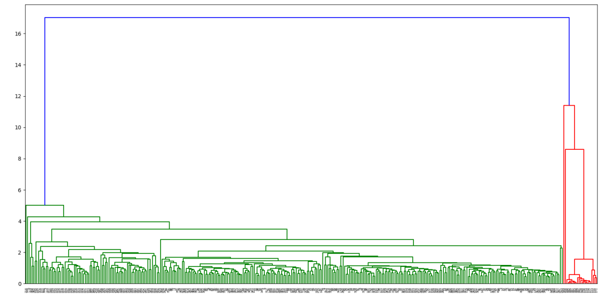

Figure 2. Dendrogram obtained using the DTW metric.

Starting with the analysis of Figure 2, referring to the hierarchical clustering of the z-normalised time series data, 2 major and uneven clusters can be identified and further analysed.

With respect to the right side of the dendrogram (represented in red) three main clusters appear to form. Further divisions can only be considered for significantly smaller distances, for which a high number of clusters needs to be considered. Analysing the left side of the dendrogram (represented in green) several divisions can be identified. For distances smaller than about 3 a high number of divisions can be identified.

Since water consumption patterns for an entire year are intended to be discovered with the goal of improving urban water management, the target number of clusters does not need to be considerably high: for example, if 300 clusters were formed it would be impossible to identify recurrent patterns of consumption.

In this sense, and considering the high number of divisions identified at the mentioned distance, a decision was made to keep it as the maximum number of clusters considered in this work. As the identified number of clusters for the mentioned distance is 8, the K-Means clustering algorithm will be applied to time series data using the DTW metric for a target number of clusters from 2 to 8. The reader's attention is drawn to Table 1, where this range is presented.

Moving on to time series data represented using dimensionality reduction techniques the scenario is different from the one presented in Figure 2: even though two main clusters can still be identified, they are not as unbalanced as the ones in the previous analysis.

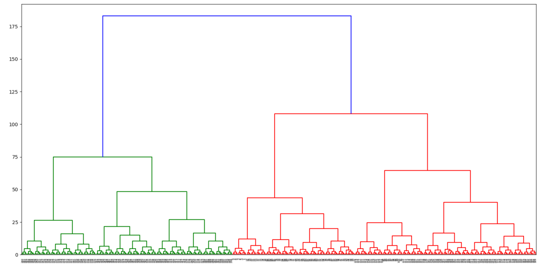

Figure 3. Dendrogram obtained using the Euclidean metric.

By using a reasoning similar to the previous analysis, for distances lower than 50 the number of clusters begins to increase rapidly, suggesting that distance to be defined as the limit for the number of clusters which, is this case, is 8. Similarly to the previous analysis, this value is presented in Table 1.

Being in a completely unsupervised scenario alternative ranges for the number of clusters could be considered. Furthermore, a cyclic process could even be considered in which the dendrogram would be initially analysed to extract estimates for the number of clusters, which would be computed using the K-Means algorithm. Upon further analysis of the results obtained with the K-Means the dendrogram could be analysed again, and new estimates for the value of k could be computed, repeating this process. Nevertheless, and because of time restrictions, this process was only performed once.

8.2 Summary of the Experiments

Following the discussion started in the previous sections, Table 1 presents a summary of the experiments performed, mentioning the algorithms and respective configurations adopted.

Table 1
Summary of the experiments performed.

Data Type	Distance Metric	Clustering Algorithm	Centroid Computation	Number of Clusters

Raw	DTW	HC
		+
		K-Means
			Average;
			DBA;
			Medoid
				2 to 8

Reduced (PCA)	Euclidean	HC + K-Means	Average	2 to 8
Raw	DTW	TADPole	-	2 to 8

8.3 Initial Results Analysis

After obtaining the initial estimations for the number of clusters the segmentation of the water consumptions must be performed. As discussed throughout the document, approaches of different natures are intended to be compared, raising the need to define metrics that assess the quality of the computed clusters, thus allowing the comparison of the different approaches.

Evaluating algorithms' performance in unsupervised problems such as clustering (without knowing the optimal number of clusters and assignments of each sample) is a challenging task and still considered an open research problem [1],mostly due to the ambiguity and subjectiveness underlining the definition of a cluster.

Popular evaluation metrics involve *internal* indexes, where cluster quality is summarised to a single score without resorting to any externally supplied labels or ground truth. Whenever such information is available *external* indexes may be applied. In the current work, a completely unsupervised scenario is faced, calling for the application of *internal* indexes.

Several internal indexes have been proposed in the literature to assess cluster quality. In the current work the following were considered:

- *Silhouette Coefficient* (SC), also referred to as silhouette index, score or value. SC measures how similar a sample is to other in its own cluster in comparison to other clusters. It is defined as the average of individual samples' silhouette values, $s(i)$, computed as:

$$s(i) = \frac{b(i) - a(i)}{max\{b(i);a(i)\}}$$

  where $a(i)$ is the average dissimilarity of sample i to all samples in the same cluster and $b(i)$ is the lowest average dissimilarity of sample i to any other cluster. SC takes a value in the range $[-1, 1]$ where values closer to 1 suggest more dense and well-separated clusters.

- *Calinski-Herabaz Index* (CH), also referred to as the *Variance Ratio Criterion* (VRC). CH is defined as follows:

$$CH = \frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)}$$

  where SS_B is the overall between-cluster variance, SS_W is the overall within-cluster variance, k is the number of clusters and N is the number of samples/observations. The SC metric is best suited for euclidean distances and, unlike the silhouette coefficient, it is not bounded: higher values of CH signal more dense and well-separated clusters.

- *Sum of Squared Errors* (SSE). In the context of clustering, the *error* of a sample can be defined as its distance to its cluster centroids. Therefore, the sum of squared errors consists in the sum of the squares of such distances. As clusters are desirably as dense as possible, smaller values of this metric suggest a more adequate cluster structure. The SSE is often defined as a measure of coherence of the computed clusters, where the smaller its value the ”better” the computed clusters.

- *Average Within-Cluster Sum Squares*. As the name suggests, this metric computed the average dissimilarity of samples belonging to the same cluster, for a given cluster composition. Such a metric is necessary for the application of the *elbow method*, a popular graphical inspection technique used to select the number of clusters for a given dataset: by plotting the value of this metric for different values of k, at some point, an angle in the graph will be verified, yielding the choice for the number of clusters.

Table 2 presents the values of the mentioned metrics for the clusters computed according to the summary presented in Table 1:

Apresentar tabela com resultados - Fazer tabela ocupar as duas colunas!

Fazer uma reflexão sobre os resultados e dizer quais aqueles que merecem ser mais estudados.

Atenção que analisei muita coisa com base no SC mas este pode nao ser o melhor método... Combinar um pouco a análise com o elbow method! (SC e um indicador normalisado, o que

Table 2
Evaluation of the clusters computed for the ranges of k obtained from the dendrograms.

Data Type	Number Clusters	Distance Metric	Centroid	Silhouette Coefficient	Calinski-Herabaz	Sum Squared Errors	Average Within-Cluster Sum Squares
raw	2	dtw	Average	0.854	702.506	2738.524	7.482
raw	3	dtw	Average	0.498	634.724	1277.772	3.491
raw	4	dtw	Average	0.494	459.867	1200.868	3.281
raw	5	dtw	Average	0.330	500.096	792.516	2.165
raw	6	dtw	Average	0.328	403.613	776.355	2.121
raw	7	dtw	Average	0.328	380.359	656.397	1.793
raw	8	dtw	Average	0.268	321.327	690.031	1.885
raw	2	dtw	DBA	0.855	702.506	2276.811	6.221
raw	3	dtw	DBA	0.798	375.516	2028.609	5.543
raw	4	dtw	DBA	0.326	553.938	1507.036	4.117
raw	5	dtw	DBA	0.324	305.732	1651.674	4.513
raw	6	dtw	DBA	0.267	338.719	1675.054	4.577
raw	7	dtw	DBA	0.247	273.032	1348.368	3.684
raw	8	dtw	DBA	0.202	258.607	1609.535	4.398
raw	2	dtw	Medoid	0.182	26.003	8125.153	22.200
raw	3	dtw	Medoid	0.198	52.744	8125.153	22.200
raw	4	dtw	Medoid	0.198	52.744	8125.153	22.200
raw	5	dtw	Medoid	0.198	52.744	8125.153	22.200
raw	6	dtw	Medoid	0.198	52.744	8125.153	22.200
raw	7	dtw	Medoid	0.121	52.146	8125.153	22.200
raw	8	dtw	Medoid	0.114	50.424	8125.153	22.200
reduced	2	euclidean	Average	0.625	1091.771	1021632.989	2791.347
reduced	3	euclidean	Average	0.587	1451.359	454165.827	1240.890
reduced	4	euclidean	Average	0.567	1807.807	255658.371	698.520
reduced	5	euclidean	Average	0.555	2162.009	163725.151	447.336
reduced	6	euclidean	Average	0.546	2503.043	114244.296	312.143
reduced	7	euclidean	Average	0.539	2851.059	83985.351	229.468
reduced	8	euclidean	Average	0.535	3198.039	64312.803	175.718
raw	2	dtw	TADPole	0.901	3315.641	39732.457	108.558
raw	3	dtw	TADPole	-0.100	110.288	39517.909	107.972
raw	4	dtw	TADPole	0.036	95.061	35628.406	97.345
raw	5	dtw	TADPole	-0.124	126.863	35393.535	96.704
raw	6	dtw	TADPole	-0.211	101.870	35300.627	96.45
raw	7	dtw	TADPole	-0.235	84.550	35311.599	96.480
raw	8	dtw	TADPole	-0.358	72.449	35278.512	96.389

tem vantagens porque nao e tao sujeito a subjectividade como e o elbow method, mas por vezes em dados reais pode ser necessario experiencia e analise visual que o SC nao consegue traduzir)

FIXME: Não esquecer de falar do facto de o SC poder nao ser a metrica mais adequada para avaliar os clusters!

8.4 Further Results Analysis

Apresentar novas métricas, resultados e reflexão

Meter gráficos aqui

9 Conclusion

De uma forma geral o numero de clusters parece andar proximo de 4, pelo menos andamos sempre a volta desse valor...

References

[1] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[2] J. M. Abreu, F. C. Pereira, and P. Ferrão, "Using pattern recognition to identify habitual behavior in residential electricity consumption," *Energy and buildings*, vol. 49, pp. 479–487, 2012.

[3] F. Calabrese, J. Reades, and C. Ratti, "Eigenplaces: segmenting space through digital signatures," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 78–84, 2010.

[4] S. Moritz, A. Sardá, T. Bartz-Beielstein, M. Zaefferer, and J. Stork, "Comparison of different methods for univariate time series imputation in r," *arXiv preprint arXiv:1510.03924*, 2015.

[5] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for dna microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[6] F. Yu and X. Xu, "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved bp neural network," *Applied Energy*, vol. 134, pp. 102–113, 2014.

[7] S. Chu, E. Keogh, D. Hart, and M. Pazzani, "Iterative deepening dynamic time warping for time series," in *Proceedings of the 2002 SIAM International Conference on Data Mining*, pp. 195–212, SIAM, 2002.

[8] T. W. Liao, "Clustering of time series data—a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[9] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, "Dynamic time warping averaging of time series allows faster and more accurate classification," in *Data Mining (ICDM), 2014 IEEE International Conference on*, pp. 470–479, IEEE, 2014.

[10] H. Izakian, W. Pedrycz, and I. Jamal, "Fuzzy clustering of time series data using dynamic time warping distance," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 235–244, 2015.

[11] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, pp. 1–35, 2013.

[12] A. Mueen and E. Keogh, "Extracting optimal performance from dynamic time warping," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2129–2130, ACM, 2016.

[13] N. Begum, L. Ulanova, J. Wang, and E. Keogh, "Accelerating

dynamic time warping clustering with a novel admissible pruning strategy," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 49–58, ACM, 2015.

[14] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[15] T. W. Liao, C.-F. Ting, and P.-C. Chang, "An adaptive genetic clustering method for exploratory mining of feature vector and time series data," *International Journal of Production Research*, vol. 44, no. 14, pp. 2731–2748, 2006.

[16] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011.

[17] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana, "The ucr time series classification/clustering homepage," *URL= http://www. cs. ucr. edu/eamonn/time series data*, 2006.