Time Series Clustering Procedure

by Aleksandras Urbonas ([aleksandras@neurotech.lt](mailto:aleksandras@neurotech.lt)) on 2019-12-30

# Introduction

Clustering is an unsupervised data mining technique with a goal to form homogeneous groups, or clusters of objects, with minimum inter-cluster and maximum inter-cluster similarity. Clustering is also an exploratory technique in time-series visualization.

The time series clustering is used as an independent research technique, and as a part of more complex data mining methods, such as rule detection, classification, anomaly detection, and so on.

There are several approaches to time series clustering. Three basic approaches of time series clustering are proposed: raw-data-based, feature-based, model-based [1, 6].

In case of raw-data-based clustering the objects are raw data in the frequency or time domain.

## Importance of clustering

To highlight the importance and the need for clustering time-series data-sets, potentially overlapping objectives for clustering of time-series data are given as follows:

1. Time-series databases contain valuable information that can be obtained through pattern discovery. Clustering is a common solution performed to uncover these patterns on time-series data-sets.
2. Time-series databases are very large and cannot be handled well by human inspectors. Hence, many users prefer to deal with structured data-sets rather than very large data-sets. As a result, time-series data are represented as a set of groups of similar time-series by aggregation of data in non-overlapping clusters or by a taxonomy as a hierarchy of abstract concepts.
3. Time-series clustering is the most-used approach as an exploratory technique, and also as a subroutine in more complex data mining algorithms, such as rule discovery, indexing, classification, and anomaly detection [22].
4. Representing time-series cluster structures as visual images (visualization of time-series data) can help users quickly understand the structure of data, clusters, anomalies, and other regularities in data-sets.

## Challenges

* Time-series clustering is a challenging issue because first of all, time-series data are often far larger than memory size and consequently they are stored on disks. This leads to an exponential decrease in speed of the clustering process.
* Second challenge is that time-series data are often high dimensional [23,24] which makes handling these data difficult for many clustering algorithms [25] and also slows down the process of clustering [26].
* Finally, the third challenge addresses the similarity measures that are used to make the clusters. To do so, similar time-series should be found which needs time-series similarity matching that is the process of calculating the similarity among the whole time-series using a similarity measure. This process is also known as “whole sequence matching” where whole lengths of time-series are considered during distance calculation. However, the process is complicated, because time-series data are naturally noisy and include outliers and shifts [18], at the other hand the length of time-series varies and the distance among them needs to be calculated. These common issues have made the similarity measure a major challenge for data miners

Clustering of time-series data is mostly utilized for discovery of interesting patterns in time-series datasets [27,28]. This task itself, fall into two categories:

1. The first group is the one which is used to find patterns that frequently appears in the data-set [29,30].
2. The second group are methods to discover patterns which happened in data-sets surprisingly [31–34].

Briefly, finding the clusters of time-series can be advantageous in different domains to answer following real world problems:

* Anomaly, novelty or discord detection

1. Recognizing dynamic changes in time-series: detection of correlation between time-series [36]. For example, in financial databases, it can be used to find the companies with similar stock price move.
2. Prediction and recommendation: a hybrid technique combining clustering and function approximation per cluster can help user to predict and recommend [37–40]. For example, in scientific databases, it can address problems such as finding the patterns of solar magnetic wind to predict today’s pattern.
3. Pattern discovery: to discover the interesting patterns in databases. For example, in marketing database, different daily patterns of sales of a specific product in a store can be discovered.

## Procedure

The following clustering procedure is used:

* Time series representation: reduce dimensionality; 115k → 256
* Distance measure: quantify dissimilarity; → L1 L2
* Prototype: summarize characteristics of all series in a cluster; → medoid, centroid, etc.
* Algorithm: choose from most common partitional or hierarchical; → DBSCAN partitional
* Cluster validity indices (CVI): calculate efficiency. → 3: ARI=0.7, 4 ARI=0.8 (max)

# Data

We consider the entire series, also known as the whole-series clustering.

Based on the amount of data, this time series can be treated as long time series.

## Feature extraction

Classification of data into groups can be performed in several ways given blast furnace sensors data: primarily, existing methods found in the literature use machine learning algorithms that group blast furnace sensors into categories through feature extraction from the blast furnace sensors data.

These features are numerical or categorical properties of the blast furnace signals which can be used to characterize and distinguish the different classes. Features can range from basic statistical properties such as the mean or the standard deviation to more complex time series characteristics such as the auto-correlation function.

These features should ideally be informative and discriminative, thus allowing for machine learning or other algorithms to use them to distinguish between classes of signals.

* basic: mean, max, min, standard deviation;
* advanced: auto-correlation.

## Transforms for Time Series Data

Four transforms are popular for *univariate time series dataset* using machine learning methods to model and make predictions:

* Power Transform:
  + Removes a shift from a data distribution to make the distribution more-normal (Gaussian). On a time-series dataset, this can have the effect of removing a change in variance over time. Popular examples are the
    - log transform (positive values) or
    - generalized versions such as the Box-Cox transform (positive values) or
    - the Yeo-Johnson transform (positive and negative values).
* Difference Transform:
  + a simple way for removing a systematic structure from the time series. For example,
    - a trend can be removed by subtracting the previous value from each value in the series. This is called first order differencing. The process can be repeated (e.g. difference the differenced series) to remove second order trends, and so on.
    - A seasonal structure can be removed in a similar way by subtracting the observation from the prior season, e.g. 12 time steps ago for monthly data with a yearly seasonal structure.
* Standardization
* Normalization

# Representation

Time-series dimension reduction also known as time series representation is a common solution for most whole time-series clustering approaches proposed in the literature [9,80–82]. Dimensionality reduction represents the raw time-series in another space:

* by transforming time-series to a lower dimensional space, or
* by feature extraction.

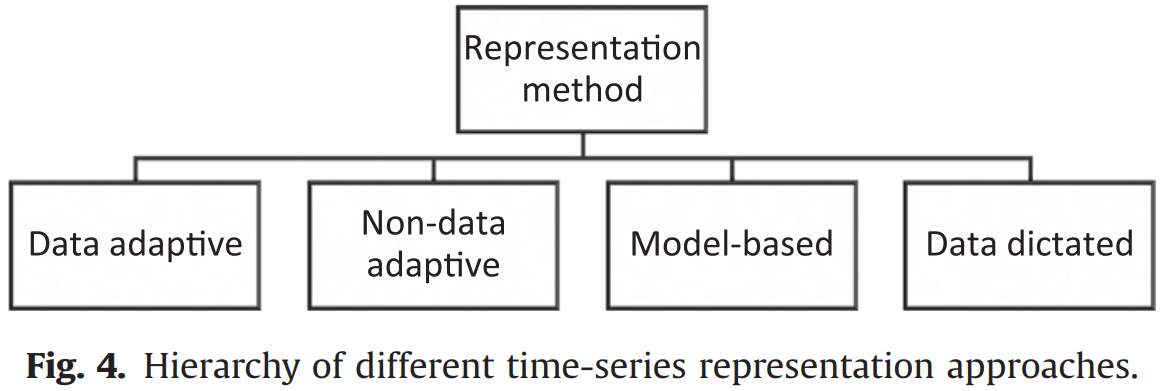
Dimensionality reduction is greatly important in clustering of time-series is:

1. Firstly, because it reduces memory requirements as all raw time-series cannot fit in the main memory [9,24].
2. Secondly, distance calculation among raw data is computationally expensive, and dimensionality reduction significantly speeds up clustering [9,24].
3. Finally, when measuring the distance between two raw time-series, highly unintuitive results may be garnered, because some distance measures are highly sensitive to some “distortions” in the data [3,83], and consequently, by using raw time-series, one may cluster time-series which are similar in noise instead of clustering them based on similarity in shape.

The potential to obtain a different type of cluster is the reason why choosing the appropriate approach for dimension reduction (feature extraction) and its ratio is a challenging task [26]. In fact, it is a trade-off between speed and quality and all efforts must be made to obtain a proper balance point between quality and execution time.

## Taxonomy of representations

There are generally four representation types [9,83,92,93]:



1. data adaptive:

* Performed on all time-series in data-sets and try to minimize the global reconstruction error [94] using arbitrary length (non-equal) segments.
* Data adaptive representations can better approximate each series, but the comparison of several time-series is more difficult.
* User can define the compression-ratio based on the application in hand.

1. non-data adaptive:

* Suitable for time-series with fix size (equal-length) segmentation,
* Comparison of representations of several time-series is straightforward.
* User can define the compression-ratio based on the application in hand.

1. Model-based:

* represent a time-series in a stochastic way such as Markov Models and Hidden Markov Model (HMM) [102–104], Statistical Models, Time-series Bitmaps [105], and Auto-Regressive Moving Average (ARMA) [106,107]
* user can define the compression-ratio based on the application in hand.

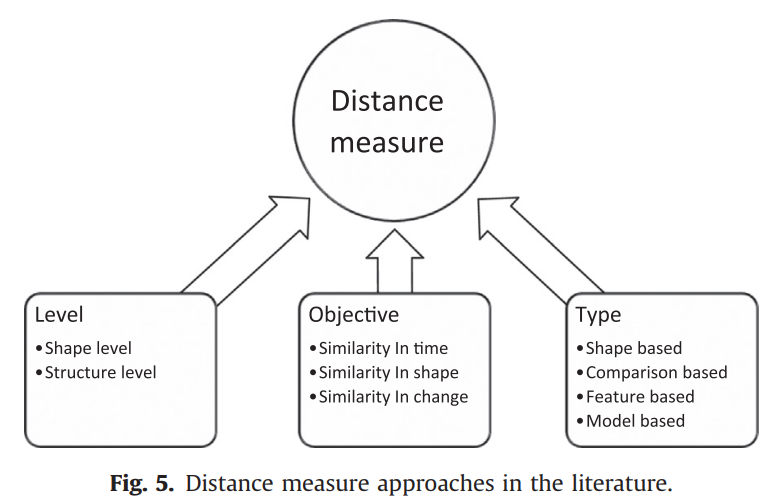
1. data dictated:

* the compression ratio is defined automatically based on raw time-series such as Clipped [83,92].

# Similarity/dissimilarity measures

The choice of a proper distance approach depends on:

* the characteristic of time-series,
* length of time-series,
* representation method,
* the objective of clustering time-series to a high extent.



## Level

Clustering approaches could be classified into two categories based on the length of time-series: “shape level” and “structure level”.

* The “shape level” is usually utilized to measure similarity in *short-length* time-series clustering such as expression profiles or individual heartbeats by comparing their local patterns,
* The “structure level” measures similarity which is based on global and high level structure, and it is used for long-length time-series data such as an hour’s worth of ECGs or yearly meteorological data.

### Similarity in time

Clustering of time-series that are correlated, (e.g., to cluster time-series of share price related to many companies to find which shares change together and how they are correlated) is categorized as *clustering based on similarity in time* [83,112].

Because this similarity is on each time step, correlation-based distances or Euclidean distance measure are proper for this objective.

However, because this kind of distance measuring is costly on raw time-series, the calculation is performed on transformed time-series, such as Fourier transforms, wavelets or Piecewise Aggregate Approximation (PAA).

Keogh and Kasetty [6], have done a comprehensive review on this matter.

### Similar in shape

The time of occurrence of patterns is not important to find similar time-series in shape. As a result, elastic methods [108,113] such as Dynamic time Warping (DTW) [114] is used for dissimilarity calculation.

Using this definition, clusters of time-series with similar patterns of change are constructed regardless of time points. For example, to cluster share price related to different companies which have a common pattern in their stock independent on its occurrence in time-series [112].

Similarity in time is an especial case of similarity in shape. A research has revealed that similarity in shape is superior to metrics based on similarity in time [115].

### Similarity in change (structural similarity)

In this approach, usually modelling methods such as Hidden Markov Models (HMM) [116] or an ARMA process [107,117] are utilized, and then similarity is measured on the parameters of fitted model to time-series. That is, clustering time-series with similar autocorrelation structure, e.g., clustering of shares which have a tendency to increase after a fall in share price in the next day [112].

This approach is proper for *long time-series*, not for modest or short time-series [21].

## Distance measures

Essentially, there are four types of distance measure in the literature.

Feature based similarity measure are proper for long time-series, such as Statistics, Coefficients

Model based similarity is proper for long time-series, such as HMM [116] and ARMA [107,117].

* Euclidean Distance (ED)
  + most commonly used distance measure, hard to beat
  + main limitations:
    - only for series of equal length
    - sensitive to time shifts
* Dynamic Time Warping (DTW) distance
  + shape based; overcomes Euclidean distance limitations
  + algorithm compares two series by creating a local cost matrix and traversing it to find the optimal warping path
  + needs/constraints:
    - choice of step pattern
    - choice of window that limits the area of the LCM (unknown a priori, so test for best size)
    - can handle series with unequal length (slanted band vs. Sakoe-Chiba window)
    - minimum alignment and divergence functions, Eq. 1.1 and 1.2
    - computationally expensive
  + DTW methods provide algorithms to optimally map a given time series onto all or part of another time series (Berndt and Clifford 1994). The remaining cumulative distance between the series after the alignment is a useful distance metric in time series data mining applications for tasks such as classification, clustering, and anomaly detection. Calculating a DTW alignment is computationally relatively expensive, and as a consequence DTW is often a bottleneck in time series data mining applications, but there are packages that implement faster search, such as package `rucrdtw` for R.
* Global alignment kernel (GAK) distance
  + similarity between time series by using kernels
  + uses a local similarity function (in Eq. 2)
  + again, get the best path at the lowest cost (soft-minimum of all alignment scores)

\[ DTW(x,y) = min\_{\pi \in \Lambda (n,m)} D\_{x,y}(\pi)\](1.1) \[ DTW(x,y) = \sum\_{i=1}^{|\pi|} \varphi(x\_{\pi\_1 (i)}, y\_{\pi\_2 (i)})\](1.2) \[ \kappa\_{GA}= \sum\_{\pi \in \Lambda (n,m)}\prod\_{i=1}^{|\pi|}\ K (x\_{\pi\_1 (i)}, y\_{\pi\_2 (i)}) \](2)

* alignment between two time series x and y of lengths n and m;

# Prototype

* Mean or median: the average of each time-point
  + poor choice (affects convergence)
  + only for series of equal length;
* Partition around medoids (PAM): medoid is a prototype of a cluster that is one of the time series itself
  + facilitates computation (“re-usable”);
* DTW barycenter averaging
* Shape extraction
* Fuzzy-based prototypes
* Local search: the best (see Decade Review)

# Algorithm

* Partitioning:
  + Random *k* [centroids](http://rstudio-pubs-static.s3.amazonaws.com/series_centroid.png) are initialized, distance to all data is determined and objects are assigned to each cluster (iterative);
  + Cons:
    - must pre-specify k;
    - stochastic, random start and may converge at a *local* optima;
  + Pros:
    - best for larger data sets.
* Hierarchical:
  + Agglomerative: start with all points as clusters, then join closest until all data are grouped;
  + Produces a dendrogram from where the best k clusters can be deduced (as compared to using CVI’s for partitioning methods)
  + Pros:
    - no needed pre-specification of k clusters
  + Cons:
    - Used for smaller data sets, such as 200 records
    - Need to specify the pairwise similarities (such as DTW) and the similarity measure between groups (such as linkage method)
      * linkage methods: i.e. single (closest pair), complete (furthest pair – generally preferred; allows more compact clusters), Diana, Wards;
* Fuzzy:
  + boundaries of clusters change with time
* Grid: ???
* Model:
  + ARMA family, Hidden Markov Models
* Density:
  + DBSCAN, OPTICS
* Multi-step:
  + clustering is used in a system as one of components
* Tadpole: ???

# Evaluation

The goal of evaluation of clustering solutions is to choose the best *k* clusters (undetermined *a priori*).

* Internal CVIs are measures of clusters purity:
  + Silhouette index (max)
  + Dunn index (max)
  + Calinski-Harabasz (CH) index (max)
  + COP index (min)
  + Davies-Bouldin index (min)
* External: compare clustering results with another source, typically by human expertise. There are many different indices, such as
  + Rand index,
  + F-measure,
  + etc.
* Choose the solution based on the highest index majority vote.