Time series clustering

Clustering is a classic unsupervised machine learning methods. The goal is to create clusters with elements of the datasets with similar behavior. The goal is then to maximize the similarities inside a cluster and to minimize it for elements outside this cluster. Here, we are particularly interested by clustering of one-dimensional time series. It is a classic datatype that exists in various fields (financial, industrial, weather,...).

The main objective of this work is to determine the efficiency of UMAP as a pre-processing step for clustering algorithm.

UMAP : Uniform Manifold Approximation and Projection

UMAP [2] is a recent reduction of dimension method. It was made mainly for visualisation of high dimensional data.

From a dataset, it starts by construct a graph $G$ where each vertex is an element of the dataset. Then, the edges are degrees of membership. For an element $y_i$, the $m$-th closest neighbor have an edge $e \in [0,1]$ with respect to the distance. Then, a Laplacian eigenmaps is used to create a graph $G'$ of smaller dimension. From the cross-entropy between $G$ and $G'$, attractive and repulsive force are determined and a forced directed graph layout is computed. So, each vertex acts as a physical point under those two forces until a physical equilibrium is obtained. The cross-entropy is now minimized between the two graphs $G$ and $G'$.

So, there is a few parameters for UMAP (as the number of neighbour). We decided to follow the advice on their website, to keep generic parameters for all databases.

As a reduction of dimension, UMAP can be used beforehand to clarify the data for clustering algorithm. However, there is still some theoretical controversies with UMAP. Indeed, by keeping the local structure, it can create pseudo groups that disturb clustering algorithm. In practice, we propose to apply UMAP with real data.

Clustering algorithms

To validate the efficiency of UMAP, we use three clustering algorithms. They are of three distinct types.

HDSCAN [1] is a prolongation of DBSCAN. For both K-means and Hierarchical, we need to know beforehand the number of clusters. To do so, we use the silhouette score. It calculates a score that gives the quality of the clustering. Small distance intra-clusters and high distance inter-clusters gives a small score. So, we apply those two clustering algorithms for multiple value of $\epsilon$ and kept the number of clusters that minimizes the silhouette score.

For our benchmark, we used the databases available at UCR Time Series Classification Archive ucr.cs.ucr.edu/ucrdata/time-series-data/. 85 databases with labelled time series of same length were available. So, we applied the three clustering algorithms on those datasets with and without UMAP beforehand. To compare the clustering results with the true labels, we use the v-measure score [3]. It is a similarity measure between two clustering results based on entropy. In the figure below, a summary of the construction of the benchmark is proposed.

We used the euclidean distance on all databases. So, we had 6 clustering results with this distance for 85 databases.

We add also another distance to complete the benchmark. For a given database, the time series are transformed in a trajectory matrix through a delay coordinate embedding (see figure below).

Then, with a reduction of dimension and orthogonalization, the time series are seen as element of the Stiefel manifold:\n\[ V_{n,p} = \{ A \in \mathbb{R}^{n \times p} : A^{t}A = I \} \]. The measure of similarity between two time series is then the geodesic (Principal angle) on this manifold. It is sufficient to apply the HDSCAN and Hierarchical algorithms. However, for K-means, we need also the definition of a mean with respect to the geodesic : the Karcher mean.

Since the computation of the Karcher mean can be a bit long, we limitate this part of the benchmark on 6 databases of the UCR Time Series Classification Archive.

Results

We computed the average v-measure score obtained across the databases for all methods. The graph below represent the results on the 85 databases with the Euclidean distance.

Discussion

Those results validate the use of UMAP for clustering even with the theoretical limits. Indeed, for all algorithms and distances, UMAP offers an improvement. This is particularly true for HDSCAN. It was expected since those two algorithms (UMAP and HDSCANN) are based on the data density. UMAP+HDSCAN is also the method with the best mean compared to the other clustering algorithms (with or without UMAP).

Moreover, we use generic parameters for UMAP. Indeed, it was not possible to optimize those parameters for each databases. So, there is still room for improvement according to the user’s knowledge of the database.

Acknowledgements

We thank all curators and administrators of the UCR archive without which this work would not have been possible.

References

