Manifolder - Topology-based Feature Extraction and Event Localization in Unsupervised Clustering of Multivariate Timeseries

Babel Publishing (list authors later)[1] and et al.[2]

[1]Alphatrai, San Diego, CA 92121, USA

[2]yy

Abstract

Unsupervised learning for multivariate timeseries remains a challenge. One approach is to treat each point in the time sequence independently and then apply clustering techniques. The problem with this approach is that important temporal dependencies are lost. Various methods have been proposed to extract temporal features. Here we extend a topologically based approach to scenarios where the timeseries is in form of snippets rather than a continuous stream. We examine the viability of this approach through several different timeseries datasets where the labels are known (NASA FTE data, lets pick 3 or more from this list http://www.timeseriesclassification.com/dataset.php). We draw a parallel to quantum physics and introduce a state transition diagnostic that quantifies the transition frequency/probability as well as temporal persistence of each cluster. We also discussed ways to identify the optimal number of clusters. In case of FTE data, we know the signature and the key variables that are used by domain experts to identify the events. This data is particularly challenging since there are many other events present in the data but the particular ones of interest are the FTE events. This provides a good test of the technique since we can compare whether the clustering can identify the events and their localization within the timeseries snippets. We also compare the technique against another method for feature extraction (srinjoy had mentioned one or two other ones). Finally, we experiment with several distance measures in clustering (multivariate DTW, any others we want to include?). The code, data source and documentation are provided.

http://www.timeseriesclassification.com/dataset.php

I. INTRODUCTION

The outline of the paper is the following: In Section II we provide an overview of the clustering approaches for timeseries as well as the different distance measures. Section III describes the manifolder and our tweaks to it, along with our choice of clustering technique and the set of distance measures used in our experiments. The choice and description of the datasets are described in Section IV. Section V discusses the application to FTEs. Section VI discusses the efficacy of the approach to localizing the event within a snippet. This technique can be used even in cases where the class label is known for the entire snippet but the beginning and end of the events are not known. (we need to think about how to use it for such cases. e.g., do we just do clustering on the class 1 ...?). Section VII describes the test results on 5 data sets. Section **??** describes the effect of different distance measures on the results. Section VII compares the efficacy of manifolder versus .... Summary is given in Section VIII.

II. LITERATURE OVERVIEW

III. OUR APPROACH

Address the meaningless issue raised by keogh (his point is abt subsequent clustering which is not what we do here.). their point is that if you start with a continuous timeseries and then break it into snippets using sliding window, you get cluster results that are random. they use the cylinder-bell-funnel data, then concatenate the snippets and then run STS clustering with k-means. they then use this data to show that STS is meaningless. we can do the same using manifolder to see if we get something different. we could also do the same on our nasa data. we could do that either by concatenating only label 1 cases or include label 0 and 1. they propose motif-based clustering as a possible fix which first reduces the data to those with motifs prior to clustering. drawing a parallel to that - i.e., manifolder has analogy to a motif, may be a way of explaining our result. (their work seems to be for univariate).

Here are alex's ideas on this: "1) we're looking for an extremal event, and don't care about the other cluster centers. 2) we're comparing the local stationary distributions of the time series, so two signals with different variances will end up in different places in our

embedding (and clustering) 3) using histograms eliminates the smooth transitions that exist in continuous time series, and thus the means here won't present a sine curve.

I also think this does suggest a potential change to the algorithm. We are looking for a motif of sorts (though there is variability in that motif). So rather than doing kmeans clustering on all segments, we could find points in space that correspond to the motif (using the fact that we have supervision) and then build a ball around those points in the embedding space. Then if new points fall into those balls, they are also a motif."

A. Empirical Intrinsic Geometry Embedding

We apply the dynamical systems diffusion map technique developed in [] to embed small windows from all of the snippets into a single, low dimensional Euclidean space. We describe the embedding algorithm here for ease of reading, see [] for a full description.

Given M D-dimensional time series snippets $\{x_i(t)\}_{i=1}^M$, we construct windows of size H across all snippets $\left[x_i(t_j - H + 1), \ ..., \ x_i(t_j)\right]_{j=1}^{N_i}$ where N_i is the number of windows available in the ith snippet. This creates a data cube of size $N_i \times H \times D$. We concatenate all of these data cubes along the 1st dimension to create a data cube X of size $N \times H \times D$, where $N = \sum_{i=1}^M N_i$.

X is transformed via histograms along each of the D dimensions, binning the H points from each window into k equally space bins ranging from the minimum value to the maximum value across all data in that dimension. This results in a matrix $\widetilde{X}$ of D histograms per data point, making the matrix of size $N \times kD$.

Add description of covariance matrices

Add description of Mahalanobis distance and kernel, including measuring to reference points

Finally, we take d left singular vectors $\Phi(x_i) = \left[\lambda_j \phi_j(x_i)\right]_{j=1}^d$ of the matrix A to characterize the embedding of the data points. A benefit of these vectors is that the diffusion distance between x_{i_1} and x_{i_2} is characterized by the Euclidean distance $\|\Phi(x_{i_1}) - \Phi(x_{i_2})\|_2$.

B. Snippet Level Classification

Add description of training/testing split at snippet level

3

Add description of nearest neighbor classification of windows based off localized features.
Mention Keogg new paper that snippet level labels alone is insufficient
Add description of thresholding to determine snippet level classification

IV. DATA DESCRIPTION

NASA data + any other set? Also, in anticipation of comparison with supervised in later section, break the data into "training", "validation" and test. There are 5440 data points per snippet and unlike THEMIS data, we do have beginning and end of events tagged. Naveen downsampled MMS to the same number (1440) as THEMIS. This allows combining THEMIS and MMS data. Just need to make sure that downsampling of MMS doesn't wipe out some of the shorter duration FTEs in MMS. We could use THEMIS as "training" for clustering and then test on MMS or some combination thereof. The fact that we have the beginning and end of evengs for MMS will help us determine overlap of ROIs between clustering and ground truth.

V. CLUSTERING RESULT FROM MANIFOLDER ON FTES

A. univariate

univariate (Bn) vs multivariate - input/variable importance

B. pointwise

comparison to pointwise clustering

C. robustness

robustness of clustering - how stable are the clustering results to window size/resolution

D. comparison with our previous unsupervised approach

i had a paper with naveen on univariate. compare it with that using Bn.

E. Unsupervised Into Supervised

compare the AUC and Average Precision (AP) (ROC and PR curve) with LSTM and the attention-based algo (naveen and homa). It would also be interesting to use the cluster assignments to train a neural net with variational dense layers to get uncertainty measures. Does the high uncertainty cases flag cases that were not properly classified by the clustering? This could provide a feedback loop and a way to combine unsupervised with supervised. Also check the uncertainty on the training set as well as the validation/test set.

F. Transition Matrices

examine whether there is any perceivable pattern in transition matrices, i.e., are events preferentially preceded by certain clusters?

VI. EVENT LOCALIZATION OF FTES

It would be interesting to train a FRCNN based on localization of the events by the clustering and check its efficacy in the valid/test. We could compare the performance of FRCNN for MMS where we know the beginning and end with that trained based on unsupervised labels.

VII. TEST ON OTHER DATA SETS

VIII. SUMMARY

Acknowledgments