Time Series Discretization

June 7, 2018

1 Introduction

Given a dataset of m time series $T_{i,j} = [T_1, T_2, .., T_m]$ where each row i is an univariate time series described by an horizontal vector of n observations $T_{1,n}^{(i)} = \left[t_1^{(i)}, t_2^{(i)}, .., t_n^{(i)}\right]$, the main goal of time series discretization known also as temporal discretization is to transform the original time serie values into categorical values through a discretization scheme $d_l = [d_1, d_2, .., d_l]$ where $2 \leq l \leq n$. To discretize a time series dataset, there are two approaches in the old and recent literature. On the one hand attribute based methods discretize the whole dataset without considering temporal order of data. On the other hand, instance based methods where designed especially for temporal data, it descretizes each instance using both time axis and vector values.

$$T_{n,m} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ t_{21} & t_{22} & \dots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \dots & t_{mn} \end{bmatrix} \tag{1}$$

2 Discretization method

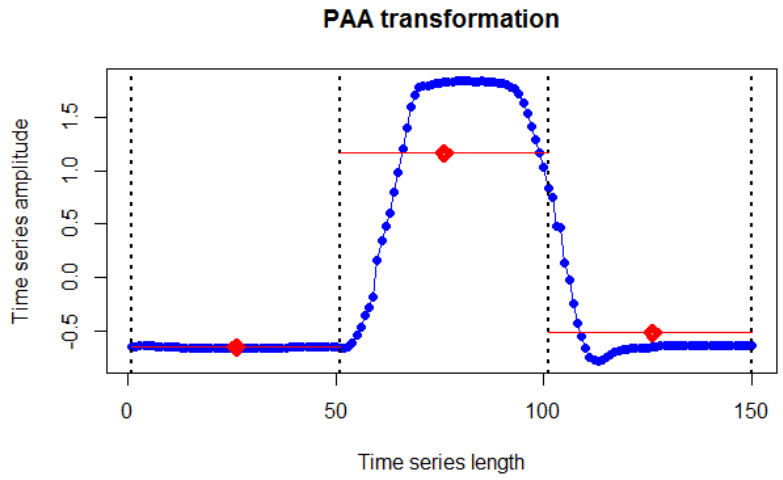
As mentioned before, classical discretization methods were applied on time series discretization problem to split attribute values and produce a symbolic representation of the time series without reducing its size. Our idea is to break with the requierement of applying discretization on the feature values, and to use classical discretization algorithms to transform instances as univariate time series. The proposed discretization method transforms a time series TS with length n to a reduced symbolic representation $\overline{TS}$ with length w ($2 \leq w \leq n$). We propose a new approach defined as a discretization method that satisfy the additional temporal property of time series. Both classical discretization methods and PAA are combined in this approach to achieve better discretization performances. The first step in our strategy is to split the time series length to construct a set of word using PAA. Then, to split data amplitude, in the second step we apply classical discretization methods to select cut-points (alphabet).

2.1 Constructing a set of word via PAA

To construct a set of word with length w, the PAA algorithm apply the same process as the well known discretization algorithm EWD. It splits the time series length into w equal size intervals. Starting by defining the intervals width as in equation 2, then it constructs the cut points list $C_i = [c_1, c_2, ..., c_{w-1}]$ using equation 3.

$$width = \frac{time - series - length}{w} \tag{2}$$

$$c_i = 1 + (i * width) \tag{3}$$



Using both equation 2 and 3, PAA algorithm constructs a list of cut points that split the time series length into a reduced equal sized intervals. Each interval I_j that belong to the list of intervals is labeled with the arithmetic mean of data falling within this interval. The representation of the intervals list can be visualized in figure 1.

3 Time Series Results

Table 1: Classification accuracy (Conditional Inference Trees).

#	**Dataset**	**SAX**	**w**	**NEW**	**k**
1	GunPoint	0.76	12	0.9139	23
2	CBF	0.6311	9	0.661	8
3	Trace	0.75	19	0.98	6
4	FaceFour	0.159	2	0.159	2
5	Lighting2	0.754	9	0.77	18
6	Lighting7	0.52	9	0.48	6
7	ECG200	0.82	17	0.81	6
8	FISH	0.46	17	0.3885	20
9	Plane	0.876	17	0.790	7
10	Car	0.433	9	0.5	14
11	Beef	0.33	7	0.33	8
12	Coffee	0.5357	2	0.5357	2
13	OliveOil	0.66	7	0.66	7
14	ArrowHead	0.537	5	0.531	7
15	BeetleFly	0.5	2	0.5	2
16	BirdChicken	0.5	2	0.5	2
17	Ham	0.533	20	0.514	2
18	Herring	0.5937	2	0.5937	2
19	ToeSegmentation1	0.5263	2	0.5263	2
20	ShapeletSim	0.5	2	0.5	2
21	Wine	0.5	2	0.5	2
22	Meat	0.66	14	0.85	19
23	Worms	0.419	2	0.43	8
24	WormsTwoClass	0.58	2	0.58	2
25	Mean	0.564075	7.958333	0.5834625	7.375