Finding persisting states for knowledge discovery in time series

Fabian Mörchen and Alfred Ultsch

Data Bionics Research Group,
Philipps-University Marburg, 35032 Marburg, Germany

Abstract. Knowledge Discovery in time series usually requires symbolic time series. Many discretization methods that convert numeric time series to symbolic time series ignore the temporal order of values. This often leads to symbols that do not correspond to states of the process generating the time series. We propose a new method for meaningful unsupervised discretization of numeric time series called "Persist", based on the Kullback-Leibler divergence between the marginal and the self-transition probability distributions of the discretization symbols. In evaluations with artificial and real life data it clearly outperforms existing methods.

1 Introduction

Many time series data mining algorithms work on symbolic time series. For numeric time series they usually perform unsupervised discretization of the values as a preprocessing step. For the discovery of knowledge that is interpretable and useful to the expert, it is of great importance that the resulting interval boundaries are meaningful within the domain. If the time series is produced by an underlying process with recurring persisting states, intervals in the value dimension should describe these states. The most commonly used discretization methods are equal width and equal frequency histograms. Both histogram methods potentially place cuts in high density regions of the observed marginal probability distribution of values. This is a disadvantage, if discretization is performed not merely for quantization and speedup of processing, but rather for gaining insight into the process generating the data. The same applies to other methods, e.g. setting cuts based on location and dispersion measures. While static data sets offer no information other than the actual values themselves, time series contain valuable temporal structure that is not used by the methods described above. We propose a new method for meaningful unsupervised discretization of univariate time series by taking the temporal order of values into account. The discretization is performed optimizing the persistence of the resulting states.

In Section 2 we give a brief overview of related methods. The new discretization algorithm is described in Section 3. The effectiveness of our approach is demonstrated in Section 4. Results and future work are discussed in Section 5.

2 Related work and motivation

A recent review of discretization methods for data mining is given in Liu et al. (2002). The only unsupervised methods mentioned are equal width and equal frequency histograms. With unsupervised discretization no class labels are available, thus there can be no optimization w.r.t. classification accuracy. But for time series data in particular there is rarely some sort of labeling available for the time points.

The choice of parameters for the Symbolic Approximation (SAX) (Lin et al. (2003)) (similar to equal frequency histograms) has been analyzed in the context of temporal rule mining in (Hetland and Saetrom (2003)). The authors suggest to use the model with the best performance on the validation data. But using support and confidence of rules as a quality score is ignoring a simple fact. Rules are typically created to gain a deeper understanding of the data and the patterns therein. Arguably, rules with high support and confidence are less likely to be spurious results. But they will not be useful if the interval boundaries of the discretization are not meaningful to the domain expert.

The related task of time series segmentation (e.g. Keogh (2004)) is beyond the scope of this paper. Segmentation does not lead to recurring state labels per se. Instead of dividing the value dimension in intervals, the time dimension is segmented to produce line or curve segments homogeneous according to some quality measure. Postprocessing the segments can lead to recurring labels like *increasing* for segments with similar positive slopes.

3 Persistence in time series

We propose a new quality score for meaningful unsupervised discretization of time series by taking the temporal information into account and searching for persistence. We argue, that one discretization is better than another if the resulting states show more persisting behavior. We expect many knowledge discovery approaches to profit from more meaningful symbols that incorporate the temporal structure of the time series, e.g. rule discovery in univariate (e.g. Hetland and Saetrom (2003), Rodriguez et al. (2000)) and multivariate (e.g. Guimaraes and Ultsch (1999), Höppner (2002), Harms and Deogun (2004), Mörchen and Ultsch (2004)) time series, or anomaly detection (e.g. Keogh et al. (2002)).

Let $S = \{S_1, ..., S_k\}$ be the set of possible symbols and $s = \{s_i | s_i \in S \; i = 1..n\}$ be a symbolic time series of length n. Let $P(S_j)$ be the marginal probability of the symbol S_j. The $k \times k$ matrix of transition probabilities is given by $A(j, m) = P(s_i = S_j | s_{i-1} = S_m)$. The self-transition probabilities are the values on the main diagonal of A.

If there is no temporal structure in the time series, the symbols can be interpreted as independent observations of a random variable according to

the marginal distribution of symbols. The probability of observing each symbol is independent from the previous symbol, i.e. $P(s_i = S_j|s_{i-1}) = P(S_j)$. The transition probabilities are $A(j,m) = P(S_j)$. The most simple temporal structure is a first order Markov model (Rabiner (1989)). Each state depends only on the previous state, i.e. $P(s_i = S_j|s_{i-1}, ...s_{i-m}) = P(S_j|s_{i-1})$. Persistence can be measured by comparing these two models. If there is no temporal structure, the transition probabilities of the Markov model should be close to the marginal probabilities. If the states show persisting behavior, however, the self-transition probabilities will be higher than the marginal probabilities. If a process is less likely to stay in a certain state, the particular transition probability will be lower than the corresponding marginal value.

A well known measure for comparing two probability distributions is the Kullback-Leibler divergence (Kullback and Leibler (1951)). For two discrete probability distributions $P = \{p_1, ..., p_k\}$ and $Q = \{q_1, ..., q_k\}$ of k symbols the directed (KL) and symmetric (SKL) versions are given in Equation 1.

$$KL(P,Q) = \sum_{i=1}^{k} p_i log\left(\frac{p_i}{q_i}\right) \quad SKL(P,Q) = \frac{1}{2}\left(KL(P,Q) + KL(Q,P)\right) \quad (1)$$

For binary random variables we define the shortcut notation in Equation 2.

$$SKL(p,q) := SKL(\{p, 1-p\}, \{q, 1-q\}) \; \forall p, q \in]0, 1] \quad (2)$$

The *persistence* score of state j is defined in Equation 3 as the product of the symmetric Kullback-Leibler divergence of the transition and marginal probability distribution for self vs. non-self with an indicator variable. The indicator determines the sign of the score. States with self-transition probabilities higher than the marginal will obtain positive values and states with low self-transition probabilities inhibit negative values. The score is zero if and only if the probability distributions are equal.

$$Persistence(S_j) = sgn(A(j,j) - P(S_j))SKL(A(j,j), P(S_j)) \quad (3)$$

A summary score for all states can be obtained as the mean of the values per state. This captures the notion of mean persistence, i.e. all or most states need to have high persistence for achieving high persistence scores. The calculation of the persistence scores is straight forward. Maximum likelihood estimates of all involved probabilities can easily be obtained by counting the number of symbols for each state for the $P(S_j)$ and the numbers of each possible state pair for A. The persistence score is used to guide the selection of bins in the *Persist* algorithm. The first step is to obtain a set of candidate bin boundaries from the data, obtained e.g. by equal frequency binning with a large number of bins. In each iteration of the algorithm all available candidate cuts are individually added to the current set of cuts and the persistence score is calculated. The cut achieving the highest persistence is chosen. This is repeated until the desired number of bins is obtained. The time complexity is $O(n)$.

4 Experiments

We evaluated the performance of the Persist algorithm by extensive experiments using artificial data with known states and some real data where the true states were rather obvious. We compared the novel algorithm with the following eight methods: **EQF** (equal frequency histograms), **SAX** (equal frequency histograms of normal distribution with the same mean and standard deviation as the data) [1] **EQW** (equal width histograms), **M±S** (mean ± standard deviation of data), **M±A** (median ± adjusted median absolute deviation (AMAD) of data), **KM** (k-Means with uniform initialization and Manhattan distance), **GMM** (Gaussian mixture model), **HMM** (Hidden Markov Model). HMM is the only competing method using the temporal information of the time series. It is not quite comparable to the other methods, however, because it does not return a set of bins. The state sequence is directly created and the model is harder to interpret.

Artificial data: We generated artificial data using a specified number of states and Gaussian distributions per state. We generated 1000 time series of length 1000 for $k = 2, ..., 7$ states. For each time series 10 additional noisy versions were created by adding 1% to 10% outliers uniformly drawn from the interval determined by the mean ± the range of the original time series. An example for 4 states and 5% outliers is shown in Figure 3(a). The horizontal lines indicate the true means of the 4 states. The large spikes are caused by the outliers. Figure 3(b) shows the Pareto Density Estimation (PDE) (Ultsch (2003)) of the marginal empirical probability distribution.

We applied all discretization methods using the known number of states. We measured the accuracy of the discretization by comparing the obtained state sequence with the true state sequence used for generating the data. The median accuracies and the deviations (AMAD) for $k = 5$ states and three levels of outlier contamination are listed in Table 1. The Persist algorithm always has a higher median accuracy than any static method with large distances to the second best. The deviation is also much smaller than for the other methods, indicating high consistency. Even with 10% outliers, the performance of the new algorithm is still better than for any static method applied to the same data without outliers! Compared to the only other temporal method, HMM, the performance of Persist is slightly worse for 0% outliers. But with larger levels of outlier contamination, the HMM results degrade rapidly, even below the results from several static methods.

The results for other values of k were similar. The absolute differences in accuracy were smaller for $k = 2, 3, 4$ and even larger for $k = 6, 7$. The performance of HMM degraded later w.r.t. outlier contamination for fewer states and earlier for more states. Figure 1 plots the median accuracies for 3 states, all methods, and all outlier levels. Again, the Persist method is always the best except for HMM at low outlier levels.

[1] This is a special case of SAX with window size 1 and no numerosity reduction.

Table 1. Median accuracy for 5 states

Outliers	0%	5%	10%
EQF	0.74 ± 0.08	0.71 ± 0.08	0.69 ± 0.07
SAX	0.74 ± 0.09	0.74 ± 0.08	0.72 ± 0.08
EQW	0.67 ± 0.16	0.33 ± 0.09	0.32 ± 0.08
M±S	0.56 ± 0.11	0.48 ± 0.10	0.43 ± 0.09
M±A	0.51 ± 0.16	0.48 ± 0.15	0.45 ± 0.13
KM	0.71 ± 0.21	0.66 ± 0.22	0.61 ± 0.24
GMM	0.79 ± 0.18	0.27 ± 0.12	0.24 ± 0.11
HMM	0.94 ± 0.08	0.52 ± 0.34	0.44 ± 0.29
Persist	*0.90 ± 0.03*	*0.86 ± 0.03*	*0.83 ± 0.03*

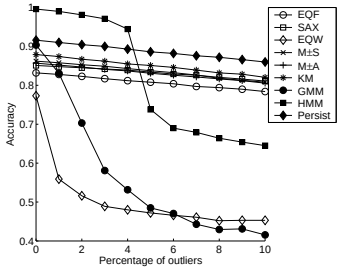


Fig. 1. Median accuracy 3 states

States	Outliers										
	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
2	−	−	−	−	−	−	−	−	−	−	○
3	−	−	−	−	○	+	+	+	+	+	+
4	−	−	○	+	+	+	+	+	+	+	+
5	○	+	+	+	+	+	+	+	+	+	+
6	+	+	+	+	+	+	+	+	+	+	+
7	+	+	+	+	+	+	+	+	+	+	+

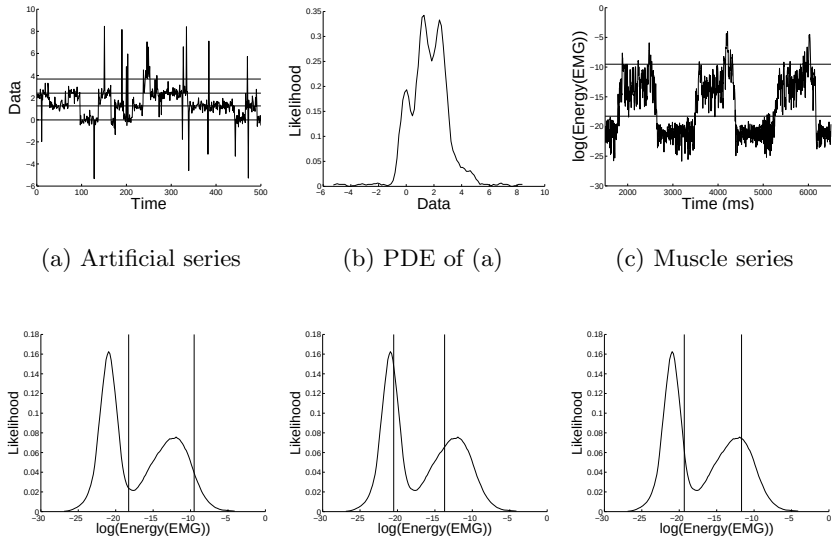
Fig. 2. Test decisions of Persist vs. HMM

In order to check the results for statistical significance, we tested the hypothesis that the accuracy of the Persist is better than the accuracy of the competing algorithms with the rank sum test. The test was performed for all k and all noise levels. For the competing static methods *all* p-values were smaller than 0.001, clearly indicating superior performance that can be attributed to the incorporation of temporal information. Compared to HMM, the results are significantly better for the larger amounts of outliers and worse for no of few outliers. The more states are present, the less robust HMM tends to be. Table 2 shows the result of the statistical tests between Persist and HMM. A plus indicates Persist to be better than HMM, for a minus the accuracy is significantly lower, circles are placed where the p-values were larger than 0.01.

In summary, the Persist algorithm was able to recover the original state sequence with significantly higher accuracy and more consistency than all competing static methods. The temporal HMM method is slightly better than Persist for no or few outliers, but much worse for more complicated and realistic settings with more states and outliers.

Real data: For real life data the states of the underlying process are typically unknown. Otherwise a discretization into recurring states wouldn't be necessary. We explored the behavior of the Persist algorithm in comparison with the other methods on two datasets that clearly show several states. The

muscle activation of a professional inline speed skater (Mörchen et al. (2005)) is expected to switch mainly between being active and relaxed. Five seconds of the data are shown in Figure 3(c). Consulting an expert we chose $k = 3$ states. The resulting bin boundaries of three selected methods are shown in Figures 3(d)- 3(f) as vertical lines on top of a probability density estimation plot. All methods (including the other methods not shown) except Persist place cuts in high density regions. EQF sets the first cut very close to the peak corresponding to the state of low muscle activation. This will result in a large amount of transitions between the first two states. EQW does the same for the second peak in the density, corresponding to high muscle activation. Persist is the only method that places a cut to the right of the second peak. This results in a state for very high activation. The validity of this state can also be seen from Figure 3(c), where the horizontal lines correspond to the bins selected by Persist. The very high values are not randomly scattered during the interval of high activation but rather concentrate toward the end of each activation phase. This interesting temporal structure is not visible from the density plot, is not discovered by the other methods, and was validated by the expert as the push off with the foot.



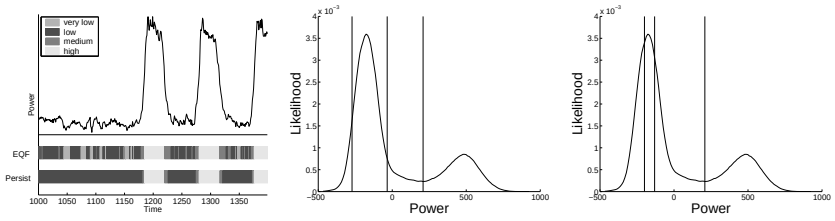
(a) Artificial series (b) PDE of (a) (c) Muscle series

(d) Persist bins (e) EQF bins (f) EQW bins

Fig. 3. Artificial data and Muscle data with results

The Power data describes the power consumption of a research center over a year (van Wijk (1999), Keogh (2002)). The data was de-trended to

remove seasonal effects, half a week is shown in Figure 4(a). Persist with four states (Figure 4(b)) corresponded to (1) very low power usage, (2) usually low usage at nighttime, (3) rather low usage at daytime, and (4) usual daytime consumption. In contrast, the EQF method places a very narrow bin around the high density peak for nighttime consumption (Figure 4(c)). There will be frequent short interruptions of this state with symbols from the two neighboring states. This is demonstrated in Figure 4(a). Below the original data the state sequences created by EQF (top) and Persist (bottom) are shown as shaded rectangles. While the high states are almost identical, Persist creates much 'cleaner' low states with higher persistence.



(a) Time series with (b) Persist bins (c) EQF bins

Fig. 4. Power data

5 Discussion

The proposed quality score and algorithm for detecting persisting states has been shown to outperform existing methods on artificial data. In the Muscle data a state of very high activity was detected, that is neglected by the other methods. In the Power data less noisy states were found. The method is simple, exact, and easy to implement. The only competing method, HMM, is far more complex. The EM algorithm needs a good initialization, is sensitive to noise, and only converges to a local maximum of the likelihood. HMM models are also harder to interpret than the result of binning methods like Persist. Using each time point or a small window for discretization will usually produce consecutive stretches of the same symbol. In Daw et al. (2003) the authors state that *"from the standpoint of observing meaningful patterns, high frequencies of symbol repetition are not very useful and usually indicate oversampling of the original data"*. But interesting temporal phenomena do not necessarily occur at the same time scale. Trying to avoid this so called oversampling would mean to enlarge the window size, possibly destroying short temporal phenomena in some places. We think that with smooth time series

it is better to keep the high temporal resolution and search for persisting states. The resulting labeled interval sequences that can be used to detect higher level patterns (e.g. Höppner (2002), Mörchen and Ultsch (2004)).

References

DAW, C.S., FINNEY, C.E.A., and TRACY, E.R. (2003): A review of symbolic analysis of experimental data. *Review of Scientific Instruments*, 74:0 916–930.

GUIMARAES, G. and ULTSCH, A. (1999): A method for temporal knowledge conversion In *Proc. 3rd Int. Symp. Intelligent Data Analysis*, 369–380.

HARMS, S. K. and DEOGUN, J. (2004): Sequential association rule mining with time lags. *Journal of Intelligent Information Systems (JIIS)*, 22:1, 7–22.

HETLAND, M.L. and SAETROM, P. (2003): The role of discretization parameters in sequence rule evolution. In *Proc. 7th Int. KES Conf.*, 518–525.

HÖPPNER, F. (2002): Learning dependencies in multivariate time series. *Proc. ECAI Workshop, Lyon, France*, 25–31.

KEOGH, E. (2002): The UCR Time Series Data Mining Archive http://www.cs.ucr.edu/~eamonn/TSDMA/index.html

KEOGH, E., LONARDI, S., and CHIU, B. (2002): Finding Surprising Patterns in a Time Series Database in Linear Time and Space In *Proc. 8th ACM SIGKDD*, 550–556.

KEOGH, E., CHU, S., HART, D., and PAZZANI, M. (2004): Segmenting time series: A survey and novel approach. *Data Mining in Time Series Databases*, World Scientific, 1–22.

KULLBACK, S. and LEIBLER, R.A. (1951): On information and sufficiency *Annals of Mathematical Statistics*, 22, 79–86.

LIN, J., KEOGH, E., LONARDI, S., and CHIU, B. (2003): A symbolic representation of time series, with implications for streaming algorithms. In *Proc. 8th ACM SIGMOD, DMKD workshop*, 2–11.

LIU, H., HUSSAIN, F., TAN, C.L., and DASH, M. (2002): Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery*, 4:6, 393–423.

MÖRCHEN, F. and ULTSCH, A. (2004): Discovering Temporal Knowlegde in Multivariate Time Series In *Proc. GfKl, Dortmund, Germany*, 272–279.

MÖRCHEN, F., ULTSCH, A., and HOOS, O. (2005): Extracting interpretable muscle activation patterns with time series knowledge mining. *Intl. Journal of Knowledge-Based & Intelligent Engineering Systems* (to appear).

RODRIGUEZ, J.J., ALSONSO, C.J., and BOSTRÖM, H. (2000): Learning First Order Logic Time Series Classifiers In *Proc. 10th Intl. Conf. on Inductive Logic Programming*, 260–275.

RABINER, L. R. (1989): A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. of IEEE*, 77(2):0 257–286.

ULTSCH, A. (2003): Pareto Density Estimation: Probability Density Estimation for Knowledge Discovery. In *Proc. GfKl, Cottbus, Germany*, 91–102.

VAN WIJK, J. J., VAN SELOW, E. R. (1999): Cluster and Calendar Based Visualization of Time Series Data. In *Proc. INFOVIS*, 4-9.