A Bayesian Hybrid Approach to Unsupervised Time Series Discretization

Yoshitaka Kameya*, Gabriel Synnaeve†, Andrei Doncescu‡, Katsumi Inoue§, Taisuke Sato*
* *Tokyo Institute of Technology, Japan, {kameya,sato}@mi.cs.titech.ac.jp*
† *Grenoble University, France, gabriel.synnaeve@gmail.com*
‡ *LAAS-CNRS, France, andrei.doncescu@laas.fr*
§ *National Institute of Informatics, Japan, ki@nii.ac.jp*

***Abstract*—Discretization is a key preprocessing step in knowledge discovery to make raw time series data applicable to symbolic data mining algorithms. To improve the comprehensibility of the mined results, or to help the induction step of the mining algorithms, in discretization, it is natural to prefer having discrete levels which can be mapped into intuitive symbols. In this paper, we aim to make smoothing of the data points along with the time axis, and make binning or clustering at the measurement axis. In particular, we propose a hybrid discretization method based on variational Bayes, in which the output of one discretization method is smoothly exploited as hyperparameters of another probabilistic discretization model such as a continuous hidden Markov model. The experiments with artificial and real datasets exhibit the usefulness of this hybrid approach.**

***Keywords*-discretization; continuous hidden Markov models; variational Bayes**

I. INTRODUCTION

In a knowledge discovery process, the data are often obtained as time series of real-valued measurements by experiments or from sensors. Discretization [1], [2] is then a key preprocessing step to make these time series applicable to symbolic data mining algorithms, e.g. frequent pattern mining or inductive logic programming. In discretization, to improve the comprehensibility of the mined results, or to help the induction step of the mining algorithms, it is natural to prefer having discrete levels which can be mapped into intuitive symbols like "high," "medium," or "low." These symbols can also be basic elements in qualitative/logical reasoning [3], [4].

One frequent problem in time series discretization is that the data just include raw measurements and have no extra clue to know the meaningfulness of a discretization. In this case, discretization needs to be performed in an unsupervised fashion, which has been less explored [2], and such unsupervised methods would inherently rely on some assumption or model behind the data. We may see that discretization of a time series is a segmentation or clustering process in a two-dimensional (measurement-time) space, preserving the temporal behavior of the time series. In this paper, we aim to make smoothing of the data points along with the time axis, and make binning or clustering at the measurement axis. This strategy basically works, but as we will see later, it is not always easy to find a desired discretization only with a single underlying segmentation/clustering model.

From this background, hybrid approaches are practically attractive where we effectively combine a couple of promising heterogeneous methods for smoothing, binning or clustering. In this paper, we propose a hybrid discretization method based on variational Bayes [5], [6], in which the output of one discretization method is smoothly exploited as hyperparameters (as "prior knowledge") of another probabilistic discretization model. Also variational Bayes is suitable by itself for unsupervised discretization in that it is known as robust against noises and provides a principled way of determining the plausible number of discrete levels (i.e. model selection). Our main tool is continuous (density) hidden Markov models (HMMs) [7], and the experiments show that the output of Persist [8] or SAX [9] can be a good guide for building a robust HMM against noisy time series.

The remainder of this paper is structured as follows. In Section II, we describe the previous approaches to unsupervised time series discretization and clarify our motivation. Section III and IV respectively describe the proposed method and the experimental results. In Section V, we conclude the paper and mention the future work.

II. PREVIOUS APPROACHES

In this section, we make a review on unsupervised discretization with a little detailed description on the methods our hybrid method uses. Also we present a reproduced result of a comparative experiment by Mörchen and Ultsch [8] and discuss the advantages and disadvantages of the existing methods to make our motivation clear.

Before starting, let us make some preliminaries. In this paper, we consider to discretize a univariate time series $x = (x_1, x_2, \ldots, x_T)$ into $\tilde{x} = (k_1, k_2, \ldots, k_T)$ where T is the length of x, each x_t is the measurement at discrete time t, and k_t is a discrete level into which x_t is converted ($t \in [1, T]$). We use positive integers to indicate the discrete levels, and will map a larger measurement into a higher discrete level. K denotes the number of possible levels (hence $k_t \in [1, K]$). Also let $\hat{\mu}(x)$, $\hat{\sigma}^2(x)$, and $\zeta_q(x)$ denote the sample mean, the sample variance and the q-percentile

value (e.g. $\zeta_{50}(x)$ is the median) of the measurements appearing in a time series x, respectively.

A. Binning, clustering and smoothing

In the literature of discretization, many algorithms have been proposed for supervised situations, but unsupervised discretization has been less explored [2], and also for the case of time series discretization. In a simplest way, binning or clustering methods, such as equal width binning, equal frequency binning, K-means, Gaussian mixture models and so on, can be applied as stand-alone discretizers (e.g. [1], [10]), though the temporal information is lost. On the other hand, while the temporal characteristics of interest (e.g. periodicity, sharp peaks/valleys, long trends and so on) would differ according to the purpose, following [8], we focus on the time series whose flat portions (referred to as "enduring states" in [8]) are crucial to the goal of the application. Hence, we aim to reveal the potentially flat portions by smoothing (by removing noises), and then to map them into discrete levels. For instance, Geurts's discretization [11] makes a smoothing based on a regression tree where the time axis is recursively segmented so that the measurements in each segmented time interval are as close as possible. Piecewise aggregate approximation, which will be described in the next section, is another smoothing method. Also as done in [10], we may apply smoothing filters like Savitzky-Golay filters in advance to the noisy time series.

B. SAX

One may consider from the above that it is reasonable to combine the binning/clustering/smoothing methods, sequentially or simultaneously. Symbolic Aggregate approXimation (SAX) [9] is a well-known algorithm in which the measurements in a time series are first smoothed in each of equal-width segments (frames) at the time axis by piecewise aggregate approximation (PAA), and the smoothed measurements are then grouped into K equal frequency bins under the assumption that the measurements follow a Gaussian distribution.

More formally speaking, in PAA, a raw time series x of length T is compressed into a time series $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_{T'})$ where $T' < T$. For simplicity, we suppose $T = wT'$ where w is some positive integer which indicates the frame width[1]. Each $\bar{x}_{t'}$ is computed as $\bar{x}_{t'} = \frac{1}{w}\sum_{t=w(t'-1)+1}^{wt'} x_t$, the average of the measurements in the t'-th segment ($t' \in [1, T']$). Then, SAX classifies $\bar{x}_{t'}$ into the k-th bin $[\beta_{k-1}, \beta_k)$ where $\beta_{k'} = \Phi^{-1}_{\hat{\mu}(x),\hat{\sigma}^2(x)}(\frac{k'}{K})$ ($k \in [1, K]$, $k' \in [0, K]$). Here Φ^{-1}_{μ,σ^2} is the inverse cumulative distribution function of a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Finally we construct a discretized time series $\tilde{x} = (k_1, k_2, \ldots, k_{T'})$ where $\bar{x}_{t'} \in [\beta_{k_{t'}-1}, \beta_{k_{t'}})$. We can see β_k's ($k \in [1, K-1]$) as the breakpoints at the measurement axis, and these breakpoints can also be used later as a byproduct.

[1]See Section 3.5 of [9] for the case in which T is not divisible by T'.

In the original description, SAX takes as input a standardized time series, and then it only depends on K and T' above, which are specified by the user. In this sense, SAX is not data-adaptive. On the other hand, the compression by PAA brings a good approximation of the original series and a significant improvement in efficiency at the later tasks, called dimension reduction [9].

C. Persist

The algorithm named Persist [8] makes variable-width binning at the measurement axis adaptively to the input time series. In this algorithm, we first regard the event that a measurement x_t at time t falls into the k-th bin $[\beta_{k-1}, \beta_k)$ as the event that the time series stays at an enduring state s_k at time t ($k \in [1, K]$, $t \in [1, T]$). Accordingly a time series is considered to move around in the state space $S = \{s_1, s_2, \ldots, s_K\}$. Then, for a state space S, we introduce a heuristic score, named the persistence score, as $Persistence(S) \stackrel{\text{def}}{=} \frac{1}{K}\sum_{k=1}^{K} Persistence(s_k)$. Each $Persistence(s_k)$ is defined as $\mathrm{sgn}(\hat{P}(s_k|s_k) - \hat{P}(s_k)) \cdot \mathrm{SKL}(\hat{P}(s_k|s_k), \hat{P}(s_k))$, where $\hat{p}(s_k|s_k)$ is the estimated self transition probability at the state s_k, $\hat{p}(s_k)$ is the estimated probability of staying at s_k, and $\mathrm{SKL}(q, q')$ is the symmetric Kullback-Leibler divergence between two Bernoulli distributions $\{q, 1-q\}$ and $\{q', 1-q'\}$. The persistence score is based on a Markov process, and measures the total persistence (or enduringness) of the state space S with respect to the input time series. Under this setting, Persist tries to find a state space, or a set of variable-width bins, S that makes $Persistence(S)$ as high as possible. After the bins found, similarly to SAX, the time series is discretized and the bin boundaries or the breakpoints β_k can be used as a byproduct.

The higher persistence leads to the higher smoothness of the discretized time series, so we can say that Persist performs binning at the measurement axis and smoothing along the time axis simultaneously. Also by definition, we can use the persistence score to measure the plausibility of the number K of discrete levels. Due to the space limit, we omit the algorithmic details.

D. Continuous hidden Markov models

HMMs are one of the standard tools for analyzing sequence data in various application fields such as speech recognition [7], [12]. As in [8], continuous HMMs can also be used for discretization with K levels, in which each hidden state s_k corresponds to the k-th discrete level. Similarly to Persist, we have a state space $S = \{s_1, s_2, \ldots, s_K\}$ and a time series is considered to move around in this space. A continuous HMM has the initial state distribution $p(Q_1 = s_k)$, the transition distributions $p(Q_{t+1} = s_{k'}|Q_t = s_k)$ and the emission distributions (densities) $p(X_t = x_t|Q_t = s_k)$, where Q_t is a random variable which takes the state at time

t and X_t is a random variable which takes the measurement at time t.

In HMMs considered in this paper[2], the emission distribution at the k-th state $p(X_t = x_t|Q_t = s_k)$ follows a Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k^2)$. In our case, this indicates that the k-th level has its own mean μ_k and variance σ_k^2 of the measurements. Without loss of generality, we assume that $\mu_k \leq \mu_{k'}$ if $k \leq k'$. Discretization is then performed by the Viterbi algorithm [7] which takes as input a raw time series $x = (x_1, x_2, \ldots, x_T)$ and returns the most probable state sequence $z^* = (s_{k_1}, s_{k_2}, \ldots, s_{k_T})$, from which we can construct the discretized time series $\tilde{x} = (k_1, k_2, \ldots, k_T)$.

Under the maximum likelihood framework, the parameters of the above distributions are estimated from the input time series x by the forward-backward algorithm [7]. The forward-backward algorithm can be seen as a temporal extension of probabilistic clustering of the measurements based on a univariate Gaussian mixture model, and the estimated positions ($\hat{\mu}_k$) and the shapes ($\hat{\sigma}_k^2$) of the emission Gaussian distributions are crucial in discretization. In the Viterbi algorithm, on the other hand, the probabilities of self transitions work as weights to keep the HMM staying at the same state as long as possible, and consequently play a role of smoothing.

E. Experiment with the enduring-state dataset

In this section, to discuss the characteristics of the existing discretization methods above, we present a reproduced result of the comparative study in [8] with an artificial dataset, which we hereafter call the enduring-state dataset. In addition to accuracy adopted in [8], we introduce normalized mutual information (NMI) [13] as an evaluation criterion on predictive performance. NMI is frequently used in evaluation of a clustering result.

In the enduring-state dataset, raw time series of length 1,000 are generated by a state machine which randomly changes its state after a random duration. See [8] for more details. At each state, the data points (the measurements) are generated with Gaussian noises around the mean proper to the state. The generation process is thus close to a hidden Markov process, but additionally, some of the data points are replaced with outliers. The ratio of these outliers varies from 0% to 10%. Fig. 1 above shows a time series with five states and 5% outliers. As in Persist and HMMs, each state corresponds to a discrete level. Hereafter the state sequence obtained in the sampling process of a raw time series is called the *answer* sequence, and the output of a discretizer is called the *predicted* sequence.

The goal here is to see how well the discretizers recover the answer sequence from the noisy time series. We pick up six methods to compare from [8]: equal width binning (EQW), equal frequency binning (EQF), SAX, Persist, clustering by Gaussian mixture models (GMM), continuous HMMs (HMM). We tested these methods on 100 time series for each number K of discrete levels and ratio R of outliers. In the forward-backward algorithm for HMMs, the means and variances of the emission distributions are initialized as the sample mean $\hat{\mu}(x)$ (with small Gaussian noises) and the sample variance $\hat{\sigma}^2(x)$ of the raw time series x, and we chose the one achieving the highest likelihood from 100 reinitializations. In SAX, we picked up the frame width from $\{1, 2, 3, 5, 10, 20, 50\}$ that works best for each pair of K and R. Fig. 1 below shows the median accuracy (left) and the median NMI (right) for the time series with five discrete levels and various ratios of outliers. The error bars indicate the 95% median absolute deviation (MAD) t confidence interval [14].

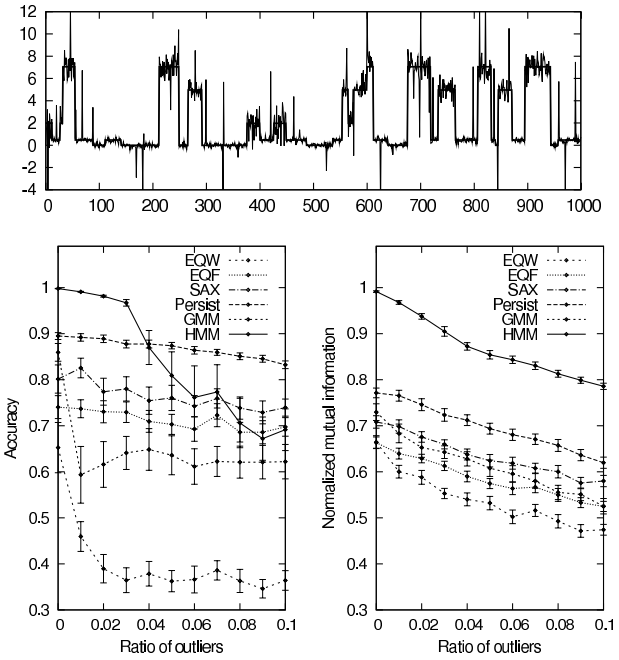

Figure 1. (above) An enduring-state time series. The plots with a thick line indicate the answer sequence. (below) Predictive performance of the existing methods in the enduring-state dataset with five levels.

Under the accuracy, as reported in [8], a direct use of binning or clustering (EQW, EQF or GMM) works poorly, and SAX's normality assumption at the measurement axis seems not to fit to this dataset. Also it is observed that continuous HMMs work nearly perfectly for the time series without outliers (this is not surprising from the way of generating time series), but their performance quickly degenerates as the outliers increase. Persist, on the other hand, is found to be stable (i.e. the variances of accuracies are small) and robust against outliers. Interestingly, the results under NMI show a different tendency. That is, the degeneration of the HMMs' performance is similar to the others, and HMMs constantly outperform them.

The difference in tendencies comes from the nature of

[2]HMMs in [7] uses a Gaussian mixture as the emission distribution. Our version is a special case with only one mixture component.

evaluation criteria and the nature of the discretization methods. The accuracy is defined on exact matchings of the discrete levels between the answer sequence and the predicted sequence (i.e. we check if k_t and k'_t are equal or not for each time t). So it is crucial under the accuracy to identify (the breakpoints or the means of) the discrete levels in the answer sequence. Contrastingly, NMI does not depend on such exact matchings and just indicates the degree of overlapping between two groupings of data points (that is why NMI is used for the evaluation of a clustering result). The answer sequences are of course smooth, and thus NMI emphasizes the smoothness of the predicted sequence. Indeed, after a closer look into the predicted sequences, we found that, in Persist, many non-outliers (only perturbed by Gaussian noises) go across the boundaries of the bins, and HMMs frequently fail to identify the answer discrete levels for the time series with many outliers. From these observations and the descriptions in Sections II-C and II-D, we summarize that, in this dataset, Persist is robust in identifying the underlying discrete levels thanks to the persistence score's *global* nature, while HMMs are good at *local* smoothing among the neighboring measurements. Next, we propose a principled way to exploit these promising characteristics at the same time.

III. HYBRIDIZATION VIA VARIATIONAL BAYES

To exploit the advantages of the heterogeneous discretizers, i.e. Persist (or another method) and continuous HMMs, we propose a hybrid method based on variational Bayes (VB) [5], [6]. Bayesian learning in general is known to be robust against noises, and VB is a fast approximate Bayesian learning that makes available a dynamic programming based procedure such as the forward-backward algorithm. Although there have been a couple of previous works which apply VB to continuous HMMs [15], [12], to the best of our knowledge, our hybrid method is the first attempt to introduce VB to discretization tasks. The rest of this section first explains how VB is applied to continuous HMMs, and then, in Section III-D, describes how we hybridize Persist and HMMs in the VB framework.

A. Model description

In Section II-D, we have explained that a continuous HMM has the initial state distribution $p(Q_1 = s_k)$, the transition distributions $p(Q_{t+1} = s_{k'}|Q_t = s_k)$ and the emission distributions $p(X_t = x_t|Q_t = s_k)$. Hereafter we denote the state at time t by z_t, and abbreviate $p(X_t = x_t)$ and $p(Q_t = z_t)$ as $p(x_t)$ and $p(z_t)$, respectively. Then, $p(x, z)$ is the joint distribution where x is a time series and z is the sequence of states the HMM stays at while generating x. In HMMs, this joint distribution is factored as follows:

$$p(x, z) = p(z_1) \left(\prod_{t=1}^{T-1} p(z_{t+1}|z_t) \right) \left(\prod_{t=1}^{T} p(x_t|z_t) \right). \quad (1)$$

Here $p(z_1)$ follows a categorical distribution $\{\pi_1, \pi_2, \ldots, \pi_K\}$, where π_k is the probability of the state s_k being chosen as the initial state. Similarly, when $z_t = s_k$, $p(z_{t+1}|z_t)$ follows a categorical distribution $\{\pi_{k1}, \pi_{k2}, \ldots, \pi_{kK}\}$, where $\pi_{kk'}$ is the probability of the state $s_{k'}$ being chosen as the next state. Besides, when $z_t = s_k$, $p(x_t|z_t)$ follows a Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k^2)$. Then, π_k, $\pi_{kk'}$, μ_k and σ_k ($k, k' \in [1, K]$) are considered as parameters of the joint distribution, and jointly denoted by θ.

In Bayesian approaches, we consider the extended joint distribution $p(x, z, \theta)$, which is the product of the prior distribution $p(\theta)$ and the likelihood $p(x, z|\theta)$. For mathematical convenience, we use conjugate priors to define the prior distribution. That is,

$$p(\theta) = p(\pi_1, \ldots, \pi_K) \prod_{k=1}^{K} p(\pi_{k1}, \ldots, \pi_{kK}) p(\mu_k) p(\lambda_k) \quad (2)$$

where we introduce the precision $\lambda_k = 1/\sigma_k^2$ and assume

$$p(\pi_1, \ldots, \pi_K) \sim \mathcal{D}(\pi_1, \ldots, \pi_K|\alpha_1, \ldots, \alpha_K) \quad (3)$$
$$p(\pi_{k1}, \ldots, \pi_{kK}) \sim \mathcal{D}(\pi_{k1}, \ldots, \pi_{kK}|\alpha_{k1}, \ldots, \alpha_{kK}) \quad (4)$$
$$p(\mu_k) \sim \mathcal{N}(\mu_k|m_k, (\tau\lambda_k)^{-1}) \quad (5)$$
$$p(\lambda_k) \sim \mathcal{G}(\lambda_k|a, b). \quad (6)$$

Here $\mathcal{D}(\rho_1, \ldots, \rho_K|\alpha_1, \ldots, \alpha_K)$ and $\mathcal{G}(\lambda|a, b)$ denote the Dirichlet distribution and the Gamma distribution, respectively (see the appendix for their definitions). Here α_k, $\alpha_{kk'}$, m_k, τ, a and b ($k, k' \in [1, K]$) are the parameters of the prior distribution, called hyperparameters and jointly denoted by ϕ.

B. VB learning for continuous HMMs

In variational Bayes (VB), we first consider the logarithm of the marginal likelihood $L(x) \stackrel{\text{def}}{=} \log p(x) = \log \sum_z \int_\Theta p(x, z, \theta) d\theta$. Here we fix the hyperparameters ϕ and omit them for the moment. Now x is said to be incomplete since we cannot know the state sequence z from x, and then, some approximation method is required to compute $L(x)$. We consider an approximation of $L(x)$ via VB. To be more specific, we introduce the variational free energy $F[q] \stackrel{\text{def}}{=} \sum_z \int_\Theta q(z, \theta) \log \frac{p(x,z,\theta)}{q(z,\theta)} d\theta$, which is a functional of the test distribution $q(z, \theta)$. It can be shown that $F[q]$ is a lower bound of $L(x)$, and hence the maximized free energy $F[q^*]$ can be seen as a good approximation of $L(x)$. Then, as we will see in the next section, the maximizer q^* is used to find a desirable discretization.

In VB, we assume $q(z, \theta) \approx q(z)q(\theta)$, and obtain a generic form of variational Bayesian EM (VB-EM) algorithm as an iterative procedure consisting of the following two updating rules:

$$\text{VB-E step}: \; q(z) \propto \exp\left(\int_\Theta q(\theta) \log p(x, z|\theta) d\theta \right) \quad (7)$$
$$\text{VB-M step}: \; q(\theta) \propto p(\theta) \exp\left(\sum_z q(z) \log p(x, z|\theta) \right) \quad (8)$$

Now we can derive the algorithm specific to continuous HMMs by substituting the distribution form of continuous HMMs to the generic VB-EM procedure above. As a result, we obtain the adjusted hyperparameters ϕ^* that specify q^*: $\overline{\alpha}_k$, $\overline{\alpha}_{kk'}$, $\overline{m}_k$, $\overline{\tau}_k$, $\overline{a}_k$ and $\overline{b}_k$. The derived VB-EM algorithm is presented in the appendix.

To determine the number K^* of discrete levels, we further extend the joint distribution to $p(x, z, \theta, K)$ where K indicate the number of discrete levels. Then we find $K = K^*$ that maximizes the probability $p(K|x)$. Assuming $p(K)$ is uniform, and having $p(K|x) = p(x|K)p(K)/p(x)$ from the Bayes' theorem, this maximization is equivalent to the maximization of the logarithm of the marginal likelihood $L(x) = \log p(x|K) = \log \sum_z \int_\Theta p(x, z, \theta|K) d\theta$. Again, $L(x)$ is approximated by the variational free energy, which is thus used as the score on the plausibility of the number of discrete levels.

C. Finding the discretized time series

From the property $L(x) - F[q] = \mathrm{KL}(q(z, \theta), p(z, \theta|x))$, where KL denotes the Kullback-Leibler divergence, finding the maximizer q^* of $F[q]$ leads to a good approximation of $p(z, \theta|x)$, the posterior distribution of hidden state sequence and the parameters. Averaging by this approximated posterior distribution, we compute the predictive distribution of the (hidden) state sequence $p(z|x) \propto \int_\Theta q^*(\theta) p(x, z|\theta) d\theta$ for the input series x. Finally we get the most probable state sequence (the discretized time series) $z^* = \mathrm{argmax}_z p(z|x)$. Note here that $p(z|x)$ is analytically obtained but cannot be computed in a dynamic programming fashion [6]. To remedy this problem, we take a heuristic approach known as reranking [16]. That is, we first find top-n ranked state sequences $\{z_1, \ldots, z_n\}$ by the n-best Viterbi algorithm (we used $n = 10$) working on an HMM whose initial state distribution, transition distributions, and emission distributions are independently averaged by $q^*(\theta)$[3]. Then, we compute $p(z_i|x)$ exactly for each $i \in [1, n]$, and finally obtain z^* as $\mathrm{argmax}_{z_i : i \in [1,n]} p(z_i|x)$.

D. Hyperparameter settings for hybridization

As in Sections II-D and II-E, in discretization, it is crucial to find the positions (μ_k) and the shapes (σ_k^2) of the emission Gaussian distribution which capture well the input series x. The posterior distribution of μ_k (i.e. $q^*(\mu_k)$) follows a Gaussian distribution whose mean is the adjusted $\overline{m}_k$, and hence $\overline{m}_k$ eventually affects the quality of discretization. In the VB-M step, we obtain $\overline{m}_k$ by

$$\overline{m}_k := (\tau m_k + \overline{T}_k \overline{x}_k)/(\tau + \overline{T}_k), \tag{9}$$

where m_k and τ are the hyperparameters, $\overline{x}_k$ and $\overline{T}_k$ can be interpreted as the mean of real values emitted while staying at the state s_k, and the expected counts of staying at s_k, respectively. We can see from Eq. 9 that $\overline{m}_k$ is an weighted sum of m_k and $\overline{x}_k$, and that $\overline{m}_k$ can be controlled by hyperparameters m_k and τ, which are the mean from our prior knowledge and its weight, respectively.

In combining HMMs with Persist, we determine m_k using the breakpoints β_k obtained as a byproduct by Persist, and the 5% and 95% percentile values of the input x. Specifically, we compute $m_k := (\beta'_{k-1} + \beta'_k)/2$ where $\beta'_k = \beta_k$ for $k \in [1, K-1]$, $\beta'_0 = \zeta_5(x)$ and $\beta'_K = \zeta_{95}(x)$. The weight τ is set to balance the prior knowledge and the input series. The hyperparameters α_k and $\alpha_{kk'}$ play the same role, and we fixed $a \approx \frac{1}{2}$ and $b = \frac{1}{2}\hat{\sigma}^2(x)$ in our experiments. Note that we can also combine HMMs with SAX using the breakpoints β_k, which are SAX's byproduct. The hybridization above is surely simple, but is flexible since the hyperparameters are the only connection point between the discretizers to be combined.

IV. EXPERIMENTS

A. The enduring-state dataset revisited

To test our hybrid method, we conducted three experiments with artificial and real datasets. First, we test our method with the enduring-state dataset, described in Section II-E. The experimental settings are the same except that we set $\alpha_k = 1$ and $\alpha_{kk'} = 1$, and chose the hyperparameter τ from {0.5, 1, 5, 10, 20, 50, 70, 100} that works best for each pair of the number K of discrete levels and the ratio R of outliers. The results with five levels are shown in Fig. 2 in which "HMM+P" and "HMM+S" respectively indicate the HMM combined with Persist and the one combined with SAX. As expected, these hybrids outperform the original single methods under both accuracy and NMI. For the other cases under accuracy, by Wilcoxon's rank sum test with the significance level 0.01, "HMM+P" is shown to be better than the original Persist, except several cases with a large number of levels and many outliers[4]. This superiority was also confirmed under NMI for all cases.

B. Time series classification

The task in the second experiment is supervised time series classification, where we classify a whole time series into one of the predefined classes. The 1-nearest neighbor (1-NN) classifiers are often used with the Euclidean distance $\Delta(x, x') \stackrel{\mathrm{def}}{=} \sqrt{\sum_{t \in [1,T]} (x_t - x'_t)^2}$ as a dissimilarity measure between two raw time series x and x'. In this experiment, to see how the discretization methods affect the classification performance, we replace Δ with its discretized

[3]Consequently, the initial state distribution, the transition distributions, and the emission distributions respectively follow a categorical distribution $\{\pi^*_k = \overline{\alpha}_k / \sum_\ell \overline{\alpha}_\ell\}_{k \in [1,K]}$, categorical distributions $\{\pi^*_{kk'} = \overline{\alpha}_{kk'} / \sum_\ell \overline{\alpha}_{k\ell}\}_{k' \in [1,K]}$ and student's t-distributions (details omitted).

[4]More precisely, in the cases (K, R) = (6, 7%), (6, 8%), (6, 10%), (7, 4%), (7, 7%), (7, 8%), (7, 9%) and (7, 10%), there is no significant difference.

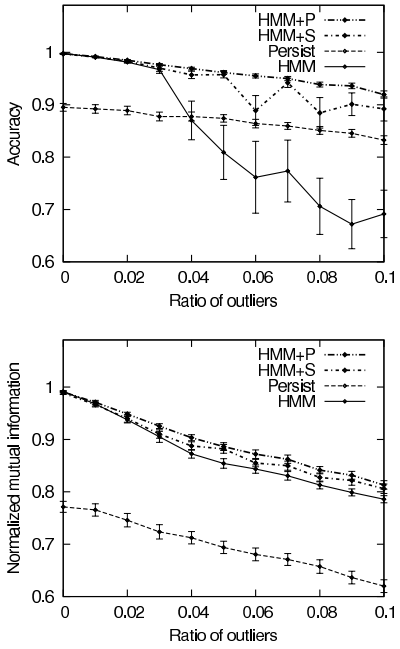

Figure 2. Predictive performance of Persist, HMMs and two hybrids in the enduring-state dataset with five discrete levels. The plots for Persist and HMMs are the same as those in Fig. 1.

version $\tilde{\Delta}(x, x') \stackrel{\text{def}}{=} \sqrt{\sum_{t\in[1,T]}(k_t - k'_t)^2}$, where $(k_1, \ldots, k_T)$ and $(k'_1, \ldots, k'_T)$ are the discretized series of x and x', respectively[5].

We worked on two widely-used artificial datasets called the control chart (CC) dataset and the cylinder-bell-funnel (CBF) dataset[6]. The CC dataset contains 600 time series of length 60 and has six classes, and the CBF dataset contains 798 time series of length 128 and has three classes. All time series in each dataset are standardized in advance. We conducted 10 times 10-fold stratified cross-validation. In each fold, we first train a discretizer with the training time series, and then discretize all time series by the trained discretizer[7]. Finally we apply the 1-NN classifier in a usual manner. In SAX, we chose the frame width w from {1, 2, 3, 5, 10, 15, 20, 25, 30} that works best. Also we trained HMMs under $\alpha_k = 1$, $\alpha_{kk'} = 1$ and τ chosen from $\{1, 10, 100, 1000\}$ with 10 reinitializations. Table I shows the error rates (%) with 95% confidence interval under Student's t-distribution with various numbers K of discrete levels. In each row, the smallest error rate is marked with the '*' symbol. Without discretization (i.e. using Δ), the error rates (%) are 7.90±0.62 for CC and 1.85±0.27 for CBF.

[5] $\tilde{\Delta}$ indicates that the penalty cost of mismatching between two discrete levels is the square of their difference. In SAX, a dissimilarity measure, called MINDIST, is proposed [9]. However, we do not present the results with MINDIST here, since the classifier based on MINDIST worked poorly with fewer discrete levels (e.g. with three levels).

[6] The CC dataset is available from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/), and the CBF dataset was generated by the authors based on the description in [11].

[7] This procedure is necessary for data-adaptive discretizers such as Persist and HMMs. We extended Persist and HMMs in a straightforward way to handle more than one time series.

Table I
PREDICTIVE PERFORMANCE IN TIME SERIES CLASSIFICATION.

Error rates (%) for the CC dataset

K	SAX	Persist	HMM	HMM+SAX	HMM+Persist
2	*24.61±0.94	24.82±1.01	38.15±1.71	36.75±0.72	36.70±0.70
3	*4.73±0.55	18.58±0.97	8.70±0.90	8.00±0.58	8.55±0.60
4	*5.08±0.58	17.17±1.07	16.48±1.67	10.38±0.80	11.65±1.16
5	*3.03±0.42	14.03±0.84	19.58±1.63	25.08±0.70	14.95±1.52
6	*2.52±0.38	10.67±0.79	15.68±2.15	25.47±0.73	23.33±0.60
7	*1.57±0.30	11.73±0.84	14.65±2.65	6.23±0.62	11.05±1.31
8	*1.40±0.30	11.40±0.71	12.92±2.48	4.40±0.62	6.47±0.92

Error rates (%) for the CBF dataset

K	SAX	Persist	HMM	HMM+SAX	HMM+Persist
2	*23.64±0.82	25.99±0.77	27.55±1.03	27.50±1.02	25.81±1.01
3	5.51±0.48	12.29±0.80	2.43±0.33	2.29±0.31	*1.74±0.27
4	4.41±0.48	4.01±1.28	*0.76±0.19	*0.76±0.19	1.24±0.24
5	2.84±0.38	1.05±0.25	1.23±0.27	*0.98±0.24	0.99±0.22
6	1.85±0.33	2.83±0.45	0.80±0.21	*0.74±0.19	0.83±0.20
7	1.40±0.24	1.90±0.30	1.20±0.25	0.90±0.26	*0.82±0.20
8	1.43±0.27	1.99±0.44	*0.83±0.21	1.01±0.25	0.88±0.19

In the results for the CC dataset, SAX outperforms the other methods including the method without discretization. This seems to be due to the effect of smoothing by PAA [1], [9], and more importantly to the assumption made by the methods other than SAX — time series have flat portions which are crucial to the goal of the application. Indeed, the time series with long increasing/decreasing trend incur more errors to these methods. On the other hand, it should be remarked that the two hybrids "HMM+SAX" and "HMM+Persist" surely improve the performance of 1-NN over the original HMMs in most cases. Also with three discrete levels, the performance with HMM-based methods is comparable to the one without discretization.

For the CBF dataset, the performance with two hybrids is better than the one without discretization, and than the one with SAX or Persist especially when we have fewer discrete levels. This result is important since from the viewpoint of comprehensibility, the number of discrete levels should be preferred to be small. Besides, we observed that, in both datasets, a large τ is sometimes needed for hybrids, since the training set contains numerous measurements (e.g. $600 \times 0.9 \times 60 = 32,400$ in the case of CC).

C. The muscle dataset

In the last experiment, following [8] again, we discretize the time series on the muscle activation of a professional inline speed skater [10][8], focusing on determining the number of discrete levels. In this dataset, the measurement is the muscle activation calculated from the original EMG

[8] The dataset is included in the package of Persist's MATLAB implementation (http://www.mybytes.de/persist.php).

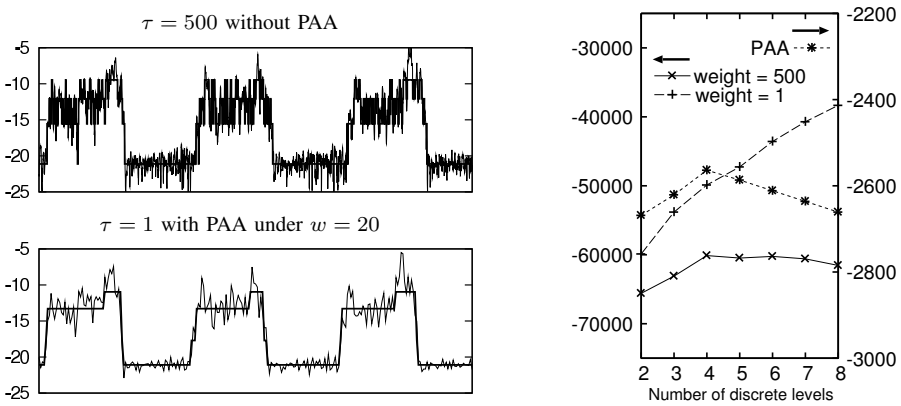

Figure 3. (left-above) The original and discretized time series. (left-below) The time series compressed by PAA and discretized. (right) The variational free energy with different number of discrete levels.

(Electromyography) as the logarithm of the energy, and nearly 30,000 measurements are recorded. The plots with a thin line in Fig. 3 left-above shows the original time series.

Since the time series contains numerous measurements, to determine the number of discrete levels, we should set relatively large numbers to τ, α_k and $\alpha_{kk'}$, the weights for prior knowledge (Section III-D). Otherwise, the estimated number K^* of discrete levels turns to be counterintuitively large. This overfitting problem is illustrated in Fig. 3 right, where the x-axis indicates the number K of discrete levels and the y-axis indicates the variational free energy (VFE) with K levels. The plots "weight = 1" and "weight = 500" show the VFEs under $\tau = 1$ and $\tau = 500$, respectively[9]. Also, we can have a similar effect by compressing the time series with PAA. The plots "PAA" indicate the VFEs under the frame width $w = 20$ and $\tau = 1$.

The plots with thick lines in Fig. 3 left indicate the discretized sequence in the case of $\tau = 500$ without PAA (above), and in the case of $\tau = 1$ with PAA under $w = 20$ (below). In both cases, K^* is estimated as four, but one of the discrete levels (whose mean is at -17.7) appears only rarely. This result coincides with the fact that the expert prefers K^* to be three [8], and the discretized sequence with PAA clearly shows the high activation at the end of each cycle, which means the last kick to the ground to move forward. Note here that this result has been obtained using Persist [8], but we reached a similar result by a more general approach via variational Bayes.

V. CONCLUSION

We proposed a hybrid method for unsupervised time series discretization based on variational Bayes (VB). By hybridization, we can benefit from heterogeneous discretizers which have their own assumption/model on the distribution of data points in a two dimensional (measurement-time) space. As examined by experiments, VB provides a principled and flexible way to build a robust discretizer and to determine the number of discrete levels, which would finally lead to a discretization meaningful for symbolic data mining or other AI systems. Also the hybrid discretizer we built is data-adaptive, and hence is expected to produce a more comprehensible discretization. To the best of our knowledge, the proposed method is the first attempt to introduce VB for discretization tasks.

There are some remaining works. We need to apply the proposed method to other datasets in time series classification. Computational issues are important as well. In particular, the computation time of HMM-related algorithms is quadratic to the number of discrete levels. This problem is remedied by a compression method like PAA, and as mentioned in Section IV-B, fewer discrete levels are preferred from the viewpoint of comprehensibility. Besides, there are cases where we wish to discretize multivariate time series with common discrete levels. Synnaeve et al. [4] realized it by extending HMMs with the notion of parameter tying [7].

[9]In this experiment, we always set $(\tau + 1)$ to the other weight hyperparameters α_k and $\alpha_{kk'}$, and tried 500 reinitializations in the VB-EM algorithm.

ACKNOWLEDGMENT

This work is supported in part by Grant-in-Aid for Scientific Research (No. 20240016) from Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

[1] C. Daw, C. Finney, and E. Tracy, "A review of symbolic analysis of experimental data," *Review of Scientific Instruments*, vol. 74, no. 2, pp. 915–930, 2003.

[2] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. of the 12th Int'l Conf. on Machine Learning (ICML-95)*, 1995, pp. 194–202.

[3] R. King, S. Garrett, and G. Coghill, "On the use of qualitative reasoning to simulate and identify metabolic pathways," *Bioinformatics*, vol. 21, no. 9, pp. 2017–2026, 2005.

[4] G. Synnaeve, A. Doncescu, and K. Inoue, "Kinetic models for logic-based hypothesis finding in metabolic pathways," in *19th Int'l Conf. on Inductive Logic Programming (ILP-09)*, 2009.

[5] H. Attias, "A variational Baysian framework for graphical models," in *Advances in Neural Information Processing Systems 12 (NIPS-99)*, 1999, pp. 209–215.

[6] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University College London, 2003.

[7] L. Rabiner and B.-H. Juang, *Fandamentals of Speech Recognition*. Prentice Hall, 1993.

[8] F. Mörchen and A. Ultsch, "Optimizing time series discretization for knowledge discovery," in *Proc. of the 11th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-05)*, 2005, pp. 660–665.

[9] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[10] F. Mörchen, A. Ultsch, and O. Hoos, "Extracting interpretable muscle activation patterns with time series knowledge mining," *Int'l J. of Knowledge-based and Intelligence Engineering Systems*, vol. 9, no. 3, pp. 197–208, 2005.

[11] P. Geurts, "Pattern extraction for time series classification," in *Proc. of the 5th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD-01)*, 2001, pp. 115–127.

[12] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda, "Application of variational Bayesian approach to speech recognition recognition," in *Advances in Neural Information Processing Systems 15 (NIPS-2002)*, 2002, pp. 1237–1244.

[13] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[14] M. O. Abu-Shawiesh, F. M. Al-Athari, and H. F. Kittani, "Confidence interval for the mean of a contaminated normal distribution," *J. of Applied Science*, vol. 9, no. 15, pp. 2835–2840, 2009.

[15] S. Ji, B. Krishnapuram, and L. Carin, "Variational Bayes for continuous hidden Markov models and its application to active learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 522–532, 2006.

[16] M. Collins, "Discriminative reranking for natural language parsing," in *Proc. of the 17th Int'l Conf. on Machine Learning (ICML-2000)*, 2000, pp. 175–182.

APPENDIX

The Dirichlet and the gamma distributions are defined as:

$$\begin{aligned}\mathcal{D}(\rho_1,\ldots,\rho_K|\alpha_1,\ldots,\alpha_K) &\stackrel{\text{def}}{=} \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_K)} \prod_{k=1}^K \rho_k^{\alpha_k-1}\\ \mathcal{G}(\lambda|a,b) &\stackrel{\text{def}}{=} \frac{1}{\Gamma(a)} b^a \lambda^{a-1}\exp(-b\lambda),\end{aligned}$$

where Γ is the gamma function. In the VB-EM algorithm, we initialize $\overline{\alpha}_k = \alpha_k + \varepsilon$, $\overline{\alpha}_{kk'} = \alpha_{kk'} + \varepsilon$, $\overline{m}_k := m_k + \varepsilon$, $\overline{a}_k := a$, $\overline{b}_k := b$ and $\overline{\tau}_k := \tau$ for each $k \in [1, K]$. Here each of ε's is a distinct small random noise, and the hyperparameters α_k, $\alpha_{kk'}$, m_k, τ, a and b are specified in advance. Then, in the VB-E step, we compute the expectations $\overline{U}_k$, $\overline{T}_{kk'}$, $\overline{x}_k$ and $\overline{S}_k$ by the following procedure:

$$\begin{aligned}q(z) &:= q(z_1)\left(\textstyle\prod_{t=1}^{T-1} q(z_{t+1}|z_t)\right)\left(\prod_{t=1}^{T} q(x_t|z_t)\right)\\ q(s_k) &:= \exp\left(\Psi(\overline{\alpha}_k) - \Psi\left(\textstyle\sum_k \overline{\alpha}_k\right)\right)\\ q(s_{k'}|s_k) &:= \exp\left(\Psi(\overline{\alpha}_{kk'}) - \Psi\left(\textstyle\sum_{k'} \overline{\alpha}_{kk'}\right)\right)\\ q(x_t|s_k) &:= \exp\left(-\tfrac{1}{2}\left(\log 2\pi + \tfrac{1}{\overline{\tau}_k} + \log \overline{b}_k - \Psi(\overline{a}_k)\right)\right)\cdot\\ &\qquad \exp\left(-\tfrac{\overline{a}_k}{2\overline{b}_k}(x_t - \overline{m}_k)^2\right)\\ \overline{U}_k &:= \textstyle\sum_{z:z_1=s_k} q(z)\\ \overline{T}_{kk'} &:= \textstyle\sum_z q(z)\left|\{t \in [2,T] \mid z_{t-1} = s_k, z_t = s_{k'}\}\right|\\ \overline{x}_k &:= \tfrac{1}{\overline{T}_k}\textstyle\sum_z q(z)\sum_{t:t\in[1,T],z_t=s_k} x_t\\ \overline{S}_k &:= \tfrac{1}{\overline{T}_k}\textstyle\sum_z q(z)\sum_{t:t\in[1,T],z_t=s_k}(x_t - \overline{x}_k)^2,\end{aligned}$$

where $\overline{T}_k \stackrel{\text{def}}{=} \sum_{k'} \overline{T}_{kk'}$ and Ψ is the digamma function: $\Psi(x) \stackrel{\text{def}}{=} \frac{d}{dx}\log\Gamma(x)$. The VB-E step can be computed by the forward-backward algorithm on an HMM whose initial state distribution, transition distributions and emission distributions are $q(z_1)$, $q(z_{t+1}|z_t)$ and $q(x_t|z_t)$, respectively. In the VB-M step, on the other hand, we update the hyperparameters as below.

$$\begin{aligned}\overline{\alpha}_k &:= \alpha_k + \overline{U}_k\\ \overline{m}_k &:= (\tau m_k + \overline{T}_k\overline{x}_k)/(\tau + \overline{T}_k)\\ \overline{\alpha}_{kk'} &:= \alpha_{kk'} + \overline{T}_{kk'}\\ \overline{a}_k &:= a + \frac{1}{2}\overline{T}_k\\ \overline{\tau}_k &:= \tau + \overline{T}_k\\ \overline{b}_k &:= b + \frac{1}{2}\left(\frac{\tau\overline{T}_k}{\tau + \overline{T}_k}(m_k - \overline{x}_k)^2 + \overline{S}_k\right)\end{aligned}$$

We iterate the VB-E step and the VB-M step alternately until the convergence of the variational free energy derived as:

$$\begin{aligned}F[q] &:= \log\textstyle\sum_z q(z) + \sum_k \log\frac{\Gamma(\overline{\alpha}_k)}{\Gamma(\alpha_k)} + \log\frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\sum_k \overline{\alpha}_k)}\\ &\quad + \textstyle\sum_{k,k'} \log\frac{\Gamma(\overline{\alpha}_{kk'})}{\Gamma(\alpha_{kk'})} + \sum_k \log\frac{\Gamma(\sum_{k'}\alpha_{kk'})}{\Gamma(\sum_{k'}\overline{\alpha}_{kk'})}\\ &\quad + \textstyle\sum_k \overline{U}_k\left(\Psi(\overline{\alpha}_k) - \Psi\left(\sum_k \overline{\alpha}_k\right)\right)\\ &\quad + \textstyle\sum_{k,k'} \overline{T}_{kk'}\left(\Psi(\overline{\alpha}_{kk'}) - \Psi\left(\sum_\ell \overline{\alpha}_{k\ell}\right)\right)\\ &\quad + \textstyle\sum_k\left(\frac{1}{2}\log\frac{\tau}{\overline{\tau}} + \log\frac{\Gamma(\overline{a}_k)}{\Gamma(a_k)} + \log\frac{b^a}{\overline{b}_k^{\overline{a}_k}}\right)\\ &\quad + \textstyle\sum_k \frac{\overline{a}_k}{\overline{b}_k}((\overline{b}_k - b) - \frac{1}{2}\tau(\overline{m}_k - m_k)^2)\\ &\quad + \tfrac{1}{2}\textstyle\sum_k \overline{T}_k(\frac{1}{\overline{\tau}_k} + \log\overline{b}_k - \Psi(\overline{a}_k)).\end{aligned}$$