JAIST Repository
https://dspace.jaist.ac.jp/

Title	Spatio-Temporal Symbolization of Multidimensional Time Series
Author(s)	Hidaka, Shohei; Yu, Chen
Citation	2010 IEEE International Conference on Data Mining Workshops (ICDMW): 249-256
Issue Date	2010-12-13
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/9785
Rights	Copyright © 2010 IEEE. Reprinted from 2010 IEEE International Conference on Data Mining Workshops (ICDMW), 2010, 249-256. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of JAIST's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Spatio-Temporal Symbolization of Multidimensional Time Series

Shohei Hidaka* and Chen Yu†
* *School of Knowledge Science, Japan Advanced Institute of Science and Technology*
1-1 Nomi, Ishikawa 923-1292 Japan, Email: shhidaka@jaist.ac.jp
†*Department of Psychological and Brain Sciences, Indiana University*
1101 East 10th Street, Bloomington, IN, 47405 USA, Email: chenyu@indiana.edu

***Abstract*—The present study proposes a new symbolization algorithm for multidimensional time series. We view temporal sequences as observed data generated by a dynamical system, and therefore the goal of symbolization is to estimate symbolic sequences that minimize loss of information, which is called generating partition in nonlinear physics. In order to utilize the theoretical property of symbol dynamics in data mining, our algorithm estimates symbols on multivariate time series by integrating both spatial and temporal information and selecting those dimensions in multidimensional time series containing useful information. Probabilistic symbolic sequences derived from our symbolization method can be used in various supervised and unsupervised data-mining tasks. To demonstrate this, the algorithm is evaluated by applying it to both simulated data and a real-world dataset. In both cases, the new algorithm outperforms its alternative approaches.**

***Keywords*-time series symbolization; generating partition; dynamical system; dimension selection; heterogeneous multivariate time series;**

I. Introduction

Many real-world applications deal with huge amounts of temporal data. State-of-the-art sensing and computing techniques allow us to continuously collect and store multistream data over time. Those data types vary from video and audio streams gathered from surveillance cameras, to financial time series, to user behaviors on social networks, and to observational data from patients. Those data are most often generated from different data resources and simultaneously recorded from different devices with potentially different sampling rates. With a large volume of temporal data (in terms of both the large number of data points and the large number of types of measurements), data mining and knowledge discovery techniques are playing more and more important roles in those domains. In many ways, the success of those applications counts on reliable and effective pattern discovery through data mining algorithms. Multidimensional time series can be viewed as a general form of those datasets (while a 1-dimensional time series can be viewed as a special case) with multiple data points at a single temporal moment and the whole temporal series consists of a sequence of multidimensional vectors. Various kinds of spatial and temporal regularities and patterns are embedded in such data.

One of the important directions in temporal data mining is to convert a time series into a symbolic sequence. By doing so, a discrete representation opens up many powerful techniques on information and communication theory in addition to the connection between discrete mathematics and dynamical systems via the theoretical study of symbolic dynamics. However, the challenge here is how to maintain the benefit of a low-precision symbolic representation and simultaneously to minimize the loss of information in the symbolization process.

Most histogram-based symbolization algorithms consist of two steps: the first step is to compute a histogram distribution of the time series by aggregating the data over time, and the second step is to determine the breakpoints of discretization and then assign each symbol to cover the range between two breakpoints. A well-known approach along this line is called the Symbolic Aggregate approXimation (SAX) [1]. In brief, SAX first transforms the input time series data into a Piecewise Aggregate Approximation (PAA) representation, and then symbolizes the distribution space of PAA representation into a set of discrete symbols. Compared with other approaches, SAX allows lower-bounding distance measures to be defined on the symbolic space that are identical with the original data space. Thus, the information loss through this symbolization and its potential effects on subsequent data processing is minimal when a time series is converted into this efficient symbolic representation. SAX has been successfully used on various pattern detection tasks, such anomaly detection [2] and motif discovery [3]. Compared with many other symbolic representations of time series introduced over the past decades (e.g. wavelets, discrete Fourier transform), the SAX representation has a special advantage of allowing distance measures between symbolic sequences that lower bound corresponding distance measures defined on the original real-vale time series (see a mathematical proof in [1]).

The new symbolization approach presented in this paper is quite different from previous histogram-based approaches (see a more detailed review in Section II). The goal is to extract a symbolic representation from heterogeneous multidimensional data, which has essential information for further pattern discovery and data analysis, and simultaneously reduces continuous raw data into a simple and effective form. The novel idea in our approach is to use both spatial and temporal information encoded in a multidimensional time series in symbolization. Our algorithm differs from previous efforts (e.g. SAX) in several important ways:

First, while most symbolization approaches cannot be effectively extended to multidimensional cases due to the curse of dimensionality, our algorithm takes a multidimensional time series and actively selects dimensions containing useful information in symbolization. The overall symbolization results are based on those principal dimensions containing more useful information. In this way, symbolization and di-

mension reduction are implemented in an integrated system.

Second, while most symbolization approaches convert a continuous value at a specific moment in time into a symbol, our approach views a time series as observed data generated from dynamical systems (with stochastic noises, [4], [5]) and makes use of temporal contextual information (before and after that particular data point) in symbolization. In this way, each symbol contains not only the information about a particular numerical value at this point, but also the information about the temporal trend of time series at this moment. For example, the same numerical value will be assigned with different symbols at different places in the time series based on their temporal contextual information.

Third, to better capture the information in a time series, our approach converts each multidimensional vector at a particular moment in a multidimensional time series into a mixture of symbols, with each symbol assigned with a probability. This can be viewed as a mixture model of symbolization. For example, in a straightforward one-to-one direct symbolization, each symbol covers a certain range of values, if a to-be-converted numerical value is close to the border of two ranges covered by two symbols, a hard-assignment method has to pick one which introduces inaccuracy in those close calls. In our case, our algorithm will keep both symbols and assign a slightly higher probability for one symbol and a slightly lower probability for the other. In this way, each multidimensional vector is mapped to a set of symbols with probabilities, and this mixture of symbols can better encode the precise information in the original time series.

Fourth, the symbolization results obtained in our algorithm give us an abstract form compared with raw multidimensional temporal data, but meanwhile this representation still encodes various kinds of statistical spatio-temporal regularities from the original data. Moreover, this intermediate symbolic representation allows us to apply various existing data mining techniques to extract useful patterns and information. We demonstrate this by using the same symbolization results obtained with our approach to perform pattern classification, information-theoretic measures, and dimension reduction/selection, showing that better results than previous approaches are achieved.

To implement those useful characteristics in symbolization, our algorithm is based on EM and Bayesian updates – iteratively selecting informative dimensions in the original data space and converting those into a mixture of symbols. In the following, we will first briefly review related work in temporal data mining, and we will then provide a conceptual description of the algorithm, followed by mathematical details. Next, we will provide examples with both simulated data and real-world data to demonstrate the utility of the algorithm.

II. Approaches for temporal data mining and Symbolization

In this section, we briefly review relevant temporal data mining and symbolization techniques, grouping methods based on two factors. The first factor is whether the approach assumes a time series is generated by a linear or nonlinear system. The second is the type of results from a specific method. More specifically, whether the analysis aims for tagging (or symbolization) of each data point in a time series, or for summarizing the overall statistics in a sequence. With those two factors, time series analysis methods can be classified into four groups. In the following, we will present one representative method in each group.

Linear/Sequence Statistics: Fourier analysis, one of the standard techniques for time series analysis, is classified as the linear and whole-sequence statistics. It is linear because the Fourier analysis assumes a time series is a linear combination of sub-components with different frequencies. Although some variants of this approach are applicable to local temporal patterns (e.g., short-time Fourier analysis with a moving window), the best temporal resolution one can achieve is theoretically limited due to the uncertainty principle [6]. Namely, selecting short temporal windows has a large degree of uncertainty while large temporal windows generate coarse patterns. Therefore, the results from Fourier analysis are classified as whole sequence statistics.

Linear/Symbolization: Symbolic Aggregation approximation (SAX; [1]) is a symbolization method in which each data point or each group of data points in a given time series is assigned one symbol. Each symbol covers a particular interval in continuous space so that the frequency of data points falling in each symbol is nearly equal across all of the data points in the time series. As a result of the symbolization of time series, the size of the symbol set reflects discretized values of the original continuous values at an arbitrary resolution. One advantage of SAX is that the symbolization process approximates the Euclidean distance between two time series in the original values with some upper bound of errors. However, SAX in principle is based on the linearity assumption, because the distance metrics used are computed by a linear sum of each local time series.

Non-linear/Sequence Statistics: The recurrence plot is one of the techniques based on the nonlinear assumption used to summarize whole sequence patterns. For example, many human (or robot) behaviors have been described as nonlinear dynamic systems, e.g. postural control of a body part and eye movements (e.g. [7]). In brief, the recurrence plot, as a method for visualization and diagnosis, summarizes the recurrence patterns between two temporal trajectories so that users can visually examine and spot patterns that cannot be easily discovered as a time series. Although this nonlinear technique would be preferable for data violating linear system assumptions, one disadvantage is that it typically characterizes the repeated overall patterns of a whole sequence, but not fine-grained patterns such as transitions from one moment to another.

Nonlinear/Symbolization: Finally, the method we propose is a nonlinear approach, and also allows us to analyze fine-grained temporal patterns. Although the linear/nonlinear assumption on the system may depend on what one is interested in, for unsupervised knowledge discovery, a nonlinear technique with few assumptions is preferable, since it can handle a wider range of heterogeneous datasets. Moreover, tagging each data point in a time series allows us to further process them using symbolic algorithms and information-theoretic measures, which may potentially reveal latent structures embedded in a time series. In the next section,

we provide a detailed description of this approach.

III. Generating Partition as symbolization

We view a time series as observable data generated by a nonlinear dynamical system. In nonlinear physics, a partition which symbolizes the subspaces of a given phase space is called generating if a symbolized sequence of sufficient length for different initial points of the system is distinguishable [8]. More intuitively, such a generating partition would not lose any information in discretizing the original phase space, since the given symbol series can be mapped back onto an unique point (or subspace) in the phase space (more explanation later and also see Figure 1). Although a generating partition has theoretical properties preferable to a non-generating partition, it has been supposed to be difficult to achieve for time series without explicit dynamical equations (See [9] for the recent review of nonlinear time series). Recently, a new technique called Symbolic False Nearest Neighbor ([5], [10]; SFNN in short) has been developed to overcome this challenge by estimating a generating partition of a time series without explicit dynamical equations. The central idea of SFNN is to construct a set of partitions which map data points in the phase space to a set of symbols such that two similar sequences in the symbol space are close in the original phase space. Namely, neighboring points in the symbol space should also be neighbors in the original space. In other words, the method constructs a partition by measuring and minimizing the number of false symbolic nearest neighbors.

The present paper is motivated by the empirical insight of SFNN – duality between symbolic and spatial nearest neighborhood is the key to specify a dynamical property of time series. In this study, we extend this idea so that it guides us to find latent structures embedded in heterogeneous time series. Different from the assumption most often used in theoretical simulations, a dataset from the real world is unlikely to be purely generated by a single dynamical system with no stochastic component. Instead, a dataset from the real world may be *heterogeneous* in which multiple independent systems interact with each other with some stochastic components. Therefore, in order to utilize the theoretical sound property of symbolic dynamics for an empirical time series, we need to find out which dimension is of interest. However, this is theoretically and practically challenging as a chicken-and-egg problem: since the essential property of a dynamical system may be characterized by a generating partition, we need to estimate the dynamical property before knowing which dimension may be of interest. Meanwhile, the estimation of the generating partition may depend on how well the given dataset is organized – ideally, it prefers a homogeneous dataset in which all the time series should be generated by a single dynamical system – but we cannot identify which dimension may contain informative structures before estimating its dynamical property.

Our solution to this chicken-and-egg problem is to simultaneously estimate a generating partition and select informative dimensions from the dataset. As mentioned above, the key issue here is to find a symbol set in which symbolic nearest neighbors tend to be the nearest neighbors in the phase space. Different from the SFNN which only optimizes the symbol set in a fixed spatial configuration, in our algorithm, both the spatial configuration and the symbol set are iteratively optimized. More specifically, our algorithm functions in both symbolic and phase spaces. In one step of optimization, a symbolic series is updated based on a given phase space, and in the other step, the spatial configuration is optimized so that the distances between data points in the phase space correlate to symbolic nearest neighbors. We call the algorithm *Stochastic Dual Nearest Neighbor* (SDNN), since both symbolic and phase spaces are mutually optimized to form *dual nearest neighbors*.

A. Probabilistic distribution of generating partition

In the following, we will first give a conceptual idea of the algorithm, and then explain the SDNN algorithm step by step with a formal description. The outline of the present algorithm is shown in Figure 1. Without loss of generality, suppose that we have a one-dimensional time series (Figure 1A). The first step for analyzing such a nonlinear dynamical system is to reconstruct the phase space from a given time series. Since the underlying dynamical nonlinear system may have higher dimensionality than the observed variable, we use time delay embedding in order to reconstruct topological structures of the phase space [11]. The step from Figure 1A to Figure 1B is an example in which a one-dimensional observed series is embedded in three dimensional phase space by taking the time delay copies $\{F(t), F(t+\delta), F(t+2\delta)\}$. In theory, the reconstructed space of $(2k+1)$ dimensions or higher is guaranteed to be embedded, meaning that topological structures have an injective mapping to the underlying phase space which is typically unobserved, with a sufficiently long time series [11]. The first step (Figure 1A to 1B) is necessary before symbolization as a time series in a low dimensional space can be degenerated.

Theoretically, a standard generating partition is a deterministic process that assigns a unique symbol to each data point in the phase space (Figure 1B). In practice, a dataset from the real world may miss some variables or have additional variables independent from the dynamical system. In those situations, it is unclear how to assign a single symbol to a data point in the phase space. With this observation, we relax the theoretical notion of generating partition, and extend it into a probabilistic form with the assumption that the dataset can be generated by a mixture of deterministic and probabilistic processes. More specifically, in the second step of optimization, each data point in a time series is assigned a *probabilistic distribution* of symbols. In Figure 1C, α_{i0} and α_{i1} correspond to the parameters of a probabilistic distribution over symbols "0" and "1". For example, the bottom half and top half of the phase space in Figure 1B are respectively assigned symbol 1 and 0, which indicates the probability of a symbol is higher than that of another symbol. The local patterns in the symbolic subsequences are called a symbol space (boxes with broken lines in Figure 1C). The probabilistic distribution of each data point gives the probabilistic distribution of the symbol set. In Figure 1D, a triplet including the symbols of previous, present and next time points forms a local temporal pattern in the symbol space (e.g., 000, 001, 011 and so forth). A

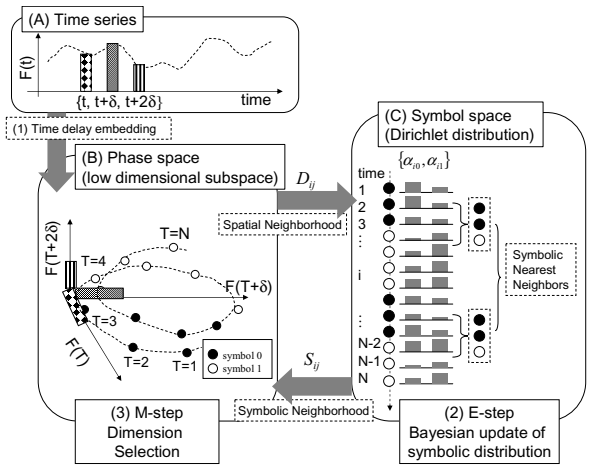

Figure 1. The overview of the Stochastic Dual Nearest Neighbor

set of the subsequences within a given window is called symbolic nearest neighbors (e.g., the pattern 001 is found in the two subsequences). A generating partition is a symbolization process with an optimal inverse mapping from the symbol space to the phase space. Therefore, it optimizes the probabilistic distribution over the symbol set in order to maximize the likelihood of symbolic nearest neighbors given a fixed set of spatial nearest neighbors. In the third step of optimization, it adjusts the spatial configuration of the phase space in order to maximize the likelihood of spatial nearest neighbors given a fixed set of symbolic nearest neighbors.

This whole alternative optimization can be formulated as an EM process [12]: the expectation step here is step 2) – to estimate the likelihood of symbolic nearest neighbors given the current parameters of the phase space; and the maximization step is step 3) – to maximize the likelihood of the spatial configuration given symbolic nearest neighbors. As a whole, it maximizes the likelihood of dual nearest neighbors given latent probabilistic distributions over the symbol set.

IV. Stochastic Dual Nearest Neighbors

This section presents mathematical details of the SDNN. As an overall goal of optimization, we are concerned with the dual nearest neighbor which is a log-likelihood of spatial nearest neighbors averaged with respect to given symbolic nearest neighbors. We define the likelihood of spatial nearest neighbors as a normal distribution capturing a distance between two data points i and j, D_{ij} with a constant variance σ^2. The log-likelihood is $\{-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma}D_{ij}^2\}$. The dual nearest neighbor DNN(W, α), as a function of weights on dimensions W and parameters of symbolic distribution α is defined as follows:

$$\text{DNN}(W, \alpha) = -\frac{1}{2}\sum_{i,j} S_{ij}(\alpha) D_{ij}^2(W) \qquad (1)$$

where $\sum_{i,j} S_{ij}(1 - \delta_{ij}) = 1$ ($\delta_{ii} = 1$ and 0 otherwise) and S_{ij} is the probability of symbolic nearest neighborhood for a pair of data points i and j. In the E-step, the probability of symbolic nearest neighborhood S_{ij} is computed based on a given spatial distance D_{ij}. In the M-step, the expectation of log-likelihood DNN is maximized with respect to the weights on dimension $W = \{w_1, w_2, \ldots, w_K\}$, where K is the dimensionality of the phase space. The maximization of DNN allows us to select a subset of dimensions $\{w_1, w_2, \ldots, w_k\}$ ($k \leq K$) based on how likely the time series on each dimension is described as a deterministic dynamical system.

A. E-step: Bayesian updates of symbol distribution

In the E-step, the expectation of the likelihood of symbolic nearest neighbors is computed for a given set of spatial distances d_{ij} $(i, j = 1, 2, \ldots, N)$. In estimating the probabilistic distribution of symbol i, X_i $(i = 1, 2, \ldots, N)$, we start with a random set of distributions and iteratively update it with respect to the given set of spatial nearest neighborhood. Since the dual nearest neighbor DNN is a function of spatial distances, the symbolic distribution is updated so that its symbolic nearest neighbors are likely to be spatial nearest neighbors. Using Bayes' theorem, with the Dirichlet prior distribution and likelihood of symbolic and spatial nearest neighbors, the posterior distribution is as follows.

$$P(X_i|\text{dual NN}, X_{j\neq i}) = \frac{P(\text{spatial NN}, \text{symbolic NN}, X)}{P(\text{spatial NN}, \text{symbolic NN}, X_{j\neq i})} \qquad (2)$$

1) Prior distribution of a symbol set: Now we assume that the probabilistic distribution of a symbol set assigned to each data point i $(i = 1, 2, \ldots, N)$ follows a Dirichlet distribution with a particular set of parameters $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iM}\}$ $(\alpha_{is} \geq 0)$. Let X_{is} denote a probabilistic variable of the data point i assigned with symbol s and x_{is} $(s = 1, 2, \ldots, M)$ be the probability of symbols $(\sum_s x_{is} = 1)$. Assuming the symbol on a data point is independent from others, the joint probability of $P(X) = P(X_1, X_2, \ldots, X_N)$ is as follows:

$$P(X) = \prod_{i=1}^{N} B(\alpha_i)^{-1} \prod_{s}^{M} x_{is}^{\alpha_{is}-1} \qquad (3)$$

where $B(\alpha_i) = \frac{\prod_{s=1}^{M}\Gamma(\alpha_{is})}{\Gamma(\sum_{s=1}^{M}\alpha_{is})}$ is a normalization term of X_i and $\Gamma(\alpha_{is})$ is the gamma function.

2) Symbolic nearest neighborhood: Symbolic nearest neighbors with a window size τ between i and j $(i \neq j)$ are defined as $X_{i+t} = X_{j+t}$ $(t = -\tau, -\tau+1, \ldots, \tau)$ corresponding to all of the symbols in a symbol subsequence $\{X_t\}$ within the given window size $i - \tau \leq t \leq i + \tau$ and the subsequence within the given window size $j - \tau \leq t \leq j + \tau$. Assuming probabilistic independence among symbol sequences, the probability of correspondence between symbol i and j is $\sum_s x_{i,s}x_{j,s}$. Thus the likelihood of data points i and j being symbolic nearest neighbors which is defined as one-to-one correspondences between paired symbol subsequences is:

$$P(\text{symbolic NN}|X_i, X_j) \propto \prod_{k=0}^{\tau}\left(\sum_s x_{i-k,s}x_{j-k,s}\right) \qquad (4)$$

3) Spatial nearest neighborhood: The conditional probability of spatial nearest neighbors given symbolic nearest neighbors also follows the normal distribution of D_{ij}, the spatial distance between i and j, with mean 0 and variance σ^2.

$$P(\text{spatial NN}_{ij}|\text{symbolic NN}_{ij}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-D_{ij}^2}{2\sigma^2}\right) \quad (5)$$

where D_{ij} is the spatial distance between data points i and j and σ is a hyper parameter for the likelihood of spatial nearest neighbors.

4) Posterior distribution of a symbol set: Using Bayes theorem (Equation 2), the posterior distribution of symbols on the data point i is given as follows. Let P_{is} denote $P(X_i = s|\text{dual NN}, X_{j\neq i})$.

$$P_{is} \propto \frac{\sum_s \left(\frac{\prod_t \Gamma(\alpha_{it}+\delta_{st})}{\Gamma(\sum_t \alpha_{it}+\delta_{st})}\right)^{-1} \prod_t^M x_{it}^{\alpha_{it}+\delta_{st}-1} Q_{is}}{\sum_s Q_{is}} \quad (6)$$

where $Q_{is} = \hat{\alpha}_{is} \sum_j \frac{\exp\left(\frac{-D_{ij}}{2\sigma^2}\right)\alpha_{js} R_{ij}^\tau}{\sum_j \alpha_{js} R_{ij}^\tau}$, $R_{ij}^\tau = \prod_{k=-(\tau-1)}^{\tau-1} \left(\sum_{t=1}^M \hat{\alpha}_{i-k,t}\hat{\alpha}_{j-k,t}\right)$, and $\hat{\alpha}_{js} = \frac{\alpha_{js}}{\sum_s \alpha_{js}}$. Equation (6) indicates that the posterior distribution of symbols is a mixture Dirichlet distribution with the mixture probability $Q_{is}(\sum_s^M Q_{is})^{-1}$. Each Dirichlet distribution in the prior distribution generates M different Dirichlet distributions in the posterior distribution. It is impossible to exactly compute those since the number of variables to be calculated grows exponentially. Therefore we approximate the mixture distribution $P_i(\alpha_{is}, Q_{is})$ with a single prototypical Dirichlet distribution $P_i(\gamma_{is}) \propto \prod_s x_{is}^{\gamma_{is}} \sim P_i(\alpha_{is}, Q_{is})$ with parameter γ_{is}. The details are given in the next section. In the iterative update of the probabilistic distribution, we start with the parameter set $\alpha_{is} = \epsilon$ ($i = 1, 2, \ldots, N$, $s = 1, 2, \ldots, M$) in which a small random positive value ($\epsilon \ll 0$) allows any symbols to occur with a nearly equal probability. The mixture probabilistic distribution with the initial parameter set $P_0(X_i; \alpha_{is})$ gives the approximated distribution $\hat{P}_0(X_i; \gamma_{is})$. Therefore we use Equation (6) to update the parameter set α_{is} by an iterative calculation of the approximated posterior distribution γ_{is} as the prior distribution in the next step $\left(\alpha_{is}^{(0)} \approx \gamma_{is}^{(0)} \equiv \alpha_{is}^{(1)} \approx \gamma_{is}^{(1)} \equiv \ldots\right)$, until it satisfies a given termination condition.

5) Maximum likelihood approximation of mixture Dirichlet distribution: Here we replace the mixture Dirichlet distribution (Equation 6) with a single Dirichlet distribution maximizing the likelihood of the mixture distribution. The following equation gives the log-likelihood of mixture distribution $H(\gamma)$ given the approximating distribution with parameters $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_N\}$.

$$\begin{aligned} H(\gamma) &= \int \log\left(\frac{\prod_s x_s^{\gamma_s - 1}}{B(\gamma)}\right) \sum_j P_j \frac{\prod_s x_s^{\alpha_{js}-1}}{B(\alpha_j)} \prod_s \mathrm{d}x_s \quad (7) \\ &= \sum_s (\gamma_s - 1) B_s - \log B(\gamma) \quad (8) \end{aligned}$$

where $\alpha_j = \{\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jM}\}$ is the parameter set of Dirichlet distribution j and $B_s = \sum_j P_j \{\psi(\alpha_{js}) - \psi(\sum_s \alpha_{js})\}$. The equation on the right hand side is derived with the formula $E[\log(x_s)] = \int \log(x_s) x_s^{\alpha_s - 1} B^{-1}(\alpha) \mathrm{d}x = \frac{\partial \log B(\alpha)}{\partial \alpha_s} = \psi(\alpha_s) - \psi(\sum_s \alpha_s)$, where $\psi(\alpha) = \Gamma(\alpha)^{-1}\Gamma(\alpha)'$ is a Gamma function, The parameter set γ is estimated by the Newton method with the first and second differentials of $H(\gamma)$ with respect to $\hat{\gamma}_s = \log(\gamma_s)$.

B. M-step: Dimension selection

Let $\tilde{S}_{ij}^{(t)}$ denote the likelihood of symbolic nearest neighbor $P(\text{symbolic NN}|X_i, X_j; D_t)$ estimated at step t. The expectation of dual nearest neighbors (Equation 1) is rewritten as: $L(\mathbf{w}) = -\frac{1}{2}\sum_{i,j} \tilde{S}_{ij}\tilde{D}_{ij}^2$ as a function of the linear projection $\mathbf{w} = \{w_1, w_2, \ldots, w_K\}^T$, where $\sum_{i,j} \tilde{S}_{ij}(1-\delta_{ij}) = 1$ ($\delta_{ii} = 1$ and 0 otherwise), and $\tilde{D}_{ij}^2 = \{(\mathbf{y}_i - \mathbf{y}_j)^T\mathbf{w}\}^2$ is a squared distance between i and j projected on $\mathbf{w}$. $\mathbf{y}_i = \{y_{i1}, y_{i2}, \ldots, y_{iK}\}$ is a vector of data point i in the phase space. The likelihood function L is maximized subject to the constant average distance $\frac{\sum_{i,j}\tilde{D}_{ij}^2}{N(N-1)} = 1$, without loss of generality. The linear projection $\hat{w}$ maximizing $L(w)$ subject to the constraint of average distance is given as the following Lagrange equation with a multiplier λ:

$$\hat{L}(\mathbf{w}) = -\frac{1}{2}\sum_{i,j} \tilde{P}_{ij}\tilde{D}_{ij}^2 + \lambda\left(\frac{\sum_{i,j}\tilde{D}_{ij}^2}{2N(N-1)} - 1\right) \quad (9)$$

Since the necessary condition for minimizing the given cost function is that the partial differential with respect to the vector w is zero, $\frac{\partial \hat{L}(\mathbf{w})}{\partial \mathbf{w}} = 0$. It yields the following generalized eigenvalue problem.

$$\left(\tilde{D} - \lambda D\right)\mathbf{w} = 0 \quad (10)$$

where $\tilde{D} = \sum_{i,j} \tilde{S}_{ij}(\mathbf{y}_i - \mathbf{y}_j)(\mathbf{y}_i - \mathbf{y}_j)^T$ and $D = \sum_{i,j}(\mathbf{y}_i - \mathbf{y}_j)(\mathbf{y}_i - \mathbf{y}_j)^T$. The eigenvector of Equation (10) with the minimum eigenvalue minimizes $\mathbf{w}^t\tilde{D}\mathbf{w}$, and it is, thus, the solution for linear projection $\mathbf{w}$. Therefore, the selection of $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ ($\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_k$), yields the minimum distance among those data points that are supposed to be symbolic nearest neighbor (thus maximizes $L(\mathbf{w})$) in the k dimensional subset of the phase space.

V. Case Study 1: Simulated Data

Here we report a case study of the algorithm with multidimensional time series generated by a pre-defined dynamical system. Although the ultimate goal of developing this algorithm is to apply it to real-world data and use it to discover unknown patterns and dynamics embedded in time series, using simulated data is a critical step toward this goal as we need to validate our algorithm with the ground truth before we apply it to unknown datasets.

We use the Ikeda map, which is one of well defined dynamical systems. In previous studies, a clean version of

the Ikeda map [1] has been used to validate the symbolizatio methods based on estimating the generating partition [5] [10]. In the present study, we generated the simulated dat by adding noises. More specifically, On top of the two dimensional time series from the Ikeda map, we adde one additional stochastic time series as the third dimensio in which each data point is independently generated by normal distribution. In total, a simulated dataset is three dimensional $\{X_t, Y_t, N_t\}$ $(t = 1, 2, \ldots, 3000)$ in whic X_t and Y_t are two dimensions from the Ikeda map, an $N_t \sim N(0, \sigma)$ is generated by a random variable from normal distribution with the average variance of the give data $\sigma^2 = \frac{1}{2}\{V(X_t) + V(Y_t)\}$. Note that the algorithm di not know in advance which dimension contains only noises Ten different 3-dimensional time series as described abov were generated randomly.

To measure the accuracy of generating-partition-base symbolization, *the mean distance rank* (MDR) [5] is applie as non-parametric statistics of the spatial distances amon those data points that are supposed to be symbolic neares neighbors. It has been shown that the MDR is correlated t the topological entropy of a dynamical system which is ofte difficult to calculate. Since a theoretical generating partitio minimizes the topological entropy, the MDR can be viewe as a substitution of topological entropy.

We first run SDNN with only the E-step (but without th M-step) to test the Bayesian update of the symbol distribu tion without optimizing the phase space. The average MDI across ten datasets is 0.161 after the algorithm converges. A typical optimization process is shown in Figure 2A, B, and C. One of the two symbols (indicated by red or green) is assigned to each data point. The ideal symbolization should have the upper half of the dataset with one color and the bottom half of data points with the other color. However, the estimated symbolic results (Figure 2C) have an unclear boundary in the middle of two attractors. This result shows that even one additional noisy series significantly distracts the estimation of an underlying dynamical system.

Next, we run the full version of SDNN, including not only optimizing symbolic nearest neighbors, but only searching and selecting better dimensions in the phase space. Figure 2X shows a typical optimization process of SDNN using the M-step on the noisy time series. The leftmost panel shows the two-dimensional projection of the original dataset assigned with random symbols. The two-dimensional projections are obtained by the principal component analysis (PCA). All the three dimensions have similar variances and small correlations, and thus the PCA projection shows nearly an equal-mixture of three dimensions. In Figure 2Y, the algorithm finds some spatial configuration by rotating and selecting the original three dimensions, and generates a better symbol set. Finally, on the right panel in Figure 2Z, the algorithm estimates the symbol set (MDR=0.005), which is significantly better than the previous results. In fact, this finally estimated spatial configuration (note: the rotation of the coordinates does not affect the result) is very similar to the original Ikeda map. This result suggests that SDNN is able to encode the time series generated by a dynamical system from a heterogeneous dataset by excluding irrelevant dimensions.

[1]The Ikeda map is given as follows: $z_{n+1} = p + Rz_n \exp\left(i\kappa - \frac{i\alpha}{1+|z_n|^2}\right)$ where $p = 1$, $R = 0.9$, $\kappa = 0.4$, $\alpha = 6$ are standard parameters, and i and z_n are imaginary and complex number of n-th point.

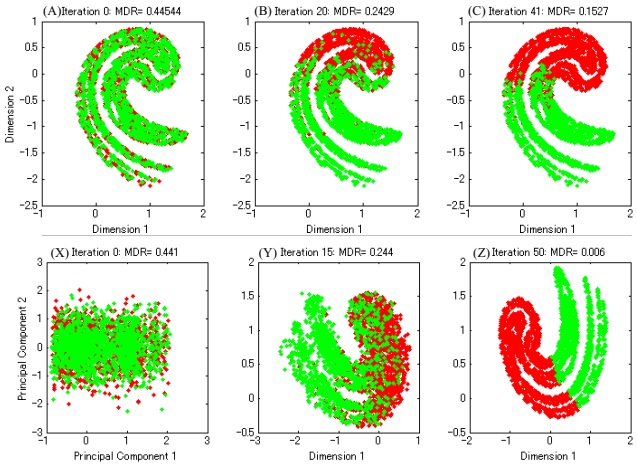

Figure 2. An optimization process of SDNN without M-step on the Ikeda map with a noise dimension (top three panels, A, B, and C), and an optimization process of SDNN with both the E-step and the M-step on the Ikeda map with a noise dimension. The spatial configuration is also optimized (bottom three panels, X, Y, Z).

VI. Case Study 2: Measuring Temporal Dynamics in Face Tracking Data

To our knowledge, previous studies on generating-partition-based symbolization have focused on mathematical simulation [5] but this approach has never been applied to complex multidimensional, time series collected from the real world with potentially various dynamic properties, as well as various levels of noise. The two major goals of this study are to show how SDNN works with real-world data and to demonstrate that the symbolic representation obtained from SDNN can be used to perform various pattern discovery tasks, such as pattern recognition and dimension selection/reduction.

A. Data

The dataset used here consists of 7 video clips. As shown in Figure 3, a person in the video facs the audience and demonstrates some actions. We selected this dataset of everyday video clips for two reasons. First, the patterns derived from SDNN can be easily perceived and understood compared with the results from other datasets collected in specific domains and from specific devices. Second, with the popularity of multimedia techniques, especially in mobile devices, this kind of video clips is ubiquitous. Therefore, multimedia data mining techniques to discover and retrieve user-interested patterns from such data also have applied utilities.

We first used the faceAPI package (www.seeingmachines.com/product/faceapi/) to automatically extract 38 landmarks from a speaker's face in a video clip. Each landmark includes x and y coordinates for a specific face landmark on the face image.

Figure 3. Snapshot of face tracking data

In total, 76 time series derived from each video clip are analyzed using SDNN.

Figure 4 shows a snapshot of the 38 facial landmarks which consist of those key feature points around two eyes, eyebrows, the nose, and lips. In analysis, each of 76 time series is normalized to have zero average and a standard deviation across the whole time series. As a result of image processing, each 76-dimensional time series consists of 600 to 1200 data points, depending on the length of a video clip. Moreover, we manually coded two frequent events in the video clips – talking and smiling. This coding was performed frame by frame, so that we obtained the ground truth of a person's facial state frame by frame.

B. Symbolization

We applied SDNN to the time series of facial landmarks, and extracted probabilistic symbolic representations. Specifically, we fixed the number of symbols to 4, and thus, each 76-dimensional data point in a raw time series was assigned with four probabilities, one for each symbol. In implementation, the hyper-parameter of symbolic representation is $\tau = 1$, and the parameter of spatial nearest neighbors is $\sigma = \frac{1}{2}$ (Note that the standard deviation is normalized to be 1 in each dimension). The linear weights ($\mathbf{w}$ in Equation 10) are used as a measure of the dynamical structure of 76 temporal variables. SDNN estimates 76 linear weights ($\{\mathbf{w}_1, \ldots, \mathbf{w}_{76}\}$), and we chose the first 50 weights out of 76 which have shorter distances among those data points, and treated them as symbolic nearest neighbors. Thus, an analysis of the linear weights suggests that the variables similar in the linear weights would be involved with similar dynamical processes.

C. Pattern Recognition of Certain Events in Time Series

One of pattern discovery tasks from time series is to detect certain events in a temporal data stream. In this section, we use the probabilistic symbolic representation from SDNN to detect certain patterns, and compare the results with those obtained from using other symbolization approaches.

As mentioned earlier, two frequent events in the video clips are talking and smiling. Therefore, we focused on classifying those two events in the data. Two standard classifiers are used for classification – logistic regression and decision tree. The goal here is to feed those classifiers with different data as a way to compare our symoblic representation against other representations. To do so, we use cross-validation, and randomly splitted the dataset into half of the training data and half of the testing data. The following reported results are based on ten trials of independently and randomly chosen training and testing datasets. Also note that those two types of events are not mutually exclusive – at some moments, a person may talk while s/he is smiling. To test the robustness of our approach, we used the same representation in the classification of two events.

Three symbolization approaches are compared:

- Probabilistic Symbolic representation from SDNN: each 76-dimensional raw data point in a time series is converted to a 4-dimensional probability representation as each probability indicates the likelihood that the raw data point is assigned to one of the 4 symbols.
- Symbol sequences by SAX: There are multiple highly correlated dimensions in the dataset and thus the symbols estimated by SAX on these variables become the identical symbol series. These identical symbols cause a technical problem in classification. Therefore, we removed these highly correlated variables using Principal Component Analysis (PCA). The PCA decomposes the covariance matrix of the data, and we chose dimensions with a larger variance until the chosen set holds 99% of the variance in the original dataset. We then estimated the symbol series on the PCA dimension described above instead of the original dataset.
- Symbol sequences by SFNN ([4], [5]): we also used the original approach of generating partition-based symbolization. This approach does not include Baysian updates, nor the E-step and the M-step in SDNN.

Table I summarizes the accuracy of classification results with two classifiers, three symbolic representations, and two event types. The bold letters show the best performance across data coding (SDNN, SFNN, and SAX) and classifier (Logistic regression and Decision tree). For all the three data coding, the decision tree is a better classifier for both training and test sets, and therefore we refer to these results for comparison. Indeed, the results based on SDNN using the decision tree as a classifier outperforms the other conditions in both the training and test sets.

	Classifier	Logistic		DecisionTree	
	Methods	Smiling	Talking	Smiling	Talking
Test	SDNN	0.8327	0.6242	**0.9214**	**0.896**
	SFNN	0.4558	0.4833	0.4455	0.4855
	SAX	0.7659	0.6286	0.8028	0.6774
Training	SDNN	0.8344	0.6335	**0.98**	**0.9718**
	SFNN	0.4579	0.4879	0.4492	0.4888
	SAX	0.7755	0.6458	0.8508	0.7359

Table I
ACCURACY IN CLASSIFICATION OF TALKING AND SMILING EVENTS. EACH CORRECT RATIO IS AVERAGED ACROSS 10 DIFFERENT TRAINING AND TEST SETS SAMPLED RANDOMLY.

D. Dimension selection: What information is encoded in symbols?

We found so far that the symbolization results from SDNN can extract useful information for pattern classification.

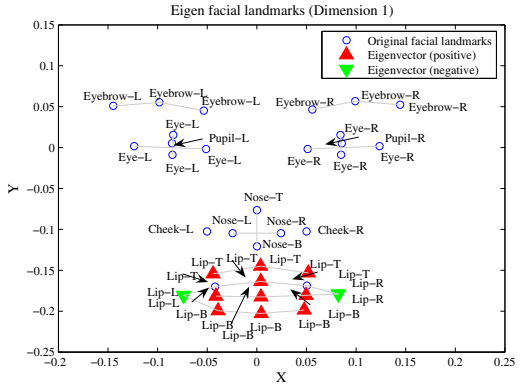

Figure 4. Eigen facial landmarks of the first dimension

Why does SDNN do so well? One advantage of using SDNN is that it provides which dimensions may be of interest, since it explicitly selects dimensions in the course of the symbolization process. To better understand what information is coded in the symbol sequences obtained by SDNN, we map numerical values in the eigenvector of the first dimension, in which the symbolic nearest neighbors have the shortest spatial distances, onto a snapshot of facial landmarks. Several facial landmarks have the large values (positive or negative)in the first selected dimension, as shown in Figure 4. One obvious pattern is that those facial landmarks with large absolute values are all from the lip area. Thus, the first selected dimension by SDNN, which is supposed to contain the richest information in the time series, encodes the motion of lips. Based on our common knowledge, we know that those landmarks should be tightly related to both smiling and talking events. However, the significant results here lie in the fact that SDNN discovers this pattern/knowledge by itself from the data without any prior information. In fact, this dimension selection shows a true utility of SDNN – it is not just a representation that can lead to better classifcation results. More importantly, it has potentials to be used to discover new knowledge from the dataset in a unsupervised manner.

VII. Conclusions

This work proposes a new symbolization algorithm for finding latent dynamical properties in multidimensional time series. From the algorithmic perspective, our method relies on the generating partition which theoretically characterizes essential properties of a given dynamical system. Unlike the previous versions of generating partitions, SDNN is robust to noise and suitable for the application to a dataset from the real world with unknown noise, due to its two inherent properties. First, SDNN offers a Bayesian framework for symbol dynamics. It enables us to access symbol dynamics in a general form. Second, the algorithm relies on both spatial and temporal information encoded in multidimensional time series. In this way, instead of viewing high dimensionality as a problem, this method tries to take advantage of higher dimensional data to capture more statistical regularities in symbolization. Moreover, our EM-based training provides a natural probabilistic framework to integrate those two sources of information in multidimensional time series. With two case studies of both simulated and real-world data, we suggest that SDNN based on a generating partition has the potential to be used as a way to analyze multidimensional time series.

From the engineering perspective, SDNN has one compelling characteristic. That is, it can be used in various pattern discovery and data analysis tasks. In order to accomplish those tasks, most often we have to develop several data mining algorithms, and each of them focuses on one task with its own data preprocessing routine and its own data representation. Probabilistic symbolic sequences derived from SDNN provide an intermediate representation of the data further data mining as this representation contains more information than the straightforward discrete symbol sequences and simultaneously it maintains an abstract form to facilitate further computations (compared with raw data). In future work, we plan to further evaluate SDNN with more real-world data and with other knowledge discovery tasks.

References

[1] 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, pp. 107–144, 2007.

[2] L. Wei, N. Kumar, V. N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Assumption-free anomaly detection in time series," in *Proceeding of the 17th International Scientific and Statistical*, 205.

[3] J. Lin, E. Keogh, S. Lonardi, and P. Patel, "Finding motifs in time series," in *Workshop notes of the 2nd workshop on temporal data mining at the 8th ACM international conference on knowledge discovery and data mining*, 2002.

[4] M. B. Kennel and M. Buhl, "Estimating good discrete partitions from observed data: Symbolic false nearest neighbors," *Physical Review Letters*, vol. 91, no. 8, p. 084102, 2003.

[5] M. Buhl and M. B. Kennel, "Statistically relaxing to generating partitions for observed time-series data," *Physical Review E*, vol. 71, no. 4, p. 046213, 2005.

[6] G. B. Folland and A. Sitaram, "The uncertainty principle: a mathematical survey," *The Journal of Fourier Analysis and Applications*, vol. 3, no. 3, pp. 207–238, 1997.

[7] K. Shockley, *Tutorials in Contemporary Nonlinear Methods for the Behavioral Sciences*. Retrieved December 1, 2009, from http://www.nsf.gov/sbe/bcs/pac/nmbs/nmbs.jsp, 2005, ch. Cross recurrence quantification of interpersonal postural activity.

[8] T. Schreiber, "Interdisciplinary application of nonlinear time series methods," *Phys. Rep*, vol. 308, pp. 1–64, 1998.

[9] C. S. Daw, F. C. E. A., and E. R. Tracy, "A review of symbolic analysis of experimental data," *Review of Scientific Instruments*, vol. 74, no. 2, pp. 914–930, 2003.

[10] M. B. Kennel and H. D. Abarbanel, "False neighbors and false stands: A reliable minimum embedding dimension algorithm," *Physical Review E*, vol. 66, no. 2, p. 026209, 2002.

[11] F. Takens, *Lecture Notes in Mathematics*. Springer-Verlag, 1981, vol. 898, ch. Detecting strange attractors in turbulence.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of Royal Statistical Society Series B*, vol. 39, pp. 1–38, 1977.