Analyzing Multimodal Time Series as Dynamical Systems [*]

Shohei Hidaka
Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa, Japan
shhidaka@jaist.ac.jp

Chen Yu
Indiana University
1101, 10th Street, Bloomington, IN 47405
chenyu@indiana.edu

ABSTRACT

We propose a novel approach to discovering latent structures from multimodal time series. We view a time series as observed data from an underlying dynamical system. In this way, analyzing multimodal time series can be viewed as finding latent structures from dynamical systems. In light this, our approach is based on the concept of generating partition which is the theoretically best symbolization of time series maximizing the information of the underlying original continuous dynamical system. However, generating partition is difficult to achieve for time series without explicit dynamical equations. Different from most previous approaches that attempt to approximate generating partition through various deterministic symbolization processes, our algorithm maintains and estimates a probabilistic distribution over a symbol set for each data point in a time series. To do so, we develop a Bayesian framework for probabilistic symbolization and demonstrate that the approach can be successfully applied to both simulated data and empirical data from multimodal agent-agent interactions. We suggest this unsupervised learning algorithm has a potential to be used in various multimodal datasets as first steps to identify underlying structures between temporal variables.

Keywords

Multi-stream time series, multi-agent communication, symbol dynamics, generating partition

1. INTRODUCTION

One of the ultimate goals of studying multimodal interfaces and multimodal interactions is to build intelligent agents (e.g. robots or virtual avatars) that can smoothly interact with human users through coupled multimodal behaviors. Achieving this goal relies on both theoretical breakthroughs and also new findings from empirical studies. With advances in computing and sensing technologies, an unprecedented amount of data has been collected in various human-computer interaction studies, including human-robot interactions, human-avatar interactions, and human-computer interactions through mobile devices. Such rich data from multimodal interactions provide an unique opportunity to systematically analyze behavioral data from human users which has the potential to lead to breakthroughs at several frontiers in human-computer interactions. First, the results from data-mining multimodal user data can be directly used to improve current multimodal interfaces. Second, the patterns derived from such data can lead to a better understanding of the fundamental principles of multimodal communication that may lead to theoretical advances. Third, this data-driven approach will provide a much more objective way to evaluate a multimodal interface. Compared with surveys or questionnaires, analyzing real-time behavior patterns directly measured from human users can better reveal user's reactions and preferences.

While there is probably no doubt that machine learning and data mining techniques are playing more and more important roles in studying intelligent mulitmodal interactions and interfaces, a particular challenge in this new data-driven venue though is how to effectively data-mine temporal sequences and successfully deal with both individual data streams changing over time, and multimodal synchrony and correlation between those streams. More specifically, multiple temporal streams may exhibit different kinds of dynamics with patterns changing from one moment to the next. Probably because of this, most existing algorithms rely on a particular kind of prior knowledge in temporal data mining. For example, sequence matching needs the user to provide a query pattern to start [1, 2]. Motif discovery (or anomaly detection) needs to be based on the assumption that a pattern is either frequent or less frequent in a given dataset [3]. For another example, Hidden Markov Models [4] and Markov random fields [5] have been widely used in various speech, text and image datasets. However, those approach work well with temporal data with certain structures that satisfy the Markov Properties. It is not clear on how extend HMMs and it variants [6, 7] to more stochastic data.

The present paper introduces a new temporal data mining algorithm without a need of any preassumption of multistream time series. Our approach is motivated by the idea of dynamical system by viewing a time series as observed data from a dynamical system. Accordingly, multimodal time series can be viewed as analyzing underlying latent structures of multimodal dynamical systems. The present paper will first introduce our approach and then provide two case studies (one with simulated data and one with real-world data from multimodal interaction) to demonstrate the potential of this approach as a general approach for a wide range of multimodal data.

2. GENERATING PARTITION AS SYMBOLIZATION

We argue that one of the important directions in temporal data mining is to convert a time series into a symbolic sequence. By so doing, a discrete representation opens up many powerful techniques of 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 benefits of a low-precision symbolic representation and meanwhile minimize the loss of information in the symbolization process. Many symbolization approaches in data mining are based on the histogram distribution of a time series. For example, in Symbolic Aggregation approximation (SAX, [2]), 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. One advantage of SAX is that the symbolization process approximates the Euclidian 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. Recently a compelling technique called Symbolic False Nearest Neighbor (SFNN) has been developed in theoretical physics and it has been demonstrated that this symbolization approach has various advantages compared with histogram-based approaches (e.g. SAX [2]). In the present paper, we suggested that the original version of SFNN is a deterministic algorithm which limits its performance to tolerate noises in the data. In light of this, we developed a probabilistic algorithm with Bayesian updates. In the following, we will first introduce the SFNN and the concept of generating partition. Next, we will present a conceptual framework of our new algorithm called stochastic Dual Nearest Neighbor (SDNN), followed by a detailed description of the algorithm. Three simulation experiments will then be reported to evaluate the performance of SDNN.

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. 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 [10, 11]; 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 heterogeneous time series. Different from the assumption most often used in theoretical simulations, a dataset from the real world is unlikely to 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 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 for 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 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.

2.1 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. Suppose that we have a one-dimensional time series (Figure 1A). The first step for analyzing such 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 [12]. 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 sufficient long time series [12]. The first step (Figure 1A to 1B) is necessary before symbolization as a time series in a low dimensional space can be is degenerated.

Theoretically, a standard generating partition is a deterministic process that assigns an 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 to the focal dynamical system. In those situations, it is unclear how to assign a signal sym-

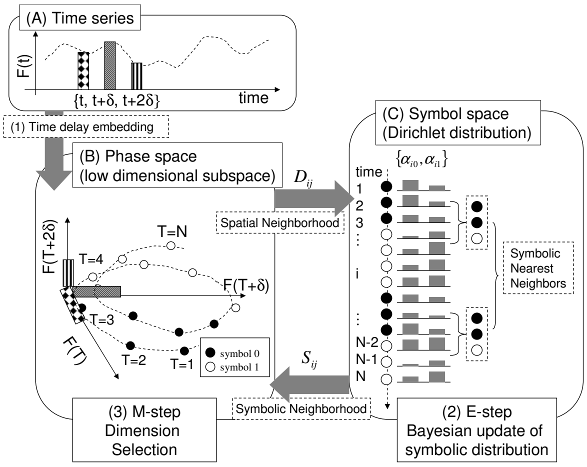

Figure 1: The overview of the Stochastic Dual Nearest Neighbor

bol 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 in previous, present and next time points forms a local temporal pattern in the symbol space (e.g., 000, 001, 011 and so forth). A set of the subsequences within a given window is called symbolic nearest neighbors (e.g., the pattern 001 are 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 given a fixed set of spatial nearest neighbors. In the third step of optimization, it adjusts the spatial configuration of 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 [13]: 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 the whole, it maximizes the likelihood of dual nearest neighbors given latent probabilistic distributions over the symbol set.

3. STOCHASTIC DUAL NEAREST NEIGHBORS

This section presents mathematical details of the SDNN. As an overall goal of optimization, we concern 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 $\mathrm{DNN}(W, \alpha)$, as a function of weights on dimensions W and parameters of symbolic distribution α is defined as follows.

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

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

3.1 E-step: Bayesian updates of symbol distribution

In E-step, the expectation of the likelihood of symbolic nearest neighbors is computed for a given set of spatial distances d_{ij} $(i, j = 1, 2, \dots, N)$. In estimating the probabilistic distribution of symbol i, X_i $(i = 1, 2, \dots, N)$, we start with a random set of distribution 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.

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

3.1.1 Prior distribution of a symbol set

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

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

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.

3.1.2 Symbolic nearest neighborhood

Symbolic nearest neighbors with a window size τ between i and j $(i \neq j)$ are defined as $X_{i+t} = X_{j+t}$ $(t = -\tau, -\tau + 1, \dots, \tau)$ 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 that 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 $\sum_s xi, sx_{j,s}$. 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:

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

3.1.3 Spatial nearest neighborhood

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

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

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.

3.1.4 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 $P(X_i = s|\text{dual NN}, X_{j\neq i})$.

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

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 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 prior distribution generates M different Dirichlet distributions in the posterior distribution. It is impossible to exactly compute 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 detail is 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.

3.1.5 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 $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_N\}$.

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

where $\alpha_j = \{\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jM}\}$ is the parameter set of Dirichlet distribution j and $B_s = \sum_j P_j \left\{ \psi(\alpha_{js}) - \psi\left(\sum_s \alpha_{js}\right) \right\}$. The equation on 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 $\hat{\gamma}_s = \log(\gamma_s)$.

3.2 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 neighbor (Equation 1) is rewritten: $L(\mathbf{w}) = -\frac{1}{2} \sum_{i,j} \tilde{S}_{ij} \tilde{D}_{ij}^2$ as function of the linear projection $\mathbf{w} = \{w_1, w_2, \ldots, w_K\}^T$ where superscript T indicates transposition, $\sum_{i,j} \tilde{S}_{ij}(1-\delta_{ij}) = 1$ ($\delta_{ii} = 1$ 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 a Lagrange equation with a multiplier λ:

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

Since the necessary condition for optimal $\mathbf{w}$ 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 gives the following generalized eigenvalue problem.

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

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 $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ $(\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_k)$ makes the minimum distance among the data points supposed to be symbolic nearest neighbor (thus maximizes $L(\mathbf{w})$) in k dimensional subset of the phase space.

4. CASE STUDY 1: SIMULATED DATA

Here we report three simulations with time series generated by a pre-defined dynamical system. Although the ultimate goal of developing this algorithm is to apply it with mulitimodal datasets to discover unknown patterns and dynamics, 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.

The first simulation validates E-step, the estimation of an expected symbol distribution, without M-step (the optimization of the spatial configuration). This is a special case of SDNN when we assume that a given time series is perfectly deterministic without a need to optimize the phase space. The second simulation concerns a mixture of time series in which some subset is generated by a dynamical system and others are from a probabilistic process. In the second simulation, we demonstrate that only E-step is not sufficient to handle the mixture of time series. The third simulation demonstrates M-step combined with E-step can properly select dimensions of the focal dynamical system and eliminate the noisy dimensions from the probabilistic process. Through out all three simulation, we set hyper parameters of symbolic nearest neighbor $\tau = 2$ and spatial nearest neighbor $\sigma = \frac{\bar{\sigma}}{2}$ where $\bar{\sigma}^2$ is average variance of given dataset.

4.1 Estimating symbol series for the dynamical system

In order to validate SDNN, we estimate a symbol series from a simulated dynamical system. We use Ikeda map which is one of well known dynamical systems. In previous studies, Ikeda map [1] is used to validate the method estimating generating partition [10, 11]. mean distance rank (MDR) [10, 11] is applied as non-parametric statistics of the spatial distances among those data points that are supposed to be symbolic nearest neighbors. It has been showed that MDR is correlated to the topological entropy of a dynamical system which is often difficult to calculate. Since a theoretical generating partition minimizes the topological entropy, we validate our algorithm to check whether it can minimize MDR as a substitution of topological entropy. In this simulation, we use datasets generated by the dynamical equation without noise for simplicity of the validation of the algorithm. Thus, without the optimization of the spatial configuration (M-step), we focus on validating the Bayesian update of the symbol distribution for a given fixed dataset.

Figure 2 shows one example of the Bayesian update of symbol distribution in which one of the two symbols (colored in either green or red) with a larger parameter α_{is} is assigned on each data point. In iteration 0, the algorithm starts with initial parameters with random values $\alpha_{is} \sim 0.1 + 0.05U(0, 1)$ where $U(0, 1)$ is an uniform distribution between 0 to 1. In iteration 14, in the middle of optimization, it gradually colored upper and lower region of the attractors with green and red respectively, but the boundary is not clear yet. In iteration 39, as the end of optimization which satisfies $|\text{MDR}_{t+1} - \text{MDR}_t| \leq 10^{-8}$, the upper and lower half are colored in green and red, and its boundary is clear in the middle. In fact, the estimated partition is very close to the ground truth as indicated by MDR (the bottom plot). In most of our simulations, it converges the similar partition in 30 to 50 iterative steps regardless of their initial values.

4.2 Estimating a generating partition for a dynamical system with noise

Next we demonstrate the application of E-step SDNN for a dataset with a noisy dimension. Again each set of data is generated by Ikeda map with different initial values. On top of the two dimensional time series from Ikeda map, we added one additional stochastic time series in which each data point is independently generated by a normal distribution. The simulated dataset is three-dimensional $\{X_t, Y_t, N_t\}$ $(t = 1, 2, \ldots, 3000)$ in which X_t and Y_t are two dimensions from Ikeda map, and $N_t \sim N(0, \sigma)$ is generated by a random variable from a normal distribution with the average variance of the given data $\sigma^2 = \frac{1}{2}\{V(X_t) + V(Y_t)\}$. Due to an additional random variable, spatial distances are different from those in the original Ikeda map. With the 10 different datasets with a noisy dimension, we run SDNN E-step in order to estimate the symbol series on the noise-contaminated dataset. The average MDR is 0.161. A typical optimization process on the Ikeda map with a noisy dimension is shown in Figure 3. Overall, the estimated symbol sets have a unclear boundary on the middle of attractors (compare it with the right panel in Figure 2). This result clearly shows even one additional noisy series significantly distracts the estimation of a standard generating partition.

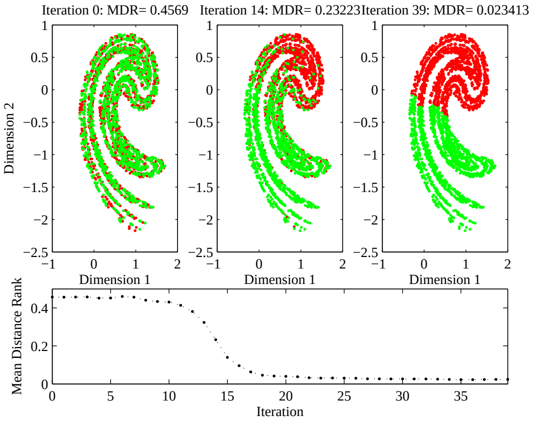

Figure 2: An optimization process of SDNN without M-step on Ikeda map.

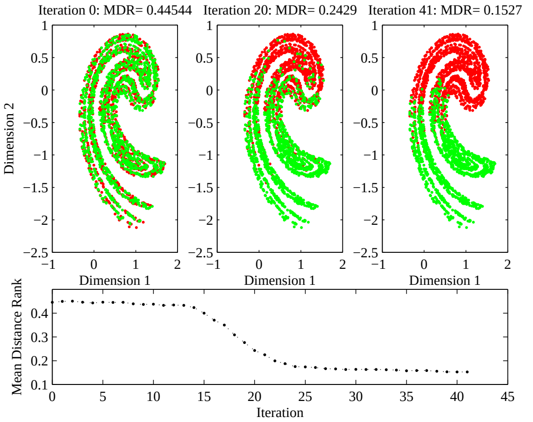

Figure 3: An optimization process of SDNN without M-step on Ikeda map with a noise dimension

4.3 Dimension selection for a dynamical system

With the dataset from the real world, it is unlikely that we know in advance which dimension would be more or less informative beforehand. The following simulation is motivated by such supposed heterogeneous time series which requires a selection of latent dynamical structures in noisy time series. Here we intend to demonstrate dimension selection on the same time series used before, 2-dimensional Ikeda map with a noisy dimension. Note that the estimation in

[1] 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.

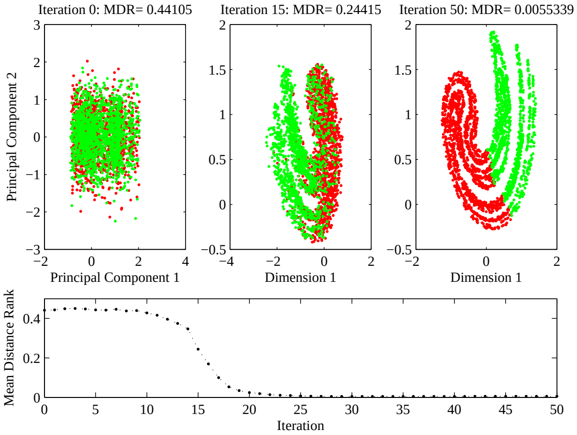

Figure 4: An optimization process of SDNN with both E-step and M-step on Ikeda map with a noise dimension. The spatial configuration is also optimized.

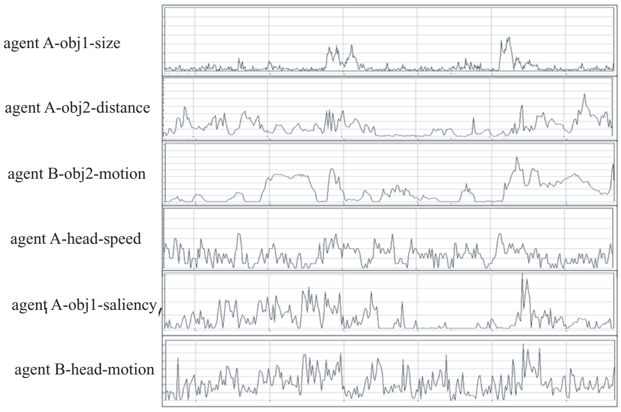

Figure 5: Examples of multi-stream continuous time series from agent A and B.

the previous simulation uses only E-step with a fixed spatial configuration. In this simulation, not only symbols but also the spatial distances are dynamically and simultaneously estimated. Figure 4 shows a typical optimization process of SDNN using M-step on the noisy time series. The most left panel shows the two dimensional projection of the dataset assigned with random symbols. The two dimensional projections are obtained by 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. On the middle panel in Figure 4, the algorithm found some spatial configuration by rotating and selecting the original three dimensions, and generate a better symbol set. Finally, on the right panel, the algorithm estimates the symbol set (MDR=0.005) as good as or even better than the estimation on the dataset without noise (MDR=0.023; Figure 2). At the same time, the finally estimated spatial configuration (note: the rotation of the coordinates does not affect the result) is very similar as the original Ikeda map. This result suggests that SDNN is able to select the time series generated by the dynamical systems from a heterogeneous dataset by removing irrelevant dimensions.

5. CASE STUDY 2: MEASURING TEMPORAL DYNAMICS IN MULTIMODAL COMMUNICATION

5.1 Multimodal data

One of our primary motivations for investigating temporal data mining is to use it to measure adaptive behaviors and temporal dynamics in human-human and human-robot interaction. We collected multimodal data in such interactions in which two agents (adults, children, or robots) were asked to jointly accomplish a task (e.g. a human user teaches a robot a set of object names, or a parent teaches his/her child how to recognize and name a set of toys). Thus, both agents jointly coordinate their behaviors to maintain a smooth interaction. The goal here is to discover the characteristics of coordinated behaviors at the multimodal sensorimotor level. Our multimodal data include video streams from up to 6 cameras recorded simultaneously with a frequency of 30 frames per second, speech, body movement data captured from a motion tracking system, and gaze data from an eye tracker. We developed various tools to automatically pre-process data and derive various time series (e.g. the location of a particular object/person over time from a particular video stream, the movement trajectories of the head over time, the visual saliency of objects/people in a video stream). Technical details of our data processing approach can be found in [14, 15]. As a result, we have extracted multiple time series measuring both perceptual and action data from interacting agents. From such data, the goal of this research is to discover the fundamental principles at the sensorimotor level that lead to smooth human-human and human-robot interaction. More specifically, within an agent, what one visually perceives determines what action would be taken next (e.g. usually we generate an eye fixation on the target object before we start to reach for the target). Similarly, the current action from the agent determines what the same agent will perceive next (e.g. a head movement would switch the agent's visual field). Meanwhile, across two agents, what one agent perceives also depends on the other agent's actions. The goal of our research is to find perception-action dynamics and dependencies from those multiple sensory and action time series. Our first effort is to quantify the overall temporal dynamics between multimodal time series.

To our knowledge, previous studies on the generating partition focused on mathematical simulation but this approach has never been applied to complex heterogeneous time series collected from the real world with potentially various dynamic properties and as well as various levels of noise. Figure 5 shows an example of 6 time series used in this study from which we cannot easily spot any patterns as those temporal profiles vary from one moment to another moment and apparently in different ways. In total, we used 71 time series like the ones shown in Figure 5, each containing from several hundreds to over a thousand data points. Again the temporal streams are derived from raw multimedia data, describing various perception and action behaviors generated by two agents in a face-to-face interaction. From those micro-level behavioral time series such as head motion, or visual patterns from multiple cameras from different viewpoints, the goal here is to reconstruct the physical and social dynamics behind the time series that captures characteristics in the interaction.

71 time series derived from multimodal interaction are analyzed using SDNN. The labels of 71 variables are classified with a combination of three terms: agents, objects, and sensor types. The two agents A and B (denoted as “A” and “B” respectively) are shown as prefix of a label. Objects including object 1, 2, 3, hands, head, naming target, and non-naming target (denoted with abbreviations O1, O2, O3, Hnd, Hd, T, and N respectively) are shown as the middle

part of a label. Sensor types including the size of an object in a camera's view, the temporal difference of object size in view or the speed of object, the distance of object from the center of view, the saliency (based on low-level visual features such as motion, orientation, and intensity) of object in view (denoted with abbreviations Sz, Spd, Dst, and Sal respectively) are shown as the suffix of a label. For example, "A-O1-Sz" indicates the temporal variable coding the size of object 1 from Agent A.

We first apply the logarithm of the original data and normalize the results to have 0 as a mean and 1 variance across time. The hyper parameter of symbolic is $\tau = 1$, and that 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 71 temporal variables. SDNN estimates 71 linear weights ($\{\mathbf{w}_1, \ldots, \mathbf{w}_{71}\}$), and we chose the first 50 weights out of 71 which have shorter distances among those data points supposed to be 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.

5.2 Results

The data contain 1029 time points of 71 variables. The patterns of linear weights, indicating the dynamical properties of variables, are visualized using Multidimensional scaling (Figure 6A). As a comparison, we also analyze the raw time series and the results are shown in Figure B. In both Figure 6A and 6B, the distances between points approximate the cosine of two variables in a higher dimension (either distance 50 dimensional eigenvector space or distance in the raw time series).

However, the ways those time series are clustered are different. Figure 6A shows two different types of clusters. one type of clusters seems to be agent-based, containing the temporal variables derived from only one of the two agents, and the other type appears to be data-type-based, containing temporal variables from both agent A and B but sharing the same data types. Specifically, on both the top and the bottom in Figure 6A, several variables from agent A and B (e.g., "A-O1-Spd", "A-O2-spd", "A-O3-Spd", etc., those variables coding the motion of three objects in agent A's view form a cluster at the bottom) are visulized as separable clusters. On the other hand, on both the left and the right side in Figure 6A, temporal variables from two agents are grouped into the same clusters with shared sensor types. For example, the distances of multiple objects from agent A's or B's camera view (e.g., "A-O1-Dst" and "B-O2-Dst") are close to each other. Those variables coding the distances of objects from a camera's view indicate which object that agent wearing the head-amount camera is attending at a moment. Based on a theoretical assumption, the variables with similar linear weights in SDNN are supposed to share similar dynamical properties. Thus, our results show that different variables coding what agent A and B are looking are identified as sharing similar dynamical properties. Based on this result, we can futher infer that agents A and B dynamically adjust their behaviors to build and maintain joint attention of the same object. For anexample, the motion saliency of the same object either in agent A's view or B's view reveals different dynamical properties, while the motion saliency of different objects in the same agent's view shares similar dynamical properties. This pattern reflects the physical setting in the experiment – multiple objects captured from the same camera may block each other and thefore co-vary over time. In sum, SDNN analysis can capture both types of temporal correlations, one from the physical setting of experiment and the other from social interaction between two agents.

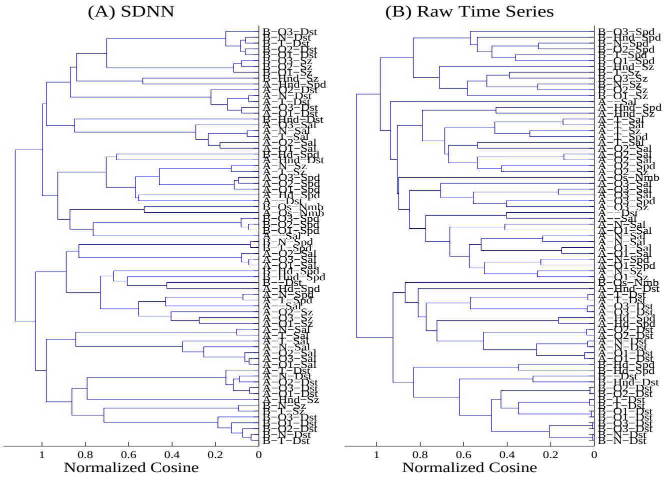

Figure 7: Hierarchical clustering of (A) SDNN and (B) raw time series

This observation is also supported by hierarchical clustering of the linear weights from SDNN (Figure 7A). In those small clusters within shorter distances, we found the variables coding similar physical properties (e.g., "B-O1-Dst", "B-O2-Dst", "B-O3-Dst" on the bottom of Figure 7A), but, in the next level of clustering with longer distances, the similar sensory variables from two agents (e.g., "B-O*-Dst", "A-O*-Dst" are next to each other on bottom of Figure 7A) are grouped together.

Meanwhile, the comparable analysis on raw time series, by taking cosine of two time series as metric, show very different patterns (Figure 6B and Figure 7B). It fails to detect any between-agent clusters and captures only the overall physical setting of sensors: variables from one agent tend to be similar in both MDS visualization and hierarchical clustering regardless of sensor types. This result suggests that the analysis on raw time series may not be able to capture deeper correlations of "social interaction" between agent A and B, but only some simple correlations at the surface. We already know that those variables derived from Agent A are generally correlate with each other. The goal of analyzing multimodal interaction data is to discover and quantify multimodal measures between interacting agents.

The present case study using time series data derived from multimodal agent-agent interactions demonstrates that SDNN successfully captures social interactions between two agents without any prior knowledge on experimental settings such as types of sensors used or the fact that two agents are communicating. Since our goal is to pursue a deeper understanding of multimodal communication between two agents, we have to go beyond what we have already know. SDNN as an unsupervised machine learning technique shows a promise that may suit for this purpose well by providing objective measures and analyses on multivariate time series.

6. CONCLUSION

The present study proposes a new symbolization algorithm for finding latent dynamical properties in heterogeneous time series. The algorithm relies on generating partition which theoretically characterizes essential properties of a given dynamical system. Unlike the previous versions of generating partition, SDNN is robust to noise and suitable for the application for the 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

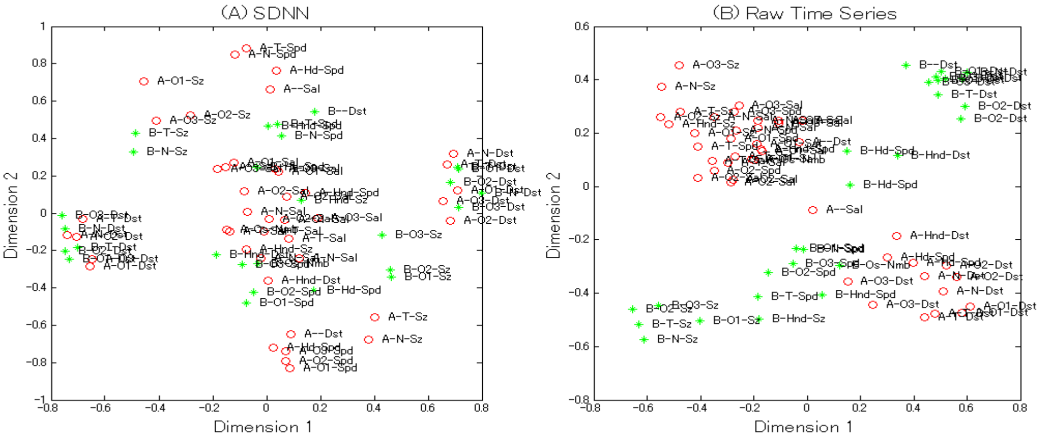

Figure 6: Multidimensional scaling visualization of (A) SDNN and (B) raw time series

to access symbol dynamics in a general form, and it opens secondary use of it. Second, supported by the probabilistic framework, unsupervised dimension selection works for discovery of dynamical property and reduction of noise factor. In particular, the second feature of SDNN, dimension selection, is useful for analyzing unknown time series including unknown dynamical property and noise factors. With two case studies in the present work, we suggest that SDNN based on generating partition has the potential to be used as a way to analyze multi-stream time series.

Large datasets of multimodal time series with high temporal (up to 10ms) and spatial (image pixel level) resolutions pose an unprecedented challenge in machine learning. Besides the pure amount of data, a particular demand is that very often such dataset creates an enormous search space for potentially interesting patterns. Therefore, a critical step is to start from scratch and analyze the overall temporal dynamics and structures from a set of temporal variables. By doing so, we can dramatically reduce the search space and bootstrap the whole pattern discovery process. Here we argue that SDNN introduced in this paper can serve this purpose by exploring statistical regularities in unsupervised mode without the need to add any prior knowledge/constraint. Just as a microscope allowing biologists to see objects too small for the naked eye, we suggest that data mining techniques combined with high-resolution multimodal data allow us to find novel micro-level behavioral patterns in multimodal interactions. Our present work shows the promise of this new data-driven venue.

7. ACKNOWLEDGEMENTS

We would like to thank Amanda Favata, Char Wozniak and Andrew Filipowicz for data collection. Thomas Smith is a great help in data pre-processing. This research is supported by NSF BCS 0924248, AFOSR FA9550-09-1-0665 and The Ogasawara Foundation for the Promotion of Science and Emgineering.

8. REFERENCES

[1] Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: Proceedings of the 1994 ACM SIGMOD international conference on Management of data. (1994) 419–429

[2] Lin, J., Keogh, E., Lonardi, S., Patel, P.: 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)

[3] Lee, W., Stolfo, S.: Data mining approaches for intrusion detection. In: Proceedings of the Seventh USENIX Security Symposium. (1998)

[4] Rabiner, L.R.: A tutorial on hidden markov models and selected applications inspeech recognition. In: Proceedings of the IEEE. (1989)

[5] Li1994: Markov random field models in computer vision. In: Computer Vision ? ECCV '94. Springer Berlin / Heidelberg (1994) 361–370

[6] Bengio, S., Bengio, Y.: An em algorithm for asynchronous input/output hidden markov models. In: International Conference On Neural Information. (1996)

[7] Bengio, S.: An asynchronous hidden markov model for audio-visual speech recognition. In: Advances in Neural Information Processing Systems. (2003)

[8] Schreiber, T.: Interdisciplinary application of nonlinear time series methods. Phys. Rep 308 (1998) 1–64

[9] Daw, C.S., A., F.C.E., Tracy, E.R.: A review of symbolic analysis of experimental data. Review of Scientific Instruments 74(2) (2003) 914–930

[10] Buhl, M., Kennel, M.B.: Statistically relaxing to generating partitions for observed time-series data. Physical Review E 71(4) (2005) 046213

[11] Kennel, M.B., Abarbanel, H.D.: False neighbors and false stands: A reliable minimum embedding dimension algorithm. Physical Review E 66(2) (2002) 026209

[12] Takens, F.: Detecting strange attractors in turbulence. In: Lecture Notes in Mathematics. Volume 898. Springer-Verlag (1981)

[13] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of Royal Statistical Society Series B 39 (1977) 1–38

[14] Yu, C., Smith, L., Shen, H., Pereira, A.F., Smith, T.: Active information selection: Visual attention through the hands. IEEE Transactions on Autonomous Mental Development 2 (2009) 141–151

[15] Yu, C., Scheutz, M., Schermerhorn, P.: Investigating multimodal real-time patterns of joint attention in an hri word learning task. In: Proceeding of 5th ACM/IEEE International Conference on Human-Robot Interaction. (2010)