Modeling Time Series and Sequences: Learning Representations and Making Predictions

by

Wenzhao Lian

Department of Electrical and Computer Engineering
Duke University

Date: __________________________
Approved:

__________________________
Lawrence Carin, Supervisor

__________________________
Robert Calderbank

__________________________
Jerry Reiter

__________________________
Galen Reeves

__________________________
Katherine Heller

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the Graduate School of Duke University
2015
Abstract

Modeling Time Series and Sequences: Learning Representations and Making Predictions

by

Wenzhao Lian

Department of Electrical and Computer Engineering
Duke University

Date: ______________________
Approved:

________________________
Lawrence Carin, Supervisor

________________________
Robert Calderbank

________________________
Jerry Reiter

________________________
Galen Reeves

________________________
Katherine Heller

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the Graduate School of Duke University 2015
Abstract

The analysis of time series and sequences has been challenging in both statistics and machine learning community, because of their properties including high dimensionality, pattern dynamics, and irregular observations. In this thesis, novel methods are proposed to handle the difficulties mentioned above, thus enabling representation learning (dimension reduction and pattern extraction), and prediction making (classification and forecasting). This thesis consists of three main parts.

The first part analyzes multivariate time series, which is often nonstationary due to high levels of ambient noise and various interferences. We propose a nonlinear dimensionality reduction framework using diffusion maps on a learned statistical manifold, which gives rise to the construction of a low-dimensional representation of the high-dimensional nonstationary time series. We show that diffusion maps, with affinity kernels based on the Kullback-Leibler divergence between the local statistics of samples, allow for efficient approximation of pairwise geodesic distances. To construct the statistical manifold, we estimate time-evolving parametric distributions by designing a family of Bayesian generative models. The proposed framework can be applied to problems in which the time-evolving distributions (of temporally localized data), rather than the samples themselves, are driven by a low-dimensional underlying process. We provide an efficient parameter estimation and dimensionality reduction methodology and apply it to two applications: music analysis and epileptic-seizure prediction.
The second part focuses on a time series classification task, where we want to leverage the temporal dynamic information in the classifier design. In many time series classification problems including fraud detection, a low false alarm rate is required; meanwhile, we enhance the positive detection rate. Therefore, we directly optimize the partial area under the curve (PAUC), which maximizes the accuracy in low false alarm rate regions. Latent variables are introduced to incorporate the temporal information, while maintaining a max-margin based method solvable. An optimization routine is proposed with its properties analyzed; the algorithm is designed as scalable to web-scale data. Simulation results demonstrate the effectiveness of optimizing the performance in the low false alarm rate regions.

The third part focuses on pattern extraction from correlated point process data, which consist of multiple correlated sequences observed at irregular times. The analysis of correlated point process data has wide applications, ranging from biomedical research to network analysis. We model such data as generated by a latent collection of continuous-time binary semi-Markov processes, corresponding to external events appearing and disappearing. A continuous-time modeling framework is more appropriate for multichannel point process data than a binning approach requiring time discretization, and we show connections between our model and recent ideas from the discrete-time literature. We describe an efficient MCMC algorithm for posterior inference, and apply our ideas to both synthetic data and a real-world biometrics application.
Dedicated to my parents.
## Contents

Abstract iv

List of Tables x

List of Figures xi

Acknowledgements xiii

1 Introduction 1

1.1 Background on time series and sequences ......................... 1

1.2 Representation learning for Time Series .......................... 2

1.3 Making predictions for Time Series .................................. 7

1.4 Representation learning for Sequences ............................. 10

1.5 Making Predictions for Sequences ................................. 13

1.6 Thesis Organization .................................................. 16

2 Multivariate Time-Series Analysis and Diffusion Maps 18

2.1 Introduction ............................................................ 18

2.2 Diffusion maps using Kullback-Leibler divergence .................. 23

  2.2.1 Underlying parametric model ................................. 23

  2.2.2 Local models and the Kullback-Leibler divergence .......... 23

  2.2.3 Laplace operator and diffusion maps .......................... 26

  2.2.4 Sequential implementation ..................................... 29

2.3 Modeling time evolving covariance matrices ........................ 31
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Detection rate and false alarm rate of the EEG samples from all the recordings</td>
<td>49</td>
</tr>
<tr>
<td>3.1</td>
<td>PAUC comparison in different scenarios, varying the false alarm region $[0, \beta]$ and the noise level $(\sigma/\mu)$</td>
<td>64</td>
</tr>
<tr>
<td>4.1</td>
<td>Effective sample size (ESS) and autocorrelation function (ACF) of the base rate $(\lambda_u)$, number of transitions of source $s_k(t)$ $(n_k)$, and the factor loadings $(w_{uk})$</td>
<td>77</td>
</tr>
</tbody>
</table>
List of Figures

2.1 (a) Normalized diffusion distance between time points. (b) Normalized approximated geodesic distance between time points. ................. 36

2.2 Analysis results of the song “A Day in the Life”. (a) Spectrogram of the song. (b)-(d) Comparison of the pairwise similarity matrices obtained by (b) diffusion maps using time evolving statistics, (c) diffusion maps using Euclidean distances between features, and (d) kernel beta process Ren et al. (2011). ......................... 40

2.3 Analysis results in the subinterval [65, 75] seconds of the song “A Day in the Life”. (a) Spectrogram of the song. (b)-(d) Comparison of the pairwise similarity matrices obtained by (b) diffusion maps using time evolving statistics, (c) diffusion maps using Euclidean distances between features, and (d) kernel beta process Ren et al. (2011). ......................... 41

2.4 The 2 principal eigenvectors as a function of time compared to human voice activity indicator .......................................................... 42

2.5 (a) A sample recording. The red line marks seizure’s onset, at approximate 56 minutes. (b) The STFT features of the recording. ................. 43

2.6 Scatter plots of the embedded samples of three 5-minute intervals of EEG recording 1. Each point is the diffusion map of the features of each time frame in the recording, setting $\ell = 3$. The colors indicate the different intervals. (a) The embedded samples under the MA modeling. (b) The embedded samples under the BAR modeling. ......................... 45

2.7 Scatter plots of the embedded samples of four 5-minute intervals of EEG recordings: two reference intervals (preseizure and interictal) from recording 1 and two new incoming intervals (preseizure and interictal) from recording 2. The representation is obtained using the reference samples and then extended to the new incoming samples according to Algorithm 2. The dimension of the embedding is set to $\ell = 3$ for visualization. (a) The obtained embedding under the MA modeling. (b) The obtained embedding under the BAR modeling. ......................... 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Left, graphical representation of HMMs; right, graphical representation of HCRFs.</td>
</tr>
<tr>
<td>3.2</td>
<td>Left (L) and right (R), coordinate 1 (L) and coordinate 2 (R) varying with time for a single instance.</td>
</tr>
<tr>
<td>3.3</td>
<td>Scatter plot of the two coordinates aggregated over time for all the instances; samples from the positive and negative classes cannot be separated (colored blue and red).</td>
</tr>
<tr>
<td>4.1</td>
<td>Normalized estimation error of instantaneous Poisson rate for proposed method and binning methods.</td>
</tr>
<tr>
<td>4.2</td>
<td>Estimation error of latent sources and factor loadings with different observation lengths and base rates.</td>
</tr>
<tr>
<td>4.3</td>
<td>Left: inferred sources using bsMJP and bMJP models. For each subplot, the first row shows the truth, the second the inferred source using bsMJP, and the third, that using bMJP. Right: posterior distribution of a shape parameter inferred using bsMJP.</td>
</tr>
<tr>
<td>4.4</td>
<td>Posterior distribution of number of latent sources.</td>
</tr>
<tr>
<td>4.5</td>
<td>Extracted arousal events from one user. Top: event arrivals; middle: jointly estimated rate. Bottom: rate inferred from single user’s trace.</td>
</tr>
<tr>
<td>4.6</td>
<td>ROC curves predicting (left) user arousal intensity, (right) user rating similarity from loading vectors $w_u$.</td>
</tr>
<tr>
<td>4.7</td>
<td>Posterior over number of dominant sources with a sample of sources (black represents 1 and red dash-dot lines delineate scenes the users were asked to recall).</td>
</tr>
<tr>
<td>4.8</td>
<td>The graphical representation of the proposed model.</td>
</tr>
</tbody>
</table>
Acknowledgements

First of all, I would like to thank my PhD advisor Lawrence Carin, who has provided me all kinds of support for years, including financial support and opportunities to collaborate with top researchers in other institutions. Larry inspired me on things in and outside research. During my years of research, his insightful ideas and ability to identify key problems have been really helpful. His way of thinking, diligent working, and philosophy of living are valuable to me throughout my whole life.

Then I would like to thank my thesis committee members, Jerry Reiter, Robert Calderbank, Galen Reeves, and Katherine Heller, for their comments on the dissertation and helpful discussions. Jerry is the first one who introduced me Bayesian statistics, his passion, knowledge, and the way of communication make me interested in this area and form a solid foundation for research. Robert and Galen raised some interesting points during the thesis writing, bringing quite some improvement. I have worked with Katherine and was impressed by her easygoingness and willingness to help. She also introduced me some related work on point processes, which was very helpful.

I would also like to thank my collaborators during my previous projects, Vinayak Rao at Purdue University and Piyush Rai at Indian Institute of Technology. Vinayak’s critical thinking, solid knowledge in statistics, and insightful ideas affect me a lot in research. He has great expertise in point processes and provides many key suggestions
during my exploration in this field. Piyush has a broad knowledge in machine learning and we had numerous discussions about ongoing and potential research topics, which was really helpful. I also want to thank Ronen Talmon at Technion University and Ronald Coifman at Yale University, who taught me a lot on ways of thinking from maths and engineering perspectives during my early projects. Since part of my research was conducted during summer internships, I want to thank my internship mentors, Brian Erickson in Technicolor Research Bay Area, Chris Meek and Asela Gunawardana in Microsoft Research Redmond, as well as Makoto Yamada, Jianhui Chen and Yi Chang in Yahoo Labs Sunnyvale. Their knowledge and different ways of thinking affect me greatly. I appreciate their help to make those summers fun and productive.

Lastly, I would like to thank my labmates at Duke, Xuejun Liao, Ricardo Henao, Bo Chen, Qisong Wu, Esther Salazar, Qi Mao, Yi Zhen, Joe Lucas, Jianbo Yang, Xin Yuan, Liming Wang, Minhua Chen, Eric Wang, Mingyuan Zhou, Zhengming Xing, Lingbo Li, Miao Liu, Xianxing Zhang, Yingjian Wang, David Carlson, Shanbo Han, Kyle Ulrich, and Changwei Hu. I had the opportunity to work closely with some of them in the past few years and those have been wonderful learning and researching experiences. Their help in and outside of research makes my years at Duke enjoyable and unforgettable.
1 Introduction

1.1 Background on time series and sequences

Time series and sequences are ubiquitous since information is usually collected over time, with applications including video processing, finance, biomedical problems, social networks, and E-commerce. Despite various formats of the temporal data, the tasks, in general, fall into two categories, learning representation and making predictions. For the representation learning category, taking video processing as an example, it might be needed to learn a low-dimensional representation for the frames in the video and measure the similarities between pair of frames. Similar applications can be also found in social networks, finance, and etc. For the prediction making category, using the electronic health records as an example, we have the hospital visit trajectories for a population of patients and it is useful to build a predictive model on when and for what reason the individual patients will visit the hospital in the future. These predictive models are also in demand in fields such as E-commerce and finance.

Before diving into the two categories of problems, we first provide a formal defi-
nition of time series and sequences discussed in the scope of this dissertation.

Time series observations usually involve a group of sequential samples \( \{ \{ x_{it} \}_{t=1}^{T_i} \}_{i=1}^{U} \), where \( U \) is the number of the collected series, \( T_i \) is the number of samples in each series. For simplicity, in this dissertation, we consider the number of samples \( T_i \) for each series \( i \) is the same, thus omitting the subscript and using \( T \) instead. \( x_{it} \in \mathbb{R}^P \) is a \( P \)-dimensional vector; in some scenarios, \( x_{it} \) can be restricted as an integer or a binary vector, for instance, the count series of types of events or the state of a subject (categorical variables can be transformed into a one-hot binary vector). For conventional time series, the sampling rate is uniform, i.e., the interval between any two adjacent samples \( x_{it} \) and \( x_{it+1} \) is the same across the whole series. In other words, no information lies in when the sample \( x_{it} \) is collected.

Sequences, or point processes, by contrast, are a collection of tuple sequences \( \{ \{ x_{ij}, y_{ij} \}_{j=1}^{N_i} \}_{i=1}^{U} \). \( y_{ij} \in \mathbb{R}^+ \) denotes the time stamp of \( j \)-th observation and \( x_{ij} \in \mathbb{R}^P \) is the observation at that time point. \( N_i \) is the number of observations of the \( i \)-th sequence, which is in general different among the sequence population. For example in electronic health records, \( y_{ij} \) denotes the time stamp of the \( j \)-th arrival of the \( i \)-th patient, and \( x_{ij} \) are the variables associated with that visit, e.g., reason/category of the visit and lab tests taken. Compared with the conventional time series observations, in sequences/point processes, the interval length between consecutive observations are informative; higher and lower frequency of observations can be useful features for both pattern discovery and prediction tasks.

In the following four sections, we describe the problem formulation in more details and examine the related literatures.

1.2 Representation learning for Time Series

Modeling time series has been studied for decades in the community of both machine learning and statistics. Popular approaches include autoregressive models, hidden
Markov models, Gaussian process based models, and neural network based models; for more details, please see West (1996); Roberts et al. (2013); Martens and Sutskever (2011). In this thesis, we focus on learning representations, in particular, dimension reduction for high-dimensional time series. Two general types of efforts have been taken to tackle this problem, latent variable model approaches (also known as the mapping methods) and data-driven approaches (known as the embedding methods). The former methods make certain assumptions about the series, introduce latent variables to model the hidden dynamics in the low-dimensional space, and at the same time, learn the mapping (Doretto et al., 2003; Damianou et al., 2011; Wang et al., 2005; Pavlović et al., 1999; Pavlovic et al., 2000; Li et al., 2007); meanwhile, the latter methods make weaker assumptions and embed the original series onto a low-dimensional space, based on the some distance measures between segments/points in the series (Jenkins and Matarić, 2004a; Talmon and Coifman, 2013). We first discuss a few mapping methods proposed in recent years and then investigate a couple of embedding approaches.

Linear dynamic systems (LDS) are the simplest models which assume an autoregressive process in the hidden space (Doretto et al., 2003), typically represented as,

\begin{align}
\mathbf{x}_t &= \mathbf{A}\mathbf{z}_t + \mathbf{e}_t, \\
\mathbf{z}_t &= \mathbf{B}\mathbf{z}_{t-1} + \mathbf{r}_t,
\end{align}

where \(\mathbf{z}_t\) is the hidden representation, \(\mathbf{A}\) is the observation matrix, \(\mathbf{B}\) is the transition matrix, and \(\{\mathbf{e}_t, \mathbf{r}_t\}\) are often assumed to be Gaussian distributed. Model parameter learning (e.g., \(\mathbf{A}, \mathbf{B}\)) and hidden variable inference (\(\mathbf{z}\)) can be done alternately, as typically done in expectation-maximization (EM) algorithms. \(\mathbf{z}\) has a much lower dimensionality than \(\mathbf{x}\) and thus, a low-dimensional representation can be learned after the inference. Because the linear constraints on the dynamics limit the expres-
sive power of LDS, switching linear dynamic systems (SLDS) are proposed (Pavlović et al., 1999; Pavlovic et al., 2000), where the transition matrix $B$ is assumed to be state-dependent,

$$z_t = B(s_t)z_{t-1} + r_t, \tag{1.3}$$

$$Pr(s_t = j | s_{t-1} = k) = \pi_{kj}. \tag{1.4}$$

In above, hidden states $\{s_t\}$ are introduced to enrich the dynamics of $z$, i.e., different transition happens in different states $s_t$. To further increase the expressive power, the mapping from $z_t$ to observation $x_t$ is also assumed to be nonlinear and approximated by a piecewise linear function (Li et al., 2007), i.e.,

$$x_t = A(s_t)z_t + e_t(s_t), \tag{1.5}$$

$$z_t = B(s_t)z_{t-1} + r_t(s_t). \tag{1.6}$$

EM-type learning algorithms with variational inference can be applied to such models, where the hidden variables $z_t$ and $s_t$ are inferred, and the transition, observation, and noise distribution parameters are learned. Thus, the nonlinear manifold and nonlinear dynamics can be learned simultaneously.

In the above mentioned models, the nonlinearity is constructed as piecewise linear approximations, where the number of pieces/hidden states is sometimes hard to set or learn. Therefore, another type of models, Gaussian process latent variable models (GPLVMs), are quite popular in the machine learning community (Damianou et al., 2011; Wang et al., 2005). GPLVMs are nonparametric and allow flexible forms of transition dynamics and observation mappings,

$$x_{td} = f_d(z_t) + e_{td}, \quad e_{td} \sim \mathcal{N}(0, \beta^{-1}), \tag{1.7}$$

$$z_q(t) \sim \mathcal{GP}(0, k_z(t_i, t_j)), q = 1, \ldots, Q, \tag{1.8}$$

$$f_d(x) \sim \mathcal{GP}(0, k_f(x_i, x_j)), d = 1, \ldots, D. \tag{1.9}$$
In above, \( k_f(\mathbf{x}_i, \mathbf{x}_j) \) and \( k_z(t_i, t_j) \) are kernel functions (e.g., the radial basis function) defined on the latent space and time domain, respectively. The assumption is that closeness in time suggests similar values \( z \) in the latent space, and thus similar observations \( x \). Because of this weak smoothness assumption, GPLVMs can model complex functions with sufficient training data. Maximum a posteriori (MAP) solutions are usually pursued in GPLVMs via variational inference. The objective is highly non-convex and the computational cost is high; how to solve the inference tasks remain an open problem (Damianou et al., 2011; Yao et al., 2011).

Compared with mapping methods, the embedding approaches do not explicitly model how the dynamics evolve. Instead, they first compute distances between time points and use the pairwise distance matrix to find the low-dimensional embedding. For example in Jenkins and Matarić (2004a), overlapping or non-overlapping segments are first extracted and the Euclidean distance between segments is computed. Second, leveraging temporal information, temporal neighbours are identified using some pre-defined criteria; then, the distances are reduced for the temporal neighbours identified to encourage smoothness. Geodesic distances between segments are then computed by finding the shortest paths between segment pairs, following which multiscale dimension scaling methods (e.g., Laplacian eigenmaps) are applied to learn the low-dimensional embeddings. In some cases it is difficult to define the temporal neighbourhood as in Jenkins and Matarić (2004a). Thus, in Talmon and Coifman (2013), an intrinsic low-dimensional diffusion process is assumed to drive the dynamics of the observed high-dimensional series, as given by the Langevin equation,

\[
    d\mathbf{z}_t^d = a_d(z_t^d)dt + dw_t^d
\]

where \( z_t^d \) is dimension \( d \) of \( \mathbf{z}_t \), \( \{a_d\} \) are (possibly nonlinear) drift functions and \( \{w_t^d\} \) are Brownian motions. Mahalanobis distances between empirical distribution estimators (e.g., histogram vectors) are used to construct the affinity measure between
segments in the series. Then anisotropic kernels are construed and diffusion maps are applied to obtain a low-dimensional embedding, which uncovers the intrinsic representation. In a non-rigorous sense, the process can be justified as follows. The probability density function (PDF) of the observed sample \( x_t \) is a linear transformation of the PDF of a clean signal component (without the additive noise), and the empirical distribution (histogram) is again a linear transformation of the PDF of the clean signal component. Because the Mahalanobis distance is invariant under linear transformations and because of the geodesic distance preserving property of diffusion maps, the Euclidean distance between low-dimensional representations obtained approximates the geodesic distance between intrinsic embeddings. The idea of reducing noise of the observed series in a probabilistic manner in Talmon and Coifman (2013) is an effective attempt, but in high-dimensional scenarios, the histogram method is not feasible due to the curse of dimensionality.

To overcome the too-strong model assumptions on dynamics of the mapping methods and the difficulty of handling ambient noise of the embedding methods, in this thesis, we propose a framework to combine the merits of both methods as briefly overviewed here. First, we introduce modest model assumptions on the series and infer time-evolving parametric distributions, where the distribution parameters serve as an intermediate representation of the original series (with smaller noise). Then, we define a distance between parametric distributions on this statistical manifold and apply diffusion maps to learn the low-dimensional embeddings. The learned low-dimensional representation preserves the geodesic distances between samples on the statistical manifold, and uncovers the underlying process that drives the original series. The detailed framework and analysis will be described in Chapter 2. This work has also been published in Lian et al. (2015b).

In recent years, the deep neural networks become popular in many areas, and also gain lots of attention in time series modeling. Because of the universal power
of approximating functions, they can learn flexible forms of transition patterns and observation mappings. Such models include the conditional Restricted Boltzmann Machine (CRBM) (Taylor et al., 2006), the recurrent neural network (RNN) (Fragkiadaki et al., 2015), and etc. The discussion of this line of work is not in the scope of this thesis.

1.3 Making predictions for Time Series

Two types of prediction problems exist for time series, forecasting (one or multistep ahead) and series classification. The former has been a popular research area for decades, including autoregressive models (West, 1996), Gaussian processes (Girard et al., 2003), hidden Markov models (Clements et al., 2004), and recurrent neural networks (Giles et al., 2001). For a detailed understanding, the readers are refereed to a recent review in Montgomery et al. (2015). However, though briefly surveyed by the data mining community in Xing et al. (2010), the latter problem on series classification has been less explored or systematically studied in the community of machine learning and statistics, which is the focus in the scope of this thesis.

The series classification approaches in general fall into two categories: feature based, and model based approaches. These can be considered as different ways to define similarity between series, i.e., similarity in the feature space or measured in terms of data likelihood given a model.

The feature based methods usually have two steps, where in the first stage features are constructed based on domain knowledge. The most straightforward feature design is simply treating the series as feature vectors, using which classifiers can be learned. But because of non-alignment and scaling issues, dynamic time warping which warps the original series by aligning and padding, in some cases, improves the robustness of the classifier (Keogh and Pazzani, 2000). Dynamic programming is used to find the optimal warping, which has a quadratic complexity, makes this
approach unrealistic in large datasets. Instead of using the whole series, computing statistics of the series and extracting a low-dimensional representation are more common in practice. For example, statistics of the series including the mean, variance, curvature, spikiness, linearity, and autocorrelation can be computed (Hyndman et al., 2015). Other heuristics like k-grams and string kernels can be used in some discrete series scenarios (Xing et al., 2010). After feature vectors are constructed, standard classifiers such as nearest neighbours and support vector machines can be directly applied. Feature based methods are flexible in the sense that domain knowledge can be incorporated in feature design, which is also a drawback in that no guarantees are provided especially if clear understanding of the data is unavailable before the classifier design. To tackle this issue, shapelet was proposed to automate the feature design process and became popular in the data mining community (Ye and Keogh, 2009). Shapelet approach basically constructs a pool of short window segments of the series and select those with discriminative power. It works well in some cases where certain crucial segments (shapelets) are strong indicators of the class label, but it loses long term structure of the series. Another reason prohibiting its usage is the computation complexity; it cannot scale as the number of series and series lengths increase to hundreds of thousands or more.

The model based methods have been popular in recent years because of their ability to constrain the model space in a principled manner (Jaakkola et al., 1999; Kim and Pavlovic, 2011; Pei et al., 2015). HMM-based classification is a straightforward yet powerful method, where it is assumed that in different classes, different parameters modeling transition and emission patterns in HMMs exist (Kim and Pavlovic, 2011). During training, the model parameters are learned for each class via the Baum-Welch algorithm, which is an EM-type algorithm, with a guarantee to converge towards a local optima. In testing, we can evaluate the likelihood of the testing series using parameters from each class and choose the one with maxi-
mum likelihood. Because models trained for maximizing the likelihood of generating observations might not be tuned for classification, Jaakkola et al. (1999) proposes using the Fisher kernel to do discriminative training under the same model assumptions in HMMs. They derive the Fisher kernel following the probability distribution parametrized by HMMs and use the Fisher kernel to compute the distance between pairs of series. By optimizing a score function (e.g., the likelihood ratio of the true class vs. the wrong one), the model parameters in the probabilistic model are learned.

Because in many scenarios, HMMs have a too constrained assumption, HCRFs are applied to time series classification tasks while more flexibly leveraging the temporal information (Quattoni et al., 2007). In HCRFs, an undirected graphical model is defined and the edges correspond to potential constraints measuring the compatibility between node values. Therefore, HCRFs are flexible where we can add or remove the edges between nodes to add or remove a constraint. The potential function can be transformed to the conditional probability of label given the observed sequence, for example, through the soft max function. By optimizing the conditional probability, we can learn the model parameters controlling the constraints. Even though the objective function in HCRFs is not convex, the local optimal found usually achieves state-of-the-art results (Pei et al., 2015). To tackle the computation complexity as the number of states increases, Pei et al. (2015) recently proposed a hidden-unit logistic model (HULM). HULM uses a multi-dimensional binary vector to represent the hidden states, where the number of states exponentially increases as the number of hidden binary nodes; HULM further improves the representation power and classification performance with the same number of model parameters.

The previously introduced HCRFs and HULMs are trained via maximizing the conditional probability of the label given time series, which can be considered as using the negative log-likelihood as the loss function. Another choice of loss is the hinge loss, corresponding to the max-margin training, which often achieves better perfor-
mance in classification tasks (Wang and Mori, 2011). In this thesis, we proposed a model-based approach with the max-margin training but focus on the classification performance in the low false-alarm regions, measured by partial area under the curve (PAUC). By directly optimizing this objective, we obtain a better performance in some cost-sensitive classification tasks, as detailedly described in Chapter 3.

1.4 Representation learning for Sequences

There are two aspects of sequence representation: first, learning the intensity underlying the sequence, i.e., modeling the event arrival rate; second, extracting the relationship between multiple sequences when observations consist of more than one of them.

In the individual level, a sequence be naturally modeled by point processes where the intensity function governs the frequency of events’ arrival, as shown in (1.11),

\[
p(Y^i) = \prod_{n=1}^{N_i} \lambda^i(y^i_n) \exp\left(- \int_{y^i_n}^{y^i_{n+1}} \lambda^i(y^i_n)d\tau \right),
\]

where \(\lambda^i(t)\) denotes the intensity function for \(i\)-th sequence. The key problem is how to model the intensity functions.

GP-modulated point processes are a popular approach for modeling event streams (Adams et al., 2009; Rao and Teh, 2011; Lloyd et al., 2014; Lasko, 2014). Assuming a smoothly varying intensity function, the intensity rate, as well as its uncertainty, can be estimated from observed streams. Specifically, we put a Gaussian process prior on the intensity function. To impose the positiveness of \(\lambda^i(t)\), we define \(\lambda^i(t) = g(f^i(t))\), where \(f^i(\cdot) : \mathcal{T} \rightarrow \mathbb{R}\), and \(g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^+\), e.g., square, exponential, and etc. Then we impose a GP prior on \(f^i\), i.e., \(f^i(\cdot) \sim GP(m(\cdot), C(\cdot, \cdot))\). By maximizing the likelihood of the observed sequences, we can learn model parameters and the posterior distribution of the intensity function, which represents the sequences.
When multiple synchronized sequences are available, the GP modulated point processes can be naturally extended, e.g., the Gaussian process factor analysis (GPFA) approach in Yu et al. (2009). One difference is that in Yu et al. (2009), they did not directly model the event arrivals, instead, they aggregated the arrivals into counts in consecutive bins and the problem degraded to traditional time series discussed in Section 1.2. They model the aggregated counts (more accurately, the square-roots of the counts) as a series of correlated neural spike trains by linearly transforming a small set of latent functions drawn from Gaussian processes. Therefore, this model reduces the sequence dimension and uncover smooth functions evolving over time. The latent function usage of the sequences provides the relationship between them, e.g., two sequences having the same increasing/decreasing trend in event counts will share a similar usage of latent functions.

One key problem with these GP modulated point process models is the smoothness assumption does not hold in many scenarios. Sudden rate changes often happen upon event arrivals, e.g., in the patients’ visiting hospital scenario, the risk of a patient’s hospital visit might change significantly after a single visit. Three types of approaches can tackle this issue, non-stationary GPs (Lasko, 2015), conditional intensity models (Parikh et al., 2012), and hidden state models (Stimberg et al., 2014; Escola et al., 2011).

In Lasko (2015), the warping idea is used to first learn a warping function \( v(t) = \frac{1}{\sigma(t)} \), where \( \sigma(t) \) denotes the time-varying correlation length in the GP kernel (which is often assumed as a constant). A GP prior is imposed on \( v(t) \) with another set of GP hyper-parameters. Using \( \{w_i\} \) to denote the warped time points \( \{t_i\} \), the arrival sequence in the warped space can be modeled by a stationary GP-modulated point process, where the correlation length of the GP in the warped space \( \sigma_w \) can be considered as a single constant throughout the sequence. By using this two layer
GP construction, this model with warped time space can deal with event sequences with non-stationary intensity functions. For example, the intensity function changes rapidly in some intervals while in some other intervals the intensity is approximately a constant.

Instead of using GPs to model the intensity function, another line of work builds models assuming the intensity rate is a function of history events (Parikh et al., 2012; Rajaram et al., 2005). Specifically in Rajaram et al. (2005), the intensity rate $\lambda(t)$ is parametrized by a generalized linear model where the regressors are constructed from the event history, e.g., the event counts for all sequences in $[t - L, t]$ where $L$ is a predefined interval length. By constructing such features and learning model parameters, we can learn a dependency structure among the sequences and represent the sequences with the learned intensity functions.

In the line of research of hidden state modeling for sequences, early work did not exploit correlation across multiple arrival streams (Daley and Vere-Jones, 1998; Kass and Ventura, 2001; Riehle et al., 1997). Hidden state variables are commonly introduced with Markov assumptions or autoregressive structures; the hidden Markov process then modulates the arrival sequences through an inhomogeneous Poisson process (Daley and Vere-Jones, 1998). The work in Smith and Brown (2003) introduces a single first-order autoregressive latent state model with a proposed EM-algorithm to estimate parameter values from the arrival data. A more complex HMM approach using multiple latent states is presented in Escola et al. (2011). They define the HMM parameters including the transition and emission probabilities as parametric functions of event histories; EM-algorithm is applied to learn the function parameters and hidden variables. More recently, to adaptively represent the intensity function using HMMs, Stimberg et al. (2014) applied the Chinese restaurant process to infer the hidden states in a non-parametric fashion. A sophisticated Markov chain Monte Carlo (MCMC) sampling scheme was proposed to infer the hidden states and the
intensity function.

In this thesis, we tackle the problem of multiple sequence representation, learning both the intensity rate functions and the sequence relationships. We take the hidden state model approach and propose a binary semi-Markov jump process to model a small set of latent sources, where each source can have two states, i.e., on or off. We further model each intensity function as a weighted combination of the latent sources. A few nice properties of the model are presented: first, the semi-Markovian nature allows the flexibility to characterize different transition patterns, such as some states favor stickiness while some states favor transition instantly; second, the number of latent sources can be inferred in a non-parametric manner via efficient Gibbs sampling; third, the relationship among the sequences can be captured by the sequence-source usage matrix. Details will be introduced in Chapter 4. This work has also been published in Lian et al. (2014).

1.5 Making Predictions for Sequences

In most applications involving point process observations, people are interested in making predictions for future events’ pattern. For example, given a patient’s hospital visiting sequences, answering the questions like when he/she will visit the hospital again or how many times he/she will visit the hospital in the next month. Similar tasks can be found in predicting device failure in a computer cluster or modeling purchasing behavior in an online business.

The GP modulated point processes introduced in Section 1.4 can be extended for prediction problems by using extrapolation (Adams et al., 2009). Specifically, after learning the posterior distribution of the intensity function $f^i$ using the observations, we can extrapolate samples to future time points. The uncertainty on the rate increases as the length of extrapolation increases. One problem with such prediction methods is because the input variable is time, they cannot capture causal
relationships. For instance, such methods can capture patterns like seasonality and linearity, but cannot extract patterns such as type A events’ happening leads to an increasing rate of type B events.

Therefore, a few models are directly designed for predictive tasks recently, some of which come from the neural decoding literature (Kulkarni and Paninski, 2007; Rad and Paninski, 2011; Pillow et al., 2008), where multiple subjects/neurons are affected by common stimuli (features), generating event streams. A generalized linear model can be trained on the stimuli and spiking history to learn the rate function, and further predict future spiking events. Meanwhile, the network structure across neurons can be inferred, which in turn, helps with prediction. Specifically,

$$\lambda^i(t) = \exp(w^T e + v^T h^i(t) + \sum_{u \neq i} \beta_{ui}^T h^u(t) + \mu^i),$$

where $e$ is the external stimulus, $h^i(t)$ denotes the spiking history of $i$-th neuron up to time $t$, and $\mu^i$ is the baseline spiking rate for the $i$-th neuron. This model can be readily used for forward prediction after learning the model parameters from the observed sequences. However, temporal alignment is also assumed in these models, which is valid in the setting of neural decoding, but not in general cases.

Another line of research is Rajaram et al. (2005); Gunawardana et al. (2011), where the intensity function is constrained to be piecewise-constant, learned using decision trees and used for prediction. Specifically, it constructs features from the past event traces. For example, the feature vector containing counts of each event type in a given time window. The function mapping from history to the future intensity rate is approximated by a decision tree which allows a non-linear mapping. The model is applied to the log data of a computing cluster, consisting the time stamps of device failures like CPUs, memory, and hard drives. It can predict the future device failure time and type. One caveat is that this model is designed for one trace (or multiple synchronized traces), where it can use a chunk of trace for training.
and perform forward prediction. If the data available for training is insufficient, which is often true, the decision tree cannot be accurately learned, thus leading to decreased performance.

Also related is the work of Weiss and Page (2013), which integrates the multiplicative forest Continuous Time Bayesian Network (mfCTBN) with the piecewise-constant conditional intensity model of Gunawardana et al. (2011). More precisely, they learn random forests (instead of trees) mapping from demographic and event history features to intensity rates. This model deals with multiple (asynchronized) traces but under the assumption that all traces have the same function mapping from history to the future rate. In other words, each event trace is a realization trajectory of some underlying model. The data from all the event traces are used to learn a single function approximator (the multiplicative forest) for the intensity rate. But in more general cases where different traces have different history-to-intensity mapping functions, this model cannot be applied.

A more recent work by the author (Lian et al., 2015a), multiple asynchronized sequences are jointly analyzed. In this work, a different setting is considered compared with Weiss and Page (2013), where the variability across subjects cannot be ignored. This difference is crucial in scenarios such as healthcare analysis and electronic commerce, where variability among members of a population affects arrival pattern discovery and predictive performance. Because the available training data might be insufficient to learn a different predictive model for each sequence, sharing across the sequences is necessary. In Lian et al. (2015a), history features are first extracted from different sequences and mapped to a common feature space. Then, on the shared feature space intensity rate functions are learned for each individual sequence, but data are shared via a multitask Gaussian process. For sequences with sufficient training data, their intensity functions are determined mainly by their own data; meanwhile, for sequences with few training data, their intensity functions are
mainly affected by the global mean intensity function determined by data from all sequences. The model is applied to electronic health records demonstrating its power to predict patients’ future hospital visit patterns, such as when the patient will visit again and how many times the patient will visit in the next month. The details of this work are presented in Lian (2015); it is summarized here for completeness of this thesis.

1.6 Thesis Organization

The main part of the thesis is organized following the same order as described previously, i.e., from representation learning and making predictions for time series to representation learning for point processes. Making predictions for point processes is thoroughly discussed in the author’s master thesis and omitted here (Lian, 2015).

In Chapter 2, multivariate time series is studied, where we learn the similarity between segments of the series and obtain a low-dimensional representation via dimension reduction. Because time series are often nonstationary due to high levels of ambient noise and various interferences. We propose a nonlinear dimensionality reduction framework using diffusion maps on a learned statistical manifold, which gives rise to the construction of a low-dimensional representation of the high-dimensional nonstationary time series. We show that diffusion maps, with affinity kernels based on the Kullback-Leibler divergence between the local statistics of samples, allow for efficient approximation of pairwise geodesic distances. To construct the statistical manifold, we estimate time-evolving parametric distributions by designing a family of Bayesian generative models. The proposed framework can be applied to problems in which the time-evolving distributions (of temporally localized data), rather than the samples themselves, are driven by a low-dimensional underlying process. We provide efficient parameter estimation and dimensionality reduction methodology and apply it to two applications: music analysis and epileptic-seizure prediction.
In Chapter 3, we focus on a time series classification task, where we want to leverage the information of temporal dynamics in the classifier design. In many time series classification problems including fraud detection, a low false alarm rate is required; meanwhile, we enhance the positive detection rate. Therefore, we directly optimize the PAUC, which maximizes the accuracy in low false alarm rate regions. Latent variables are introduced to incorporate the temporal information, while maintaining a max-margin based method solvable. An optimization routine is proposed with its properties analyzed. Simulation results are provided to demonstrate the performance of the proposed method compared with conventional approaches.

In Chapter 4, we focus on pattern extraction from correlated point process data, which consist of multiple correlated sequences observed at irregular times. The analysis of correlated point process data has wide applications, ranging from biomedical research to network analysis. We model such data as generated by a latent collection of continuous-time binary semi-Markov processes, corresponding to external events appearing and disappearing. A continuous-time modeling framework is more appropriate for multichannel point process data than a binning approach requiring time discretization, and we show connections between our model and recent ideas from the discrete-time literature. We describe an efficient MCMC algorithm for posterior inference, and apply our ideas to both synthetic data and a real-world biometrics application.

In Chapter 5, we conclude the thesis and point out a few interesting directions to explore.
2

Multivariate Time-Series Analysis and Diffusion Maps

2.1 Introduction

In the study of high-dimensional data, it is often of interest to embed the high-dimensional observations in a low-dimensional space, where hidden parameters may be discovered, noise suppressed, and interesting and significant structure revealed. Due to high dimensionality and nonlinearity in many real-world applications, nonlinear dimensionality reduction techniques have been increasingly popular (Tenenbaum et al., 2000; Belkin and Niyogi, 2003; Coifman and Lafon, 2006). These manifold-learning algorithms build data-driven models, organizing data samples according to local affinities on a low-dimensional manifold. Such methods have broad applications to, for example, analysis of financial data, computer vision, hyperspectral imaging, and biomedical engineering (Durante et al., 2012; Fox and West, 2011; Coifman and Hirn, 2012).

The notion of dimensionality reduction is useful in multivariate time series analysis. In the corresponding low-dimensional space, hidden states may be revealed,
change points detected, and temporal trajectories visualized (Zuur et al., 2003; Talmon and Coifman, 2013, 2012). Recently, various nonlinear dimensionality reduction techniques have been extended to time series, including spatio-temporal Isomap (Jenkins and Matarić, 2004b) and temporal Laplacian eigenmap (Lewandowski et al., 2010). In these methods, besides local affinities in the space of the data, available temporal covariate information is incorporated, leading to significant improvements in discovering the latent states of the series.

The basic assumption in dimensionality reduction is that the observed data samples do not fill the ambient space uniformly, but rather lie on a low-dimensional manifold. Such an assumption does not hold for many types of signals, for example, data with high levels of noise (Durante et al., 2012; Fox and Dunson, 2011; Carter et al., 2011, 2009). In Carter et al. (2011) and Carter et al. (2009), the authors consider a different, relaxed dimensionality reduction problem on the domain of the underlying probability distributions. The main idea is that the varying distributions, rather than the samples themselves, are driven by few underlying controlling processes, yielding a low-dimensional smooth manifold in the domain of the distribution parameters. An information-geometric dimensionality reduction (IGDR) approach is then applied to obtain an embedding of high-dimensional data using Isomap (Tenenbaum et al., 2000), thereby preserving the geodesic distances on the manifold of distributions.

Two practical problems arise in these methods, limiting their application. First, in Carter et al. (2011, 2009) multiple data sets were assumed, with the data in each set drawn from the same distributional form, with set-dependent distribution parameters. The embedding was inferred in the space of the distribution parameters. In this setting a large number of data sets are required, and time dependence in the evolution of the distribution parameters is not considered. By considering time evolution of the distribution from a single time-evolving dataset, we here substantially
reduce the amount of needed data, and we extend the method to analysis of time series. A second limitation of previous work concerns how geodesic distances were computed. In Carter et al. (2011, 2009) the approximation of the geodesic distance between all pairs of samples was computed using a step-by-step walk on the manifold, requiring $O(N^3)$ operations, which may be intractable for large-$N$ problems.

In this chapter, we present a dimensionality-reduction approach using diffusion maps for nonstationary high-dimensional time series, which addresses the above shortcomings. Diffusion maps constitute an effective data-driven method to uncover the low-dimensional manifold, and provide a parametrization of the underlying process (Singer and Coifman, 2008). The main idea in diffusion maps resides in aggregating local connections between samples into a global parameterization, via a kernel. Many kernels implicitly induce a mixture of local statistical models in the domain of the measurements. In particular, it is shown that using distributional information outperforms using sample information when the distributions are available (Carter et al., 2011). We exploit this assumption and articulate that the observed multivariate time series $X_t \in \mathbb{R}^N, t = 1, \ldots, T$, is generated from a smoothly varying parametric distribution $p(X_t | \beta_t)$, where $\beta_t$ is a local parameterization of the time evolving distribution. We propose to construct a Bayesian generative model with constraints on $\beta_t$, and use Markov Chain Monte Carlo (MCMC) to estimate $\beta_t$. Diffusion maps are then applied to reveal the statistical manifold (of the estimated distributions), using a kernel with Kullback-Leibler (KL) divergence as the distance metric. Noting that the parametric form of distributions significantly affects the structure of the mapped data, the Bayesian generative model should avoid using a strong informative prior without substantial evidence.

Diffusion maps rely on constructing a Laplace operator, whose eigenvectors approximate the eigenfunctions of the backward Fokker-Planck operator. These eigenfunctions describe the dynamics of the system (Nadler et al., 2006). Hence, the
trajectories embedded in the coordinate system formulated by the principal eigenvectors can be regarded as a representation of the controlling underlying process $\theta_t$ of the time series $X_t$.

One of the main benefits of embedding the time series samples into a low-dimensional domain is the ability to define meaningful distances. In particular, diffusion-map embedding embodies the property that the Euclidean distance between the samples in the embedding domain corresponds to a diffusion distance in the distribution domain. Diffusion distance measures the similarity between two samples according to their connectivity on the low-dimensional manifold (Coifman and Lafon, 2006), and has a close connection to the geodesic distance. Thus, diffusion maps circumvent the step-by-step walk on the manifold (Carter et al., 2011), computing an approximation to the geodesic distance in a single low-cost operation.

Another practical advantage of the proposed method is that we may first reveal the low-dimensional coordinate system based on reference data, and then in an online fashion extend the model for newly acquired data with low computational cost. This is demonstrated further when considering applications in Section 2.4.

The proposed framework is applied to two applications in which the data are best characterized by temporally evolving local statistics, rather than based on measures directly applied to the data itself: music analysis and epileptic seizure prediction based on electroencephalography (EEG) recordings. In the first application, we show that using the proposed approach, we can uncover the key underlying processes: human voice and instrumental sounds. In particular, we exploit the efficient computation of diffusion distances to obtain intra-piece similarity measures, applied to well-known music, which are compared to state-of-the-art techniques.

In EEG, one goal is to map the recordings to the unknown underlying “brain activity states”. This is especially crucial in epileptic seizure prediction, where pre-seizure (dangerous) states can be distinguished from interictal (safe) states, so that
patients can be warned prior to seizures (Frei et al., 2010). In this application, the observed time series is the EEG recordings and the underlying process is the brain state: preseizure or interictal. EEG recordings tend to be noisy, and hence, the mapping between the state of the patient’s brain and the available measurements is not deterministic, and the measurements do not lie on a smooth manifold. Thus, the intermediate step of mapping the observations to a time-evolving parametric family of distributions is essential to overcome this challenge. We use the proposed approach to infer a parameterization of the signal, viewed as a model summarizing the signal’s distributional information. Based on the inferred parameterization, we show that preseizure state intervals can be distinguished from interictal state intervals. In particular, we show the possibility of predicting seizures by visualization and simple detection algorithms, tested on an anonymous patient.

This chapter makes three principal contributions. We first present an efficient diffusion map approach based on distributional information of time-series data, which preserves the geodesic distances between samples on a statistical manifold, and uncovers an underlying process that consists of the controlling factors. Second, we propose a class of Bayesian models with various prior specifications, to learn the time-evolving statistics. We finally apply the proposed framework to two applications: music analysis and analysis of EEG recordings.

The remainder of the chapter is organized as follows. In Section 2.2 we review the diffusion-maps technique, propose an extended construction and examine its theoretical and practical properties. We propose in Section 2.3 multiple approaches to model multivariate time series with time-evolving distributions. In Section 2.4, results on two real-world applications are discussed. Conclusions and future work are outlined in Section 2.5.
2.2 Diffusion maps using Kullback-Leibler divergence

2.2.1 Underlying parametric model

Let $X_t \in \mathbb{R}^N$ be the raw data or extracted features at time $t$. The key concept is that the high-dimensional representation of $X_t$ exhibits a characteristic geometric structure. This structure is assumed to be governed by an underlying process on a low-dimensional manifold, denoted by $\theta_t$, that propagates over time as a diffusion process according to the following stochastic differential equation (SDE)

$$d\theta^i_t = a_i(\theta^i_t)dt + dw^i_t$$  \hspace{1cm} (2.1)

where $\theta^i_t$ is component $i$ of $\theta_t$, $a_i$ are (possibly nonlinear) drift functions and $w_t$ is a Brownian motion. In particular, we assume that the underlying process induces a parametric family of probability distributions in the measurable domain, i.e., $p_X(\cdot|\theta_t)$. In other words, the underlying process controls the statistics of the measured signals.

Note that $\theta_t$ controls the time-evolution of the underlying distribution of the data, rather than directly the data itself. We do not assume a priori knowledge of the form of the distribution $p(\cdot|\theta_t)$.

2.2.2 Local models and the Kullback-Leibler divergence

We use empirical densities to represent the local statistics of the signal. In particular, as an intermediate step, we assume a parametric family of local distributions. Let $p(X_t|\beta_t)$ denote the local density of $X_t$. We emphasize that the assumed parameterization of the local distribution $\beta$ is considerably different than $\theta$: $\theta$ is the fundamental parametrization of the low-dimensional manifold, that represents the intrinsic state governing the signal; $\beta$ is merely used within a chosen local distribution, employed as an intermediate step, with the goal of inferring $\theta$.

\footnote{In this chapter, superscripts represent access to elements in vectors, i.e., $x^i$ is the $i$-th element of the vector $x$.}
The key thing to note is that because the data are assumed noisy, we do not assume the data itself lives on a low-dimensional manifold. Rather, we assume that there is an underlying and unknown distribution \( p(X_t|\theta_t) \), that evolves with time, and that is responsible for the data. To uncover the time evolution of \( \theta \) (and to infer the dimension of the parameter vector \( \theta \)), we assume a form of a generally different distribution \( p(X_t|\beta_t) \), typically selected to balance accuracy with computational simplicity. We then compute distances between data at time \( t \) and \( t' \), based on \( p(X_t|\beta_t) \) and \( p(X_{t'}|\beta_{t'}) \), using an appropriate kernel. From the resulting distance matrix we seek to uncover \( \theta_t \) for all \( t \). In the end we still do not know the responsible distribution \( p(X_t|\theta_t) \), but we uncover how the (typically low-dimensional) parameters \( \theta_t \) evolve with time, manifesting a useful embedding. In Carter et al. (2011, 2009) the authors also estimated distributions \( p(X_n|\beta_n) \) for multiple data sets, here with \( X_n \) representing all the data in dataset \( n \in \{1, \ldots, N\} \); they estimate the associated \( \{\theta_n\} \) via a similar embedding procedure. By leveraging time, we effectively infer \( N \) local datasets, characterized by time-evolving distributions.

We propose to use the Kullback-Leibler (KL) divergence as a metric between the parametric probability density functions (pdfs). For any pair of measurements \( X_t \) and \( X_{t'} \), the KL divergence between the corresponding parametric pdfs is defined as
\[
D(p(X|\beta_t)||p(X|\beta_{t'})) = \int_X \ln \left( \frac{p(X|\beta_t)}{p(X|\beta_{t'})} \right) \cdot p(X|\beta_t) dX.
\]
(2.2)

Let \( \beta_{t_0} \) and \( \beta_t = \beta_{t_0} + \delta \beta_t \) be two close samples in the intermediate parametric domain. It can be shown (Dahlhaus, 1996) that the KL divergence is locally approximated by the Fisher information metric, i.e.,
\[
D(p(X|\beta_t)||p(X|\beta_{t_0})) \approx \delta \beta_t^T I(\beta_{t_0}) \delta \beta_t
\]
(2.3)
where \( I(\beta_{t_0}) \) is the Fisher information matrix.

We then define the Riemannian manifold \((\mathcal{M}, g)\), where the Fisher metric in (2.3) is associated with the inner product on the manifold tangent plane \( g \) between the
local distributions,

\[ g_{ij}(\beta_t) = \sum_{i,j} \int \frac{\partial \log p(X|\beta_t)}{\partial \beta^i_t} \frac{\partial \log p(X|\beta_t)}{\partial \beta^j_t} p(X|\beta_t) dX \]  

(2.4)

The points residing on \( M \) are parametric probability density functions \( p(X|\beta_t) \). Thus, for \( p(X|\beta_{t0+\delta t}) \) in a local neighbourhood of \( p(X|\beta_{t0}) \), the Fisher metric between these two points can be approximated by \( D(p(X|\beta_t)||p(X|\beta_{t0})) \). Therefore, we use the KL divergence to construct the affinity kernel and build the graph for diffusion maps, thus obtaining diffusion distance approximating a Riemannian metric on the manifold of local distributions. This will be addressed in detail in Section 2.2.3.

For the signal types reported in this chapter (music and EEG), we have empirically found that a simple local Gaussian model (Gaussian mixture model) with zero mean and time evolving covariance matrices can effectively describe the local empirical densities of the selected feature sets of the signals. Thus, the intermediate parameterization \( \beta_t \) is a local covariance matrix \( \Sigma_t \), and the local distribution of \( X_t \) is approximated by \( \mathcal{N}(0, \Sigma_t) \). In this case, the KL divergence can be explicitly written as

\[ D(p(X|\beta_t)||p(X|\beta_{t\prime})) = \frac{1}{2} \text{Tr}(\Sigma^{-1}_{t} \Sigma_{t\prime} - I_N). \] 

(2.5)

Based on the KL divergence, we define a symmetric pairwise affinity function using a Gaussian kernel

\[ k(X_t, X_{t\prime}) = \exp \left\{ \frac{D(p(X|\beta_t)||p(X|\beta_{t\prime}))+D(p(X|\beta_{t\prime})||p(X|\beta_t))}{\varepsilon} \right\} \]

(2.6)

The decay rate of the exponential kernel implies that only pdfs \( p(X|\beta_t) \) within an \( \varepsilon \)-neighborhood of \( p(X|\beta_{t\prime}) \) are taken into account and have non negligible affinity. Thus, we can use the approximation of the KL divergence using the Fisher metric
and obtain that

\[ k(X_t, X_{t'}) \approx \exp \left\{ -\frac{(\beta_t - \beta_{t'})^T (I(\beta_t) + I(\beta_{t'})) (\beta_t - \beta_{t'})}{\varepsilon} \right\} \]  

(2.7)

2.2.3 Laplace operator and diffusion maps

Let \( W \) be a pairwise affinity matrix between the set of measurements \( X_t \), whose \((t, t')\)-th element is given by

\[ W_{t,t'} = k(X_t, X_{t'}). \]  

(2.8)

Based on the kernel, we form a weighted graph, where the measurements \( X_t \) are the graph nodes and the weight of the edge connecting node \( X_t \) to node \( X_{t'} \) is \( W_{t,t'} \). The particular choice of the Gaussian kernel exhibits a notion of locality by defining a neighborhood around each measurement \( X_t \) of radius \( \varepsilon \), i.e., measurements \( X_{t'} \) such that \((\beta_t - \beta_{t'})^T (I(\beta_t) + I(\beta_{t'})) (\beta_t - \beta_{t'}) > \varepsilon \) are weakly connected to \( X_t \). In practice, we set \( \varepsilon \) to be the median of the elements of the kernel matrix. According to the graph interpretation, such a scale results in a well connected graph because each measurement is effectively connected to half of the other measurements. For more details, see Hein and Audibert (2005); Coifman et al. (2008).

Using the KL divergence (the Fisher information metric) as an affinity measure has the effect of an adaptive scale. Consider the parametric family of normal distributions. In particular, assume that the process \( X_t \) is one dimensional and is given by

\[ X_t = Y_t + V_t \]  

(2.9)

where \( Y_t \sim \mathcal{N}(0, \theta_t) \) and \( V_t \) is an adaptive white Gaussian noise with zero mean and fixed \( \sigma_v^2 \) variance. According to the parametric model (Section 2.2.1), \( \theta_t \) follows a diffusion process propagation model which results in time varying distributions. Consequently, the parametric pdf of the measurements is given by

\[ p(X|\beta_t) = \mathcal{N}(0, \beta_t) \]  

(2.10)
where $\beta_t = \theta_t + \sigma^2_v$, and the corresponding Fisher Information matrix is

$$I(\beta_t) = \frac{1}{2\beta_t^2}. \quad (2.11)$$

In this case, the corresponding kernel based on the KL divergence is

$$k(X_t, X_{t'}) = \exp \left\{ -\frac{\|\theta_t - \theta_{t'}\|^2}{\varepsilon(\beta_t, \beta_{t'})} \right\} \quad (2.12)$$

where

$$\varepsilon(\beta_t, \beta_{t'}) = \frac{\varepsilon}{2} \left( \frac{1}{(\theta_t + \sigma^2_v)^2} + \frac{1}{(\theta_{t'} + \sigma^2_v)^2} \right)^{-1} \quad (2.13)$$

is a locally adapted kernel scale with the following interpretation: when the noise rate $\sigma^2_v$ increases, a larger scale (neighborhood) is used in order to see “beyond the noise” and capture the geometry and variability of the underlying parameter $\theta$. We remark that in this specific case, the adaptive scale is the local covariance of the measurements. Thus, this metric is equal to the Mahalanobis distance between the measurements (Talmon and Coifman, 2012).

Let $D$ be a diagonal matrix whose elements are the sums of rows of $W$, and let $W^{\text{norm}} = D^{-1/2}WD^{-1/2}$ be a normalized kernel that shares its eigenvectors with the normalized graph-Laplacian $I - W^{\text{norm}}$ (Chung, 1997). It was shown (Coifman and Lafon, 2006) that $W^{\text{norm}}$ converges to a diffusion operator that reveals the low-dimensional manifold and a subset of its eigenvectors give a parameterization of the underlying process. We assume that these eigenvectors are the principal eigenvectors associated with the largest eigenvalues, although there is no guarantee. Thus, the eigenvectors of $W^{\text{norm}}$, denoted by $\tilde{\Phi}_j$, reveal the underlying structure of the data. Specifically, the $t$-th coordinate of the $j$-th eigenvector can be associated with the $j$-th coordinate of the underlying process $\theta_t$ of measurement $X_t$. The eigenvectors are ordered such that $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{T-1} > 0$, where $\lambda_j$ is the eigenvalue associated with eigenvector $\tilde{\Phi}_j$. Because $W^{\text{norm}} \sim P = D^{-1}W$, and $D^{-1}W$ is row-stochastic,
\(\lambda_0 = 1\) and \(\vec{\varphi}_0\) is the diagonal of \(D^{1/2}\). In addition, \(W_{\text{norm}}\) is positive semidefinite, and hence, the eigenvalues are positive. The matrix \(P\) may be interpreted as a transition matrix of a Markov chain on the graph nodes. Specifically, the states of the Markov chain are the graph nodes and \(P_{t,t'}\) represents the probability of transition in a single Markov step from node \(X_t\) to node \(X_{t'}\). Propagating the Markov chain \(n\) steps forward corresponds to raising \(P\) to the power of \(n\). We also denote the probability function from node \(X_t\) to node \(X_{t'}\) in \(n\) steps by \(p_n(X_t, X_{t'})\).

The eigenvectors are used to obtain a new data-driven description of the measurements \(X_t\) via a family of mappings that are called diffusion maps (Coifman and Lafon, 2006). Let \(\Psi_{\ell,n}(X_t)\) be the diffusion mappings of the measurements into the Euclidean space \(\mathbb{R}^\ell\), defined as

\[
\Psi_{\ell,n}(X_t) = (\lambda_1^n \varphi_1^n, \lambda_2^n \varphi_2^n, \ldots, \lambda_\ell^n \varphi_\ell^n)^T
\]

where \(\ell\) is the new space dimensionality ranging between 1 and \(T-1\). Diffusion maps have therefore two parameters: \(n\) and \(\ell\). Parameter \(n\) corresponds to the number of steps of the Markov process on the graph, since the transition matrices \(P\) and \(P^n\) share the same eigenvectors, and the eigenvalues of \(P^n\) are the eigenvalues of \(P\) raised to the power of \(n\). Parameter \(\ell\) indicates the intrinsic dimensionality of the data. The dimension may be set heuristically according to the decay rate of the eigenvalues, as the coordinates in (2.14) become negligible for a large \(\ell\). In practice, we expect to see a distinct “spectral gap” in the decay of the eigenvalues. Such a gap is often a good indicator of the intrinsic dimensionality of the data and its use is a common practice in spectral clustering methods. The mapping of the data \(X_t\) into the low-dimensional space using (2.14) provides a parameterization of the underlying manifold and its coordinates represent the underlying processes \(\theta_t\) (2.1). We note that as \(n\) increases, the decay rate of the eigenvalues also increases (they are confined in the interval \([0,1]\)). As a result, we may set \(\ell\) to be smaller, enabling to capture the underlying structure of the measurements in fewer dimensions. Thus, we may claim that a larger number of steps usually brings the measurements closer in the sense of the affinity implied by \(P^n\), and therefore, a more “global” structure of the
signal is revealed.

The Markov process aggregates information from the entire set into the affinity metric $p_n(\mathbf{X}_t, \mathbf{X}_{t'})$, defining the probability of “walking” from node $\mathbf{X}_t$ to $\mathbf{X}_{t'}$ in $n$ steps. For any $n$, the following metric

$$D^2_n(\mathbf{X}_t, \mathbf{X}_{t'}) = \int_{\mathbf{X}_s} [p(\mathbf{X}_t, \mathbf{X}_s) - p_n(\mathbf{X}_{t'}, \mathbf{X}_s)]^2 w(\mathbf{X}_s) d\mathbf{X}_s \quad (2.15)$$

is called diffusion distance, with $w(\mathbf{X}_s) = 1/\tilde{\varphi}_0(\mathbf{X}_s)$. It describes the relationship between pairs of measurements in terms of their graph connectivity, and as a consequence, local structures and rules of transitions are integrated into a global metric. If the integral is evaluated on the points of the observed data, it can be shown that the diffusion distance (2.15) is equal to the Euclidean distance in the diffusion maps space when using all $\ell = T - 1$ eigenvectors (Coifman and Lafon, 2006), i.e.,

$$D_n(\mathbf{X}_t, \mathbf{X}_{t'}) = \|\Psi_{\ell,n}(\mathbf{X}_t) - \Psi_{\ell,n}(\mathbf{X}_{t'})\|_2 \quad (2.16)$$

Thus, comparing between the diffusion mappings using the Euclidean distance conveys the advantages of the diffusion distance stated above. In addition, since the eigenvalues tend to decay fast, for large enough $n$, the diffusion distance can be well approximated by only the first few eigenvectors, setting $\ell << T - 1$.

### 2.2.4 Sequential implementation

The construction of the diffusion maps embedding is computationally expensive due to the application of the eigenvector decomposition (EVD). In practice, the measurements are not always available in advance. Thus, the computationally demanding procedure should be applied repeatedly whenever a new set of measurements become available. In this section, we describe a sequential procedure for extending the embedding, which circumvents the EVD applications and may be suitable for supervised techniques (Kushnir et al., 2012; Haddad et al., 2013; Talmon et al., 2012).

Let $\mathbf{X}_t$ be a sequence of $T$ reference measurements that are assumed to be available in advance. The availability of these measurements enables one to estimate the
local densities and the corresponding kernel based on the KL divergence. Then, the embedding of the reference measurements can be computed.

Let $X_s$ be a new sequence of $S$ measurements, which are assumed to become available sequentially. As proposed in Haddad et al. (2013); Talmon et al. (2012), we define a nonsymmetric pairwise metric between any new measurement $X_s$ and any reference measurement $X_t$, similarly to (2.7) as

$$a(X_s, X_t) = \exp \left\{ - \frac{(\beta_s - \beta_t)^T I(\beta_t)(\beta_s - \beta_t)}{\varepsilon} \right\}$$

where $\beta_t$ and $\beta_s$ are the parametrization of the local densities of the measurements at $t$ and $s$, respectively, and a corresponding nonsymmetric kernel

$$A_{s,t} = a(X_s, X_t).$$

The construction of the nonsymmetric kernel requires the feature vectors of the measurements and the Fisher Information matrix of merely the reference measurements.

Let $\tilde{A} = D_A^{-1} A Q^{-1}$, where $D_A$ is a diagonal matrix whose diagonal elements are the sums of rows of $A$, and $Q$ is a diagonal matrix whose diagonal elements are the sums of rows of $D_A^{-1} A$. It was shown by Kushnir et al. (2012); Haddad et al. (2013) that

$$W = \tilde{A}^T \tilde{A}$$

where $W$ is the pairwise affinity matrix on the $T$ reference measurements $X_t$ as defined in Section 2.2.3.

We define now the dual extended $S \times S$ kernel between the new samples as $W_{\text{ext}} = \tilde{A} \tilde{A}^T$. It is shown in Talmon et al. (2012) that the elements of the extended kernel are proportional to a Gaussian defined similarly to (2.7) between a pair of new measurements. Combining the relationship between the kernels $W$ and $W_{\text{ext}}$ yields: (1) the kernels share the same eigenvalues $\lambda_j$; (2) the eigenvectors $\varphi_j$ of $W$ are the right singular vectors of $\tilde{A}$; (3) the eigenvectors $\psi_j$ of $W_{\text{ext}}$ are the left singular
vectors of $\tilde{A}$. As discussed in Section 2.2.3, the right singular vectors represent the underlying process of the reference measurements, and by Kushnir et al. (2012); Haddad et al. (2013), the left singular vectors naturally extend this representation to the new measurements. In addition, the relationship between the eigenvectors of the two kernels is given by the singular value decomposition (SVD) of $\tilde{A}$ and is explicitly expressed by

$$
\psi_j = \frac{1}{\sqrt{\lambda_j}} \tilde{A} \varphi_j.
$$

(2.20)

Now, the extended eigenvectors $\psi_j$ can be used instead of $\tilde{\varphi}_j$ to construct the embedding of the new measurements $\Psi_{t,n}(X_s)$ in (2.14).

2.3 Modeling time evolving covariance matrices

To calculate the KL divergence, we need to estimate the local/intermediate parametric distribution $p(X_t|\beta_t)$ at each time. The amount of data in each time window is limited, and therefore, assumptions have to be made to constrain the parameteric space. In this chapter, we assume that the signal sample at each time is drawn from a multivariate Gaussian distribution with time evolving parameters. For simplicity, we focus on zero mean Gaussian distributions. We assume that the time evolving covariance matrices characterize the dynamics of the time series. Such a time evolving covariance model can be applied to many multivariate time series, including volatility analysis in finance (Durante et al., 2012) and EEG activity in neurology (Fox and West, 2011). Popular approaches for estimating smoothly varying covariance matrices include the exponentially weighted moving average (EWMA) model (Alexander, 2008) and multivariate generalized autoregressive conditional heteroscedasticity (GARCH) models (Bauwens et al., 2006). The former captures the smoothly varying trends, but fails to handle missing data, and requires long series to achieve high estimation accuracy (Tsay, 2005). The latter handles missing data at the expense of over-restricting the flexibility of the dynamics of the covariance ma-
trices (Durante et al., 2012). Most moving average type approaches can be simplified as $\hat{\Sigma}_t = \pi_t \hat{\Sigma}_{t-1} + (1 - \pi_t) \Sigma_t$, where $\Sigma_t$ is the covariance matrix of the sample at $t$ and $\pi_t$ is a smoothing parameter. $\hat{\Sigma}_t$ can be considered as the posterior mean estimate for $\Sigma_t$ using an inverse-Wishart prior with mean proportional to $\hat{\Sigma}_{t-1}^{-1}$. Given their broad use in practice, moving average type approaches are also tested in our experiments.

In this chapter, we present a latent variable model to infer the time evolving covariance matrices, inspired by the idea proposed in Fox and Dunson (2011). Although this generative model cannot reconstruct the observed data samples, it allows for handling missing data and aims to capture the dynamics of the evolving structure of the covariance matrices. We assume at each time $t$, $t = 1, \ldots, T$, a factor analyzer (FA) model fits the observed data sample $X_t \in \mathbb{R}^N$:

$$X_t \sim \mathcal{N}(F^t S_t, \alpha^{-1} I_N)$$ (2.21)

$$S_t \sim \mathcal{N}(0, I_K)$$ (2.22)

$$\alpha \sim \text{Ga}(\epsilon_0, f_0)$$ (2.23)

where $F^t \in \mathbb{R}^{N \times K}$ formulates a time evolving factor loading matrix, with a prior that will be described next. $S_t$ is a $K$ dimensional variable in the latent space, and $\alpha$ models the noise level in the observation space. This model constrains the high-dimensional data to locally reside in a $K$ dimensional space, but does not assume local stationarity due to the time evolving factor-loading matrix. Thus, at time $t$, the marginal distribution of $X_t$ is

$$X_t \sim \mathcal{N}(0_N, \Sigma_t)$$ (2.24)

$$\Sigma_t = (F^t)(F^t)^T + \alpha^{-1} I_N$$ (2.25)

Therefore, we could use posterior estimates of $F^t$ and $\alpha$ to estimate $\Sigma_t$. Because we are only interested in $(F^t)(F^t)^T$, instead of $F^t$ or $S_t$, this latent variable model circumvents the identifiability problem encountered in common FA models.
To impose smoothness and facilitate parameter estimation, two types of priors for $F^t$ are proposed: a Gaussian process (GP) (Rasmussen, 2006) and non-stationary autoregressive (AR) process (Zuur et al., 2003).

In the GP model, elements of $F^t$ are constructed as

$$F_{ij}^t \sim \mathcal{GP}(0, \Omega_{ij}^t)$$

$$\Omega_{i_1,t_2}^{j_1} = \sigma^{-1}(k(t_1, t_2) + \sigma_n \delta(t_1 - t_2))$$

$$\sigma \sim \text{Ga}(c_0, d_0)$$

where $\sigma^{-1}$ represents the variance of the factor loadings over time. To infer $\sigma^{-1}$ from the data, a broad gamma distribution prior is proposed. The hyperparameter $\sigma_n$ represents the noise variance for the factor loadings, which is kept fixed at a small value ($10^{-3}$ in our case). Each time varying factor loading $F_{ij}^t$, $t = 1, \ldots, T$, is constructed from a GP. Various kernel functions $k(t_1, t_2)$ can be used for the GP, including the radius basis function (RBF) $k(t_1, t_2) = \exp\left(-\frac{(t_1 - t_2)^2}{2\tau^2}\right)$. We have tested different kernel functions and RBF is chosen to allow for simple interpretation in our experiments. $\tau$ is the length-scale parameter, which determines globally the shape of autocorrelation function (Fox and Dunson, 2011). This facilitates strong correlation between $F_{ij}^{t_1}$ and $F_{ij}^{t_2}$ if $|t_1 - t_2| < \tau$ and inhibits the correlation otherwise. $\tau$ can be estimated from the data by putting a discrete uniform prior over a library of candidate atoms (Banerjee et al., 2011). However, in practice, the correlation length might be available a priori and used as the appropriate value, which often works effectively. Standard MCMC sampling can be used to infer model parameters, as summarized in the Appendix. Sampling $F_{ij}^t$ from the GP has a $O(T^3)$ complexity, which can be alleviated using the random projection idea in Banerjee et al. (2011).

In the non-stationary AR process prior model, elements of $F^t$ are constructed as

$$F_{ij}^t = F_{ij}^{t-1} + \xi_{ij}^t$$

$$\xi_{ij}^t \sim \mathcal{N}(0, \eta_{ij}^{-1})$$

$$\eta_{ij} \sim \text{Ga}(g_0, h_0)$$
The time varying factor loading matrix $F^t, t = 1, \cdots, T$, is a random walk whose smoothness is determined by $\xi^t$. A shrinkage prior (Chen et al., 2010) favoring $\xi^t$ to be sparse is added to encourage smoothness. Other kinds of sparseness-promoting priors, including spike-slab (Ishwaran and Rao, 2005) and generalized double Pareto (Armagan et al., 2011), can also be considered. The zero mean distribution for $\xi_{ij}$ models the difference between consecutive covariance matrices as a stationary process, which captures the drift of factor loadings over time. In this model, the trend of each factor loading is assumed independent. However, correlated trends of $F^t$ and group sparsity of $\xi^t$ can be considered for more sophisticated data, as a future direction. A forward filtering backward sampling (FFBS) (Frühwirth-Schnatter, 1994) method is used to sample $F^t$ (summarized in the Appendix).

Choosing which estimation method to use is data specific and depends on the available prior knowledge. If the covariance structures are highly correlated for nearby samples, while they are highly uncorrelated for faraway samples, the model with GP prior should be adopted. If the difference between consecutive covariance matrices is approximately a stationary process, the model with non-stationary AR process prior should be considered. If the covariance matrices are evolving smoothly over time, local stationarity approximately holds, and a large number of data samples are provided, moving average type approaches can be easily implemented. This is considered in Section 2.4. The generalized framework can be summarized in Algorithm 3.

2.4 Applications

The proposed framework is applied to a toy example and two real-world applications. In the synthetic toy example, we show that the estimated diffusion distance between data points recovers the geodesic distance on the statistical manifold, where IGDR (Carter et al., 2011) is used as a baseline method. In the first real-world application, we analyze a well-know music piece, where we estimate the diffusion distance between time points to discover the intra-piece similarities as a function of time. In the second
Algorithm 1 Diffusion maps using time evolving statistics

Require: Observations $X_t \in \mathbb{R}^N$, $t = 1, \ldots, T$, diffusion step $n$, neighbourhood size $\epsilon$, embedding dimension $l$

Ensure: Embeddings $\{\Psi_{t,n}(X_t), t = 1, \ldots, T\}$

1: At each time $t$, estimate a distribution $p(X_t|\beta_t)$ (using either moving average (MA), Bayesian generative model with AR process prior (BAR), or Gaussian Process prior (BGP) to infer $\beta_t$)

2: Compute the $T \times T$ matrix $W$, where $W_{t_1,t_2} = \exp \left( -\frac{D(p(X_t|\beta_t)p(X_{t_2}|\beta_t))}{\epsilon} \right)$

3: Set $D = \text{diag} \left\{ \sum_{\tau=1}^{T} W_{t,\tau} \right\}$ and compute the normalized kernel $W^{\text{norm}} = D^{-1/2} WD^{-1/2}$

4: Keep the top $l$ non-trivial eigenvalues $\lambda_j$ and eigenvectors $\tilde{\varphi}_j \in \mathbb{R}^T$ of $W^{\text{norm}}$, $j = 1, \ldots, l$, and construct the corresponding diffusion maps embedding $\Psi_{t,n}(X_t) = (\lambda_1^{n}\tilde{\varphi}_1^{t}, \lambda_2^{n}\tilde{\varphi}_2^{t}, \ldots, \lambda_l^{n}\tilde{\varphi}_l^{t})^T$

application, the proposed framework is used to discover the different brain states of an epileptic patient based on EEG recordings.

2.4.1 Synthetic Experiment

We simulate data $X_t$ from a zero-mean multivariate Gaussian distribution, with time-evolving covariance matrix $\Sigma_t$, constructed from a GP prior as defined in (2.24)-(2.27). We consider observation length $T = 500$, dimension $N = 5$, local latent dimension $K = 3$, and GP length-scale parameter $\tau = 0.02$. The goal is to recover the geodesic distance between data points on the statistical manifold, defined by their corresponding covariance matrices. Because the pairwise symmetric KL distance matrix is needed in both the proposed framework and IGDR, we let both methods know the true $\Sigma_t$ therefore we can focus on the comparison between the two dimensionality reduction schemes. In other words, in this experiment we do not estimate $\Sigma_t$ from the data, but simply assume it is known. The purpose of this test is to compare the diffusion-distance method with IGDR, on the same time-evolving density function.

We apply diffusion maps with the pairwise affinity kernel defined in (2.6). We consider $n = 200$ and obtain the low-dimensional embeddings of $X_t$ as defined in (2.14). The pairwise geodesic distance between data points can be approximated by the Euclidean distances between the corresponding embeddings, as shown in Fig. 2.1(a).
On the other hand, using IGDR, a shortest-path algorithm is executed to find approximate pairwise geodesic distances, followed by classical multidimensional scaling (MDS) methods (i.e., Laplacian eigenmaps in Belkin and Niyogi (2003)) for dimensionality reduction. The approximated pairwise geodesic distances are presented in Fig. 2.1(b) and used as a comparison. As illustrated in Fig. 2.1, both methods yield similar distances. However, the running time of the proposed method is a couple of seconds, whereas the running time of IGDR is approximately 400 seconds, because of the $O(T^3)$ complexity required to compute the pairwise shortest-path distances. These computations were performed on a computer with 2.2GHz CPU and 8GB RAM, with all software written in Matlab.

\[ \begin{align*}
\text{Time [sec]} & \quad 100 \quad 200 \quad 300 \quad 400 \quad 500 \\
\text{Time [sec]} & \quad 50 \quad 100 \quad 150 \quad 200 \quad 250 \quad 300 \quad 350 \quad 400 \quad 450 \quad 500 \\
\text{Time [sec]} & \quad 0 \quad 0.1 \quad 0.2 \quad 0.3 \quad 0.4 \quad 0.5 \quad 0.6 \quad 0.7 \quad 0.8 \quad 0.9 \quad 1 \\
\end{align*} \]

\textbf{Figure 2.1:} (a) Normalized diffusion distance between time points. (b) Normalized approximated geodesic distance between time points.

\subsection{2.4.2 Music analysis}

In music information retrieval, automatic genre classification and composer identification are of increasing interest. Thus, one goal is to compute similarities between short intervals at different times of a music piece (Ren et al., 2011). For comparison with a previous method, we test our framework on the same music piece used in Ren et al. (2011), “A Day in the Life” from the Beatles’ album Sgt. Pepper’s Lonely Hearts Club Band. The song is 5 minutes 33 seconds long and sampled at 22.05 KHz. The music piece is divided into 500ms contiguous frames to obtain Mel Frequency Cepstral Coefficients (MFCCs), which leads to 667 available frames overall.
As depicted in Fig. 2.2 (a), 40 normalized (i.e., with zero mean) MFCCs are used as features, yielding $X_t \in \mathbb{R}^N, t = 1, \ldots, T$, where $N = 40$ and $T = 667$.

In this application we will compare music analysis based on distances computed directly between MFCC feature vectors, and based upon the statistics of MFCC features within a local temporal window. The motivation of this work is that the local statistics of MFCC features within time moving windows constitutes a better representation of the similarities and differences in the music than distances computed directly on the MFCC features. For the former, we must compute distances between time-evolving distributions, which motivates the methods discussed in Section 2.2.

In Fig. 2.2(a) we plot the frequency content of the frames of music, as captured via a spectrogram; the spectrogram frequency content is closely related to the MFCC features. By modeling the evolution in the statistics of multiple contiguous frames of frequency content, the hope is that we capture more specific aspects of the music, than spectral content at one point in time. Specifically, the frequency spectrum at one time point may miss statistical aspects of the music captured by frequency content at neighboring times.

In the proposed model, we assume that the frames have a time evolving distribution parametrized by $\Sigma_t$, as in (2.24). The diagonal elements of $\Sigma_t$ denote the energy content in each frequency band at time $t$, and the off-diagonal elements represent the correlation between different frequency bands. In music, we observe that nearby frames in time usually tend to have similar $\Sigma_t$, whereas, temporally distant frames tend to have different $\Sigma_t$. Thus, the time evolution of $\Sigma_t$ is smooth and modeled with the GP prior. As described in (2.21)-(2.23) and (2.26)-(2.28), the GP prior explicitly encodes this belief and is utilized to estimate $\Sigma_t$. The length-scale parameter is set to $\tau = 5$, the number of local factors is set to $K = 5$, and the other hyperparameters are set to $c_0 = d_0 = e_0 = f_0 = \sigma_n = 10^{-3}$. Empirically, we find that this model fits the data well and the performance is relatively insensitive to the parameter settings (many different parameter settings yielded similar results). A total of 4000 MCMC iterations are performed, with the first 2000 samples discarded as burn-in. The covariance matrix $\Sigma_t$ is calculated by averaging across the collected samples.
Then the pairwise affinity kernel is computed according to (2.5)-(2.6), the diffusion-map algorithm is applied, and the diffusion distances are calculated according to (2.14)-(2.16).

To demonstrate the advantage of organizing the music intervals based on local statistics, we compare the obtained results to diffusion maps using pairwise affinity kernel constructed with Euclidean distances between the MFCC features directly. Additionally, we compare our method to results from the kernel beta process (KBP) (Ren et al., 2011). The KBP does not explicitly learn an embedding, but rather represents the MFCCs at each time in terms of a learned dictionary, yielding a low-dimensional representation. The statistical relationships between pieces of music are defined by the similarity of dictionary usage. The KBP approach models each MFCC frame, but imposes that temporally nearby frames are likely to have similar dictionary usage. The proposed method explicitly utilizes a time-evolving covariance matrix; the subspace defined by that covariance matrix is related to the subspace defined by the KBP dictionary-based method, but the GP model does not explicitly impose dictionary structure (the covariance is allowed to vary more freely).

In Figs. 2.2(b)-(d) we illustrate the intra-piece relationships as a function of time, based on the three approaches considered. The results in Figs. 2.2(b)-(c) were computed via the diffusion-based embedding, where (b) used the proposed method of computing distances between data at two points, and (c) used Euclidian distance between MFCC feature vectors. The results in Fig. 2.2(d) were computed by KBP.

In Figs. 2.2(b)-(c), the relationship between data at times $t$ and $t'$ is represented as $f(d_{tt'}) = \exp\left(-\frac{d_{tt'}}{\delta}\right)$, where $d_{tt'}$ represents diffusion distance, and $\delta = 10^{-4}$. This plots the relationship between the data on a scale of $[0,1]$. In Fig. 2.2(d) the correlation is shown between the probability of dictionary usage at times $t$ and $t'$, as done in Ren et al. (2011).

At the website http://youtu.be/XhDz0npyHmg, one may listen to the music, and examine how the music maps onto the segmentations and relationships in Figs. 2.2(b)-(d). It is evident that diffusion maps based on Euclidian distance applied
directly to the MFCC features, in Fig. 2.2(c), does not capture the detailed temporal relational information of the proposed approach in Fig. 2.2(b) and of the KBP approach in Fig. 2.2(d). Figs. 2.2(b) and (d) are in good agreement with regard to large-scale behavior, but the proposed method in (b) appears to capture more-detailed temporal structure.

For example, interval [10, 100] seconds consists of a solo, dominated by human voice, whereas the subinterval [30, 42] seconds contains a rhythm different from other parts of the piece. KBP analysis in Fig. 2.2(d) is unable to capture the latter detailed structure, but it is clearly evident in the results of the proposed algorithm, in Fig. 2.2(b). The KBP approach seems to be effective in inferring large-scale temporal relationships, but not as good at distinguishing localized, fine-scale temporal differences. The method by which the diffusion analysis is performed appears to be important, as the results in Fig. 2.2(c), in which a simple Euclidian distance was used in the diffusion kernel, yield relatively poor results, missing most large-scale and fine-scale details.

To further illustrate this point, we analyze the performance of the three methods in the short interval [65, 75] seconds in Fig. 2.3. As shown in Fig. 2.3(a), in the interval [68.5, 71.5] seconds, the human voice harmonics disappear, indicating a break in the singing. Comparing the corresponding distances in Figs. 2.3(b)-(d), we find that diffusion maps using time evolving statistics capture this break, whereas KBP fails to capture this break. Although diffusion maps based on Euclidean distances between the features capture the break, other false breaks are captured as well. Similar trends and performance can be also found in the intervals [97, 99] seconds and [223, 225] seconds.

The diffusion mapping formulates an embedding space that discovers the low-dimensional manifold of the data. Fig. 2.4 depicts two coordinates of the mapping (2.14) corresponding to the eigenvectors associated with the largest two non-trivial eigenvalues. For evaluation, we annotated the song with markers indicating whether human voice appears. As depicted in the figure, we find that the second coordinate
correlates with human voice. For instance, we observe that the second coordinate is significantly large during the interval [10, 100] seconds, which consists of the voice solo. In addition, the first coordinate correlates with the overall background sounds: it takes small values for the first 312 seconds of the song, and then exhibits a sudden increase when peaky humming appears. Such information can be utilized to interpret the similarity between frames and may be used for other music-analysis tasks. This suggests that the coordinates of the embedding, i.e., the eigenvectors of the graph, indeed represent the underlying factors controlling the music. See http://youtu.be/4uPaLgbMSQw for an audio-visual display of these results.

Figure 2.2: Analysis results of the song “A Day in the Life”. (a) Spectrogram of the song. (b)-(d) Comparison of the pairwise similarity matrices obtained by (b) diffusion maps using time evolving statistics, (c) diffusion maps using Euclidean distances between features, and (d) kernel beta process Ren et al. (2011).
Figure 2.3: Analysis results in the subinterval [65, 75] seconds of the song “A Day in the Life”. (a) Spectrogram of the song. (b)-(d) Comparison of the pairwise similarity matrices obtained by (b) diffusion maps using time evolving statistics, (c) diffusion maps using Euclidean distances between features, and (d) kernel beta process Ren et al. (2011).

2.4.3 Epileptic seizure prediction

We now consider epileptic-seizure prediction based on EEG recordings. It is important and desirable to predict seizures so that patients can be warned a few minutes prior. Many studies have been conducted in order to devise reliable methods that can distinguish interictal and preseizure states (Frei et al., 2010). Recent literature suggests that the correlation between frequency components of EEG signals indicate the underlying brain state (Gadhoumi et al., 2013; Alvarado-Rojas et al., 2011; ?). However, because EEG recordings tend to be very noisy, and because the brain activity states and their relationship to the EEG activities are unknown, it is considered
Figure 2.4: The 2 principal eigenvectors as a function of time compared to human voice activity indicator

a difficult problem without existing solutions (Talmon and Coifman, 2012; Duncan et al., 2013).

In the present work, we consider intracranial EEG (icEEG) data collected from a patient at the Yale-New Haven Hospital. Multiple electrode contacts were used during the recording; in this work, we focus on the contacts located at the seizure onset area in the right occipital lobe. Discovering the relationships between different areas will be a subject of future research. We study four 60-minute long icEEG recordings with a sampling rate of 256 Hz, each containing a seizure. A detailed description of the collected dataset can be found in Duncan et al. (2013).

Figure 2.5 presents a 60-minute EEG recording (Fig. 2.5(a)) and its short time spectrogram (Fig. 2.5(b)). As shown in Fig. 2.5(a), the seizure occurs after about 56 minutes in this recording, and is visible. However, it is difficult by observation to notice differences between recording parts that immediately precede the seizure and parts that are located well before the seizure. Our goal is, first, to infer a low-dimensional representation of the recordings, which discovers the intrinsic states (i.e., the brain activities) of the signal. By relying on such a representation, we detect
anomaly patterns prior to the seizure onsets (preseizure states) and distinguish them from samples recorded during resting periods (interictal state), thereby enabling prediction of seizures.

The short time Fourier transform (STFT) with a 1024 sample window and 512 sample overlap is applied to obtain features in the frequency domain. The amplitude of frequency components in Delta (0.5-4 Hz) and Theta (4-8 Hz) bands, with 0.25 Hz spacing, are collected for each time frame and used as feature vectors in the following experiments. Thus, the feature vectors $X_t \in \mathbb{R}^{32}$ of the data in the frequency domain are obtained, as shown in Fig. 2.5(b). The Beta (13-25 Hz) and Gamma (25-100 Hz) bands were also included but empirically showed no significant contribution. In this experiment, two 5-minute intervals from each recording are analyzed: one is the 5-minute interval immediately preceding the seizure onset (preseizure state), and the other is a 5-minute interval located 40 minutes before the seizure (interictal state). Therefore, for each 5-minute interval, we have a set of vectors $X_t \in \mathbb{R}^N$, $t = 1, \ldots, T$, where $N = 32$ and $T = 149$. The obtained features are centered, and hence, each $X_t$ is considered as a sample from a zero-mean multivariate Gaussian distribution $\mathcal{N}(0_N, \Sigma_t)$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.5a.png}
\caption{(a) A sample recording. The red line marks seizure’s onset, at approximate 56 minutes. (b) The STFT features of the recording.}
\end{figure}
In this application, unlike the prior knowledge we have in the music analysis case, we do not have a reliable notion of the way the covariance matrices are correlated at different times. Thus, the only belief we can incorporate into the generative model is that $\Sigma_t$ are smoothly varying. This information is encoded in two priors, which are used to estimate $\Sigma_t$: A Bayesian approach with a latent variable model using a non-stationary AR process prior (BAR) and an empirical approach using the moving averaging (MA). The former assumes stationarity of the differences between consecutive covariance matrices. The latter assumes local stationarity, i.e., feature vectors within a small window are generated from the same Gaussian distribution.

In our experiments, both models (BAR and MA) are used for covariance matrix estimation in each 5-minute interval. The BAR model is implemented according to (2.21)-(2.23) and (2.29)-(2.31), where the number of local factors is set to $K = 5$ and the hyperparameters are set to $e_0 = f_0 = g_0 = h_0 = 10^{-3}$; 4000 MCMC iterations are performed, with the first 2000 discarded as burn-in. The collection samples are averaged to estimate $\Sigma_t$. Under the MA model, we simply estimate the covariance directly based on the features in a local neighborhood in time, i.e.,

$$\hat{\Sigma}_t^M = \frac{\sum_{s=t-M}^{t+M} X_s X_s^T}{(2M + 1)}$$

where $2M + 1$ is the length of the window in which local stationarity is assumed to hold. We need to choose $M$ large enough to provide an accurate estimate of $\Sigma_t$ while small enough to avoid smoothing out the local varying statistics. In practice, we set $M = 32$ according to an empirical criterion of being the smallest value that yields sufficient smoothness, formulated by the admittedly ad hoc criterion:

$$\frac{\sum_{t=1}^{T} ||\Sigma_t^M - \Sigma_t^{M+1}||_2}{\sum_{t=1}^{T} ||\Sigma_t^M||_2} \leq 0.05.$$  

Circumventing the need for such criteria is one reason the BAR approach is considered.

In Fig. 2.6, we present the obtained embedding (by applying Algorithm 3) of multiple intervals of an EEG recording (indexed as recording 1) using diffusion maps based on the time-evolving statistics. We embed three 5-minute intervals: one pre-seizure interval and a couple of interictal intervals (located 30 and 40 minutes prior to the seizure onset, respectively). The presented scatter plot shows the embedded
samples in the space formulated by the 3 principal eigenvectors (setting $\ell = 3$ in (2.14)), i.e., each 3 dimensional point corresponds to the diffusion map of a single feature vector $X_t$.

We observe that under both models (BAR and MA) the low-dimensional representation separates samples recorded in the preseizure state (colored red) from samples recorded in interictal states (colored blue and green). In both Figs. 2.6(a) and (b), the embedded samples of the two interictal intervals are located approximately in the same region, with the embedded samples of one of the interictal intervals (colored green) tend slightly towards the embedded samples of the preseizure interval. This result exemplifies the ability of the proposed method to discover the underlying states. Indeed, without prior knowledge of the labeling of the different intervals (pre-seizure or interictal) and based merely on the recorded signal, the proposed method organizes the signal according to the intrinsic state of the patient.

We remark that the embeddings obtained under MA and BAR models, in Figs. 2.6(a) and (b), respectively, are different. Under the BAR modeling, the method tends to recover trajectories with similar evolving patterns due to the strong time correlation assumption. On the other hand, under the MA modeling, the method tends to form point clouds resulting from the local Gaussian distribution assumption.

**Figure 2.6**: Scatter plots of the embedded samples of three 5-minute intervals of EEG recording 1. Each point is the diffusion map of the features of each time frame in the recording, setting $\ell = 3$. The colors indicate the different intervals. (a) The embedded samples under the MA modeling. (b) The embedded samples under the BAR modeling.

We now test the consistency and extensibility of the obtained low-dimensional
representation. In practice, it is desirable to learn the mapping from reference recordings, and then, when new recordings become available to embed them into the low-dimensional space in an online fashion in order to identify and predict seizures.

Figure 2.7 depicts the embedded samples obtained by applying Algorithm 2 using recording 1 as the reference set and recording 2 as the new incoming set. As observed, the new incoming samples are embedded into the same regions as the reference samples from corresponding interval types. For example, we observe in Fig. 2.7(a) that the new samples of the interictal state interval are embedded close to the reference samples of the interictal state interval. In addition, we observe that the samples of the interictal state intervals are embedded around the origin, whereas the samples of the preseizure intervals are embedded further away from the origin, suggesting that the preseizure state intervals are “anomalies” that tend to stick out from the learned “normal state” model. These properties allow for a simple identification of preseizure samples: preseizure labels can be assigned to new samples when their corresponding embedded points are near the region of preseizure reference samples and far from the origin. As shown in Fig. 2.7(b), under the BAR modeling, the embedded points have a different representation. In this case, the embedded samples from different intervals form different trajectories. Nevertheless, we can assign labels to new incoming points in a similar manner - based on their distance to the reference trajectories of each state.

From the presented results in Figs. 2.6 and 2.7, we conclude that the proposed method indeed discovers the brain activity and enables, merely by visualization in 3 dimensions, to distinguish preseizure states from interictal states. Furthermore, the results imply that the coordinates of the diffusion embedding, i.e., the eigenvectors of the constructed kernel, have a real “meaning” in this application. Similarly to the music application, where the coordinates indicate, for example, the human voice, here they capture different aspects of the intrinsic state of the patient. In the present work, we exploit this representation and devise a simple classification procedure to identify preseizure states, which enables to predict seizures. We remark that a simple algorithm is sufficient since the embedding already encodes the required information.
Figure 2.7: Scatter plots of the embedded samples of four 5-minute intervals of EEG recordings: two reference intervals (preseizure and interictal) from recording 1 and two new incoming intervals (preseizure and interictal) from recording 2. The representation is obtained using the reference samples and then extended to the new incoming samples according to Algorithm 2. The dimension of the embedding is set to $\ell = 3$ for visualization. (a) The obtained embedding under the MA modeling. (b) The obtained embedding under the BAR modeling.

and enables to distinguish the different states.

We repeat the experiment described above, and extend the model to three unseen recordings according to Algorithm 2. As before, recording 1 is used as reference set to learn the model, which in turn is extended to the other three recordings (2-4). In each recording, there is a preseizure state interval whose $T$ samples are labeled as “preseizure” and an interictal interval whose $T$ samples are labeled as “interictal”. By using the labels of the reference samples as training data, we train standard linear classifiers in the low-dimensional embedding space to distinguish samples recorded in preseizure states from samples recorded in interictal states. In our algorithm, the classification boundary is the hyperplane lying in the middle of the two empirical means of the embedded reference samples of each state.

Table 2.1 summarizes the detection rate and false alarm rate of the samples from all the recordings. As can be seen, both implementations relying on the MA and BAR modeling perform well in classifying samples into preseizure and interictal states. In general, a larger portion of data samples in the preseizure state interval are correctly identified, while only a small fraction of data samples in the interictal state are identified as preseizure state. Further, we find BAR modeling has an
Algorithm 2 Sequential implementation of diffusion maps based on time evolving statistics

**Require:** Reference observations \( \{ X_t \}, t = 1, \cdots, T \), new incoming observations \( \{ X_s \}, s = 1, \cdots, S \), diffusion step \( n \), neighbourhood size \( \epsilon \), dimension \( \ell \)

**Ensure:** Embeddings \( \{ \Psi_{t,n}(X_t) \}, t = 1, \cdots, T \) and \( \{ \Psi_{t,n}(X_s) \}, s = 1, \cdots, S \)

1: Estimate distribution \( p(X_t|\beta_t) \) for all time points (using moving average (MA), Bayesian model with AR process prior (BAR) or Bayesian model with Gaussian process prior (BGP) to infer \( \beta_t \))
2: Compute the \( T \times T \) matrix \( W \), where \( W_{t_1,t_2} = \exp \left( -\frac{D(p(X_t|\beta_t)||p(X_{t_2}|\beta_{t_2}))}{\epsilon} + D(p(X_t|\beta_t)||p(X_{t_1}|\beta_{t_1})) \right) \)
3: Apply eigenvalue decomposition to \( W \) and keep \( \ell \) principle eigenvalues \( \lambda_j \) and eigenvectors \( \varphi_j \)
4: For new incoming data \( X_s, s = 1, \cdots, S \), estimate the distribution \( p(X_s|\beta_s) \) (using MA, BAR or BGP).
5: Compute the \( S \times T \) nonsymmetric kernel matrix \( A \), where \( A_{s,t} = \exp \left( -\frac{D(p(X_t|\beta_t)||p(X_s|\beta_s))}{\epsilon} \right) \)
6: Construct \( R = D_A^{-1}A \), where \( D_A = \text{diag} \left\{ \sum_{t=1}^{T} A_{s,t} \right\} \)
7: Construct \( \tilde{A} = RQ^{-1} \), where \( Q = \text{diag} \left\{ \sum_{t=1}^{T} R_{s,t} \right\} \)
8: Calculate the diffusion maps embeddings of the new incoming samples \( \Psi_{t,n}(X_s) = (\psi_1^s, \psi_2^s, \ldots, \psi_T^s)^T \), where \( \psi_j = \frac{1}{\sqrt{\lambda_j}} \tilde{A} \varphi_j \)
9: Calculate the diffusion maps embeddings of the reference samples \( \Psi_{t,n}(X_t) = (\lambda_1 \varphi_1^t, \lambda_2 \varphi_2^t, \ldots, \lambda_T \varphi_T^t)^T \)

overall higher detection rate while MA has a lower false alarm rate. One reason is in MA, we assume local Gaussian distribution of data samples and smooth variation of controlling parameters, which inhibit sudden changes, thus reducing the probability of detecting anomaly samples. The other reason is the Nyström method used in Algorithm 2 (Step 8) to embed new incoming data samples has the effect of shrinking coordinates’ amplitude (in the reference set, more embeddings lie close to the origin than far away from it). In MA, this causes identifying more new incoming samples as interictal state because of the reference set, interictal state samples are embedded around origin (see Fig. 2.7). While in BAR, we assume similar evolving patterns of data samples, organizing samples from two state intervals into two trajectories. Therefore, identifying states of new incoming data samples is not effected by the shrinking effect of Nyström method.
An interesting result is that the large portion of embedded samples from a pre-seizure interval actually reside in the interictal state region. Such a result was observed in other experiments on human collected data. It implies that during the preseizure interval, the subject’s brain tries to maintain the normal state and resist the seizure. As a result, the states of the samples alternate rapidly between normal and anomaly states. Thus, to effectively predict seizures, relying on single samples/time frames is insufficient, and an online algorithm that aggregates a consecutive group of samples is required. For example, it is evident from the results in Table 2.1 that tracking the percentage of anomaly samples within a 5-minute interval may be adequate: if the percentage of anomalies is greater than a predefined threshold of 0.35, a prediction of seizure is reported. Designing more sophisticated classification algorithms and testing them on larger dataset with multiple subjects will be addressed in future work.

Table 2.1: Detection rate and false alarm rate of the EEG samples from all the recordings

<table>
<thead>
<tr>
<th>Model</th>
<th>MA</th>
<th>BAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Detection</td>
<td>False Alarm</td>
</tr>
<tr>
<td>1</td>
<td>94.3</td>
<td>26.2</td>
</tr>
<tr>
<td>2</td>
<td>41.8</td>
<td>20.5</td>
</tr>
<tr>
<td>3</td>
<td>62.3</td>
<td>19.7</td>
</tr>
<tr>
<td>4</td>
<td>48.4</td>
<td>19.7</td>
</tr>
</tbody>
</table>

2.5 Conclusions

A dimensionality-reduction method for high-dimensional time series is presented. The method exhibits two key components. First, multiple approaches to estimate time evolving covariance matrices are presented and compared. Second, using the Kullback-Leibler divergence as a distance metric, diffusion maps are applied to the probability distributions estimated from samples, instead of samples themselves, to obtain a low-dimensional embedding of the high-dimensional time series. Theoretical and experimental results show that the embedding inferred by this method discovers the underlying factors, which govern the observations, and preserves the geodesic
distance between samples on the corresponding statistical manifold. Encouraging results are obtained in two real-world applications: music analysis and epileptic seizure prediction. Especially for the latter application, an online seizure identification system is developed, showing the possibility of predicting epileptic seizures based on time evolving statistics of EEG recordings. In future work, we will propose models capturing higher order time evolving statistics beyond the covariance matrices.

2.6 Appendix

For ease of notation, we use \( p(\beta | \sim) \) to denote the posterior probability of the variable \( \beta \) conditional on all other variables and observations.

Sample the factor loadings \( F_{ij}^{1:T} \) with GP prior,

\[
p(F_{ij}^{1:T} | \sim) \propto \mathcal{N}(F_{ij}^{1:T}; 0, \sigma^{-1}\Omega) \prod_{t=1}^{T} \mathcal{N}(X_t; F^tS_t, \alpha^{-1}I_N) \tag{2.32}
\]

\[
\propto \mathcal{N}(\mu_{ij}, \Sigma_{ij})
\]

where \( \Sigma_{ij} = ((\sigma^{-1}\Omega)^{-1} + \text{diag}(S_{j1}^2, \ldots, S_{jT}^2))^{-1}, \mu_{ij} = \Sigma_{ij} \text{diag}(S_{j1}, \ldots, S_{jT})X_i^{-j} \) and \( X_i^{-j} = X_i^{\cdot} - \sum_{k \neq j} F_{ik}^sS_{kt} \)

Sample the precision of factor loadings \( \sigma \),

\[
p(\sigma | \sim) \propto \text{Ga}(a_0, b_0) \prod_{ij} \mathcal{N}(F_{ij}^{1:T}; 0, \sigma^{-1}\Omega) \tag{2.33}
\]

\[
\propto \text{Ga}(a_0 + \frac{TNK}{2}, b_0 + \frac{1}{2} \sum_{ij} F_{ij}^{1:T^T}\Omega^{-1}F_{ij}^{1:T})
\]

where \( \Omega_{i,j} = \exp(-\frac{(t_i - t_j)^2}{2\sigma^2}) + \sigma_n \delta(t_i - t_j) \)

Sample the latent factors \( S_t \),

\[
p(S_t | \sim) \propto \mathcal{N}(0, I_K)\mathcal{N}(X_t; F^tS_t, \alpha^{-1}I_N) \tag{2.34}
\]

\[
\propto \mathcal{N}((I_K + \alpha(F^t)^tF^t)^{-1} \alpha(F^t)^tX_t, (I_K + \alpha(F^t)^tF^t)^{-1})
\]
Sample the observation noise $\alpha$,

\[
p(\alpha |) \sim \mathcal{Ga}(\epsilon_0, f_0) \prod_{t=1}^{T} \mathcal{N}(X_t; F^t S_t, \alpha^{-1} I_N) \tag{2.35}
\]

\[
\mathcal{Ga}(\epsilon_0 + NT/2, f_0 + 1/2 \sum_{t=1}^{T} (X_t - F^t S_t)^T (X_t - F^t S_t))
\]

Sample the factor loadings $F^{1:T}$ with AR prior using FFBS method, using $f_t^i$ to denote a column vector $F^t_{ij}$, $j = 1, \cdots, K$.

Forward step: for $i = 1, \cdots, N$, update the mean $m_t$ and covariance $M_t$ for $f_t^i$ forward in time $t = 1, 2, \cdots, T$.

\[
f_{t}^{i} | X_{i1,\cdots, it} \sim \mathcal{N}(m_{t}, M_{t})
\]

\[
m_t = m_{t-1} + \frac{S_t^T (M_{t-1} + \Gamma)}{S_t^T (M_{t-1} + \Gamma) S_t^T + \frac{1}{\alpha}} (X_{it} - S_t^T m_{t-1})
\]

\[
M_t = M_{t-1} + \Gamma - \frac{(M_{t-1} + \Gamma) S_t S_t^T (M_{t-1} + \Gamma)}{S_t^T (M_{t-1} + \Gamma) S_t^T + \frac{1}{\alpha}}
\]

\[
\Gamma = \text{diag}(\xi_{i1}, \xi_{i2}, \cdots, \xi_{iK})
\]

Backward step: for $i = 1, \cdots, N$, sample $f_t^i$ backward in time $t = T, T - 1, \cdots, 1$.

\[
p(f_t^i |) \sim \mathcal{N}(f_t^i; m_t, M_t) \mathcal{N}(f_t^{i+1}; f_t^i, \Gamma)
\]

\[
\mathcal{N}((M_t^{-1} + \Gamma^{-1})^{-1} (M_t^{-1} m_t + \Gamma^{-1} f_t^{i+1}), (M_t^{-1} + \Gamma^{-1})^{-1}) \tag{2.37}
\]
Partial AUC Optimization with Latent Variables for Time Series Classification

3.1 Introduction

Time series are ubiquitous, ranging from video processing to biomedical applications. Among the time series analysis problems, sequence classification is an important issue, for example, event/activity detection (Tang et al., 2012), anomaly detection (Lian et al., 2015b), and object classification (Kim and Pavlovic, 2011). To deal with the high-dimensionality and auto-correlation nature of time series, existing methods in general fall into two categories: dimension reduction/feature extraction followed by a standard classifier design (Lian et al., 2015b; Hyndman et al., 2015), or incorporating temporal correlation during classifier learning (Tang et al., 2012; Kim and Pavlovic, 2011). The former class of methods have flexibility in feature design but requires domain knowledge on how to construct features from the raw data which has the most discriminative power, e.g., the spikiness and seasonality (Hyndman et al., 2015). Meanwhile, the latter methods directly model the temporal structure of the sequences and automatically discover the discriminative features. In addition,
because of this one-stage design, it usually has performance guarantee under certain model assumptions (Kim and Pavlovic, 2011).

Max-margin based approaches are widely used in discriminative learning, including sequence classifications problems (Tang et al., 2012; Kim and Pavlovic, 2011). These approaches learn a decision boundary which minimizes the loss incurred by misclassification or computed from the distances between data points and the decision boundary. However, in many real-world applications, different misclassification errors should be treated in a different manner. For example, in fraud detection problems based on user logs, it is acceptable if the algorithm misclassifies a few fraud users as normal users; but the cost is huge if it raises a false alarm, i.e., labeling a normal user as a fraud. Such low false alarm rate requirements also exist in other scenarios including proactive recommendation systems (Melguizo et al., 2007) and biomedical applications (Barbu et al., 2006).

Motivated by such cost-sensitive classification tasks, in this work, we design a cost-sensitive objective function using the concept of partial area under the curve (PAUC) (Narasimhan and Agarwal, 2013). The PAUC is a classification performance measure focusing on a subregion in the receiver operating characteristic (ROC) curve, e.g., false alarm rate constrained within $[0, \beta]$. Directly optimizing a loss derived from the PAUC learns a decision boundary making fewer large cost errors (misclassifying a negative example as positive). Our main contribution is two-folds: first, we propose a PAUC optimization framework for sequence classification problems, and build connections with other methods including Hidden Markov Model (HMM) based classifiers; second, we derive a discriminative training approach and an efficient learning algorithm. Simulation experiments are provided to demonstrate the benefits of the proposed method.
3.2 Problem Formulation

The data is collected as a set of observation pairs: \( \{ (x_i, y_i) \}_{i=1}^{N} \), where \( x_i \in \mathbb{R}^P, y_i \in \{-1, +1\} \). The sequences \( \{ x_i \} \) may have different lengths, but for notation simplicity we use a universal sequence length \( T \) throughout this chapter. The goal is to learn a classifier to assign labels \( \hat{y} \) to new time series observations \( x_{st} \). Specifically, we want to learn a scoring function \( g(\cdot) \), which, given a threshold \( \gamma \), assigns \( \hat{y} = +1 \) if \( g(\cdot) > \gamma \) and \( \hat{y} = -1 \) otherwise. Therefore given a scoring function and a threshold, the true positive (TP) rate and false positive (FP) rate can be defined (Narasimhan and Agarwal, 2013),

\[
\begin{align*}
TP_g(\gamma) &= Pr_{x_i \sim D^+}(g(x_i) > \gamma), \\
FP_g(\gamma) &= Pr_{x_j \sim D^-}(g(x_j) > \gamma).
\end{align*}
\]

Empirically, the AUC can be evaluated on a finite dataset as,

\[
AUC_g = \mathbb{P}_{x_i \sim D^+, x_j \sim D^-}(g(x_i) > g(x_j)) = \frac{1}{N^+ N^-} \sum_{i:y_i=1} \sum_{j:y_j=-1} \mathbb{I}(g(x_i) > g(x_j)),
\]

(3.1)

where \( N^+, N^- \) denote the number of positive and negative instances in the dataset. Because we want to optimize the classification performance in a low false alarm rate range \( [0, \beta] \), the PAUC is defined as the objective,

\[
PAUC_g = \frac{1}{N^+ \beta N^-} \sum_{i:y_i=1} \sum_{j:y_j=-1} \mathbb{T}(x_j) \mathbb{I}(g(x_i) > g(x_j)),
\]

(3.2)

\[
\mathbb{T}(x_j) = \mathbb{I}(|\{k : y_k = -1 \text{ and } g(x_k) > g(x_j)\}| \leq \beta N^-).
\]

(3.3)

The hinge loss is used as an objective to maximize the empirical PAUC in (3.2). Using \( w \) to denote the parameters in the scoring function \( g(\cdot) \), e.g., \( g(x) = w^T \phi(x) \),
we have,

\[ l_{P\text{AUC}}(\mathbf{w}) = \frac{\lambda}{2} \| \mathbf{w} \|^2 + \frac{1}{N^+ N^-} \sum_{i:y_i=1,j:y_j=-1} \sum_{t} \mathbb{T}(\mathbf{x}_t)(1 - [g(\mathbf{x}_i) - g(\mathbf{x}_j)])^+, \quad (3.4) \]

where \( \lambda \) is the regularizer on the model parameters \( \mathbf{w} \). The key issue is how to learn the scoring function incorporating the temporal information.

We introduce a set of hidden variables \( z_{it} \in \{1, \cdots, K\} \) associated with each time point \( t \) for a given sequence \( \{\mathbf{x}_i\}_{t=1}^T \), and \( z_{it} \) can be considered as a discrete state. Similar to HMMs, each state \( k \) has a set of parameters \( \theta_k \) used to measure the compatibility of an observation \( \mathbf{x}_{it} \) and the \( k \)-th state. The \( K \) states are assumed to be shared by both positive and negative classes; however, assigning different class labels to the observation sequence \( \{\mathbf{x}_i\}_{t=1}^T \) may result in different state sequences \( \{z_i\}_{t=1}^T \), especially in noisy scenarios. The inferred states can be used to construct augmented features for classification tasks as will be shortly elaborated.

Using the idea of the hidden conditional random field (HCRF) (Quattoni et al., 2007), we define a potential function \( \psi(y, z, x; \Theta) \) and a larger potential indicates better compatibility of the tuple \( \{y, z, x\} \). But for classification tasks, the compatibility of the pair \( \{y, x\} \) is needed. One approach is model averaging (Quattoni et al., 2007), where \( \psi(y, z, x; \Theta) \) is integrated over all possible state paths \( z \) to get an averaged potential \( \psi'(y, x; \Theta) \) as the compatibility measure. An alternative is model selection (Wang and Mori, 2011) where the most probable hidden state sequence \( z \) is selected, as adopted in our work. We define a function \( f(y, x) \) as,

\[ f(x, y; \Theta) = \max_z \psi(y, z, x; \Theta). \quad (3.5) \]

Therefore, we define the scoring function used in classification as,

\[ g(x) = f(x, +1; \Theta) - f(x, -1; \Theta), \quad (3.6) \]
where the implicit dependency of $g(\cdot)$ on $\Theta$ is omitted for brevity. A positive label is assigned if $g(x) > \gamma$ and vice versa. We did not pursue the direction of using model averaging in the scoring function as,

$$f'(x, y; \Theta) = \sum_z p(z; y, x, \Theta) \psi(y, z, x; \Theta), \quad (3.7)$$

where $p(z; y, x, \Theta)$ is derived from the potential function $\psi(y, z, x; \Theta)$ (e.g., transformation with the softmax function). This is also a sensible approach for classification tasks, which is left as an interesting future direction.

The potential function $\psi(y, z, x; \Theta)$ is generally flexible to design and application specific; here we provide the construction used in our work assuming $\psi(y, z, x; \Theta) = w^T \phi(y, z, x; \Theta) (\Theta = \{w\})$. In the current feature construction, $\phi(x, z, y)$ has dimension $(2K^2 + 2K + PK)$, defined as,

$$\phi(x, z, y) = (3.8)$$

$$\sum_{t=1}^{T-1} \mathbb{I}(z_t = 0, z_{t+1} = 0, y = -1), \sum_{t=1}^{T-1} \mathbb{I}(z_t = 0, z_{t+1} = 1, y = -1), \cdots, \sum_{t=1}^{T-1} \mathbb{I}(z_t = K-1, z_{t+1} = K-1, y = -1),$$

$$\sum_{t=1}^{T-1} \mathbb{I}(z_t = 0, z_{t+1} = 0, y = 1), \sum_{t=1}^{T-1} \mathbb{I}(z_t = 0, z_{t+1} = 1, y = 1), \cdots, \sum_{t=1}^{T-1} \mathbb{I}(z_t = K-1, z_{t+1} = K-1, y = 1),$$

$$\sum_{t=1}^{T} \mathbb{I}(z_t = 0, y = -1), \cdots, \sum_{t=1}^{T} \mathbb{I}(z_t = K-1, y = -1), \sum_{t=1}^{T} \mathbb{I}(z_t = 0, y = 1), \cdots, \sum_{t=1}^{T} \mathbb{I}(z_t = K-1, y = 1),$$

$$\sum_{t=1}^{T} x_t^T \mathbb{I}(z_t = 0), \cdots, \sum_{t=1}^{T} x_t^T \mathbb{I}(z_t = K-1)^T.$$

The first $2K^2$ terms (the second and third lines in (3.8)) are to indicate the compatibility between the transition patterns and the sequence label. The middle $2K$ terms (the fourth line in (3.8)) are used to measure the compatibility between the state assignments and the sequence label (how often a state appears in a sequence label as positive/negative). The last $PK$ terms are used to compute how compatible the observation vectors are to the state assignments. Therefore, we have completely
defined the scoring function,

\[ g(x) = \max_z w^T \phi(x, z, +1) - \max_z w^T \phi(x, z, -1). \] (3.9)

Plugging (3.9) into the PAUC loss function defined in (3.4), we have the objective function to minimize solving for \( w \),

\[
  l(w) = \frac{\lambda}{2} ||w||^2 + \sum_{i:y_i=+1} \sum_{j:y_j=-1} T(x_j) [1 + \max_z w^T \phi(x_j, z, +1) \\
  - \max_z w^T \phi(x_j, z, -1) - \max_z w^T \phi(x_i, z, +1) + \max_z w^T \phi(x_i, z, -1)]^+. \] (3.10)

3.3 Model Learning

The objective in (3.10) is semi-convex (Felzenszwalb et al., 2010) in \( w \) because of the terms \( \{- \max_z w^T \phi(x_i, z, +1), i : y_i = +1\} \) and \( \{- \max_z w^T \phi(x_j, z, -1), j : y_j = -1\} \). Semi-convexity here refers to that, if we fix the state sequence \( z \) for positive examples with positive label assignment and negative examples with negative label assignment, the objective becomes convex. Therefore, we can alternate between the following two steps:

**Step 1:** Fix \( w \), calculate \( z_i^{+1} = \max_z w^T \phi(x_i, z, +1), i : y_i = +1 \) and \( z_j^{-1} = \max_z w^T \phi(x_j, z, -1), j : y_j = -1 \).

**Step 2:** Fix \( z_i^{+1}, i : y_i = +1 \) and \( z_j^{-1}, j : y_j = -1 \), and solve the following convex subproblem,

\[
  \min_w \sum_{i:y_i=+1} \sum_{j:y_j=-1} T(x_j) [1 - w^T \phi(x_j, z_j^{-1}, -1) - w^T \phi(x_i, z_i^{+1}, +1) \\
  + \max_z w^T \phi(x_i, z, -1) + \max_z w^T \phi(x_j, z, +1)]^+. \] (3.11)

For the subproblem in (3.11), multiple approaches exist including subgradient methods and cutting plane methods (Joachims et al., 2009), of which we are using
the former. For each positive and negative example pair in (3.11), the function value and gradient can be easily computed,

\[
\begin{align*}
    l'(x_i, x_j; w) &= (1 + \max_z w^T \phi(x_j, z, 1) - w^T \phi(x_j, z^0_j, -1) - w^T \phi(x_i, z^1_i, 1) + \max_z w^T \phi(x_i, z, -1)) + \\
    \nabla l'(x_i, x_j; w) &= \mathbb{I}((w^T \phi(x_i, z^1_i, 1) - \max_z w^T \phi(x_i, z, -1)) - (\max_z w^T \phi(x_j, z, 1) - w^T \phi(x_j, z^0_j, -1)) < 1) \\
    &\quad [\max_z \phi(x_j, z, 1) - \phi(x_j, z^0_j, -1) - \phi(x_i, z^1_i, 1) + \max_z \phi(x_i, z, -1)]. \tag{3.12}
\end{align*}
\]

Following the proposed two-step alternating algorithm, the objective in (3.10) is guaranteed to converge to a local optima. To see this, we first define an auxiliary objective \( l(w, Z^*) \), where \( Z^* \) denotes the hidden state paths for positive examples with positive label assignment and negative examples with negative label assignment, i.e., one set of hidden variables assigned in Step 1. Then we have the following property,

\[
l(w) = l(w, Z^*_\text{optimal}) \leq l(w, Z^*). \tag{3.13}
\]

Therefore, both Step 1 and 2 maintain or improve the auxiliary objective \( l(w, Z^*) \), which is an upper bound on the PAUC objective \( l(w) \). Focusing on the auxiliary objective, the algorithm can be considered as a coordinate descent approach (Felzenszwalb et al., 2010). In Step 1, fixing \( w \), we search over the exponentially large space for the optimal set of hidden variable assignments \( Z^* \); in Step 2, fixing \( Z^* \), we use the subgradient method to optimize over the model parameters \( w \). As we alternate between the two steps minimizing the upper bound on \( l(w) \), the objective \( l(w) \) itself is guaranteed to converge to a local optima. In both steps the state sequence \( z \) needs inference for each sample \( x \) and each label assignment \( \{+1, -1\} \); the Viterbi algorithm can be utilized because of the tree structure formed by the state variables.
3.4 Related Work

3.4.1 HMM-based classification

As mentioned, HMM is a generative model, which can be used for time-series classification (Kim and Pavlovic, 2011). Suppose all instances are generated from class-dependent HMMs defined on the same set of states with state-dependent parameters \( \{\mu_k\}_{k=1}^K \), and each class has a different set of HMM parameters including state-transition and initial-state probabilities, denoted as \( \pi^y \) and \( \xi^y \). Thus, given the model parameters, the joint likelihood of the sequence label \( y \), observations \( x \), and hidden states \( z \) can be written as,

\[
p(y, x, z; \Theta) = p(y)p(z_0; \xi^y) \prod_{t=1}^Tp(z_t|z_{t-1}; \pi^y)p(x_t|z_t; \mu_{z_t}). \tag{3.14}
\]

The parameters \( \Theta = \{\pi^c, \xi^c\}_{c=\pm1}, \{\mu_k\}_{k=1}^K \} \) can be learned via the Baum-Welch algorithm (Kim and Pavlovic, 2011), which is an EM-type algorithm, with a guarantee to converge towards a local optima.

After training the model, for new sequences, we can assign the label \( \hat{y} \) achieving the maximum likelihood,

\[
\hat{y} = \arg \max_y p(y, z; \Theta) = \arg \max_y \sum_z p(y, x, z; \Theta). \tag{3.15}
\]

The summation over all possible state paths \( z \) can be done via forward algorithms because of the tree structure in the model.

3.4.2 HCRF-based classification

For classification tasks, discriminative training \((p(y|x))\) is often preferred over generative models \((p(y, x))\). The reason is in generative models, the model parameters need to fit both \( x \) and \( y \). Meanwhile in discriminative models, only the dependency structure of \( y \) on \( x \) is modeled, and the relationship within \( x \) is ignored by the model. Therefore, given the same model complexity, the discriminative model usually has a
better classification performance than generative models, especially when the learned model is not in the true model class. In terms of time series classification, HCRFs are a discriminative approach using energy-based models compared with the HMMs which are probabilistic models (Quattoni et al., 2007).

As shown in Figure 3.1, HMMs possess a directed graphical structure while HCRFs have an undirected graphical structure. In HCRFs, each edge can be considered as associated with a potential indicating the compatibility between nodes values. The graphical structure in the right plot of Figure 3.1 corresponds to a potential function parametrized by model parameters $\Theta$ as $\psi(y, z, x; \Theta)$,

$$
\psi(y, z, x; \Theta) = w^T \phi(y, z, x)
$$

$$
= u^T \sum_{t=1}^{T} \phi_1(y, z_t, x_t) + v^T \sum_{t=1}^{T} \phi_2(y, z_t, z_{t-1}). \quad (3.16)
$$

In (3.16), The first term corresponds to the observation likelihood in HMMs and the second term corresponds to the state transition probabilities. One benefit of HCRF is its flexibility of feature design by adding edges and nodes not constrained by the probabilistic interpretation. For example, we can connect all observation nodes $x_t$ to a common node and add a constraint on this node.

Since the goal is to optimize $p(y|x)$, we need to construct the conditional probability from the potential function. One possible way is first constructing $p(y, z|x; \Theta)$
as,

\[ p(y, z|x, \Theta) = \frac{\exp \psi(y, z, x; \Theta)}{\sum_{y', z'} \exp \psi(y', z', x; \Theta)}. \]  

(3.17)

Then, integrating out hidden variables \( z \), the model gives the label assignment probability,

\[ p(y|x, \Theta) = \sum_z p(y, z|x, \Theta) = \frac{\sum_z \exp \psi(y, z, x; \Theta)}{\sum_{y', z'} \exp \psi(y', z', x; \Theta)}. \]  

(3.18)

When we train the model, we need to maximize the conditional probability \( p(y|x, \Theta) \) in (3.18). Even though this objective is not concave (its negative logarithm is not convex), in practice, the local optima found usually achieves reasonable results and better performance than models without hidden variables (Quattoni et al., 2007).

After model training, for a given new sequence, we can assign the label achieving the largest \( p(y|x, \Theta) \),

\[ \hat{y} = \arg \max_y p(y|x, \Theta). \]

3.5 Experiments

In this section, we evaluate the proposed method on synthetic datasets and compare it with several baselines.

Instances in each class are generated from a HMM-Gaussian model, with class-dependent state-transition probabilities. The number of sequences is \( N = 1000 \) with the sequence length \( T = 100 \), and the observation dimension \( P = 2 \). The labels are randomly assigned with 30 percent being positive. The class-dependent state transition matrices for two classes are \([0.8, 0.2; 0.2, 0.8]\) (negative label) and \([0.5, 0.5; 0.5, 0.5]\) (positive label). The state-dependent distributions are defined as Gaussian with the same covariance matrix but different mean parameters \([\pm 1]\) and \([\pm 1]\). Two settings are used for the covariance, \( i.e., \) a diagonal matrix with variance parameter 1 or 4 indicating low and high noise scenarios. One sample sequence is plotted in Figure 3.2. As can be seen, the sequence with positive label has a faster switching rate between
the positive and negative values, while the sequence with negative label alternates less frequently.

To demonstrate the performance of the proposed algorithm, we compare it with three approaches: two approaches not using temporal information, \textit{i.e.}, SVM and PAUC optimization (Kar et al., 2014) and another baseline based on the true model, \textit{i.e.}, the HMM-based classifier. In SVM and PAUC approaches, we first aggregate the observed sequences over the time domain and use the aggregated features (with normalization) as the input to the classifier. This is used to show the importance of temporal information; the aggregated features are plotted in Figure 3.3, where the examples from the positive and negative classes are not separable if losing the temporal dynamics. For all the approaches in the experiment, we use 50/50 training and testing split. As mentioned, we have both the high and low noise level scenarios. For each scenario, we further focus on low false alarm rate regions in \([0, \beta]\) with \(\beta \in \{0.05, 0.10, 0.15\}\). For each of the 6 settings, we compute the corresponding PAUC as the evaluation metric following (3.2). We repeat the experiments 10 times in each setting (the observed sequences are regenerated from the same model in each repeat) and report the mean and standard deviation.

As show in Table 3.1, the proposed method is comparable or slightly better than the HMM-based classifier on different settings. SVM and PAUC can not separate the sequences because of the loss of temporal information. The HMM-based classifier

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Left (L) and right (R), coordinate 1 (L) and coordinate 2 (R) varying with time for a single instance.}
\end{figure}
Figure 3.3: Scatter plot of the two coordinates aggregated over time for all the instances; samples from the positive and negative classes cannot be separated (colored blue and red).

is trained via EM with multiple initializations, with the best local optima selected. Because the HMM is the true model class, the learned HMM-based classifier can be considered as a strong baseline in terms of classification error minimization. The comparable or better performance obtained by the proposed method demonstrates its discriminative power. In most settings, the PAUC achieved by our method is improved over the HMM-based classifier because it adjusts the decision boundary to optimize the performance in low false alarm rate regions. Other features can be designed for different problems, not constrained by probabilistic interpretations.

One caveat is because the objective in the proposed method is not convex, the learned classifier might not be the optimal solution. This can be a problem in more complicated and noisy data scenarios. In our experiments, if we further increase the noise level, e.g., letting the additive noise have a variance of 9 (mean is ±1 for the two classes), the algorithm sometimes cannot find a good local optima and achieves worse performance than the HMM-based classifier. The model parameters need careful initialization in these scenarios. Exploring other optimization techniques is
Table 3.1: PAUC comparison in different scenarios, varying the false alarm region $[0, \beta]$ and the noise level $(\sigma/\mu)$

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$\sigma/\mu$</th>
<th>SVM</th>
<th>PAUC ± 0.0098</th>
<th>HMM ± 0.0147</th>
<th>PROPOSED ± 0.0292</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>1.0</td>
<td>0.0254 ± 0.0191</td>
<td>0.0331 ± 0.0098</td>
<td>0.9765 ± 0.0292</td>
<td>0.9797 ± 0.0273</td>
</tr>
<tr>
<td>0.10</td>
<td>1.0</td>
<td>0.0543 ± 0.0099</td>
<td>0.0619 ± 0.0274</td>
<td>0.9828 ± 0.0147</td>
<td>0.9938 ± 0.0055</td>
</tr>
<tr>
<td>0.15</td>
<td>1.0</td>
<td>0.0693 ± 0.0155</td>
<td>0.0679 ± 0.0121</td>
<td>0.9899 ± 0.0256</td>
<td>0.9957 ± 0.0054</td>
</tr>
<tr>
<td>0.05</td>
<td>2.0</td>
<td>0.0265 ± 0.0105</td>
<td>0.0322 ± 0.0130</td>
<td>0.8605 ± 0.0711</td>
<td>0.8433 ± 0.0705</td>
</tr>
<tr>
<td>0.10</td>
<td>2.0</td>
<td>0.0519 ± 0.0175</td>
<td>0.0488 ± 0.0096</td>
<td>0.8722 ± 0.0986</td>
<td>0.8941 ± 0.0488</td>
</tr>
<tr>
<td>0.15</td>
<td>2.0</td>
<td>0.0721 ± 0.0207</td>
<td>0.0724 ± 0.0229</td>
<td>0.9304 ± 0.0713</td>
<td>0.9341 ± 0.2560</td>
</tr>
</tbody>
</table>

left as a future direction to explore.

3.6 Conclusions

In this work, a partial AUC optimization approach is proposed for time series classification problems, where the temporal information is incorporated by introducing hidden variables. The connections are built with other classification approaches including the HMM-based classifier and HCRFs. Simulation results demonstrate the effectiveness of optimizing the performance in the low false alarm rate regions.

A couple of interesting directions are left as future work. First, developing online or distributed algorithms for long sequences and large dataset remains challenging because of the non-decomposable loss function designed in the objective. Second, capturing high-order temporal information is useful in many scenarios though the parameter learning can be a challenge.
Modeling correlated arrival events with latent semi-Markov processes

4.1 Introduction

The study and development of complex dynamical systems has led to the availability of increasingly large datasets, recording events evolving asynchronously and at multiple time-scales. Modeling such data by discretizing time at the resolution of the fastest events is inefficient and inelegant, especially when the relevant time-scales themselves must be inferred. It is much more natural to work directly in continuous-time, and there has been a growth of such applications in the statistics and machine learning communities (e.g., Nodelman et al. (2002); Scott and Smyth (2003); Golightly and Wilkinson (2011); Saeedi and Bouchard-Côté (2011); Teh et al. (2011)). Nevertheless, the scope of this prior research is limited and much work remains to apply continuous-time models to high-dimensional problems with interesting structure in the latent states and dynamics of the latent process.

The specific application motivating the methodology of this chapter is the analysis of biometric galvanic skin response (GSR) data. User arousal (or excitation) events
translate to the generation of specific observed GSR waveforms (Silveira et al., 2013). Here, we consider GSR observations taken from a number of users exposed simultaneously to a common video stimulus. The ability to accurately characterize the latent stimulus events generating the observed biometric excitation reactions has applications in the areas of recommendation, market research, advertising, etc. More generally, our ideas are also applicable to point process data from other domains, like neuroscience (e.g., Yu et al. (2009)), biometrics (e.g., Barbieri et al. (2005)), network data analysis (e.g., Ryu and Lowen (1996)), and ecology (e.g., Ogata and Tanemura (1985)).

In all these applications, one observes point process data exhibiting significant variability, or inhomogeneity, over time. In the context of GSR observations from movie stimulus, an intense action scene may have many GSR arousal reactions (i.e., arrivals) in a short time, while a quiet dialog scene may elicit very few arrivals over a potentially long interval. In other applications, the rate of webpage requests can vary with external events, while the spiking of a neuron can vary with stimulus. Without explicit knowledge of the external events, we look to extract the latent structure from the observed point process data.

For a single observed point process, an appropriate modeling framework is that of Markov-modulated Poisson processes (MMPP) (e.g., Scott and Smyth (2003)). This is a doubly-stochastic Poisson process, where the unknown rate is modeled as a realization of a continuous-time Markov jump process (MJP). In our case, we observe a number of correlated inhomogeneous Poisson processes, which we couple via a common low-dimensional underlying process. Specifically, we assume that the arrival rates are described by a small number of binary switching signals, indicating the presence or absence of a relevant external event. Examples for biometric point processes from movie stimulus could include action scenes, dramatic dialog, comedic events, or a main character appearing. We model this as a collection of continuous-
time binary signals linked by a factor-loading matrix to the intensities of the observed processes. Rather than risking under-fitting with a small number of factors, we allow the model to choose from a large number of relevant sources by placing a shrinkage prior on a source loading matrix. Furthermore, rather than modeling the binary sources as memoryless (as in MMPPs), we allow more flexibility in the dynamics general hazard functions. This mirrors recent work in the discrete-time literature (e.g., Fox et al. 2011; Johnson and Willsky 2013) to model state persistence and more flexible distributions over state durations.

We evaluate the performance of our model and inference methodology on both synthetic and real-world biometrics data. For the latter, we apply our methodology to a biometric galvanic skin response dataset taken from users viewing a full feature-length film. We find that the resolved latent structure correlates with explicit feedback (i.e., user ratings) in terms of both excitement during specific scenes in the film and the users’ overall impression of the film.

4.2 Problem and Model formulation

We wish to model a collection of \( U \) sequences of events over an interval \([0, T]\). For “user” \( u \in \{1, \cdots, U\} \), we denote the set of arrival times as \( \{y_{u,j}\} \), with \( y_{u,j} \) being the time stamp of the \( j \)-th event in stream \( u \). Each sequence \( y_{u,\cdot} \) is an inhomogeneous Poisson process with instantaneous rate \( \gamma_u(t) \). The latter is expressed as a user-specific base Poisson rate \( \lambda_u \) modulated by a stimulus-determined function over time. The rates of these \( U \) Poisson processes share information through \( K \) binary latent sources \( s_k(t) \) with \( k \in \{1, \cdots, K\} \). Below, we use \( s.(t) \) to denote at time \( t \), the column vector consisting of all \( K \) sources \( s_k(t) \). Calling the loading matrix \( W \) (with \( \mathbf{w}_u \in \mathbb{R}^K \), a row vector specific to user \( u \)), we have
\[ y_{u}, \sim \mathcal{PP}(\gamma_u(\cdot)), u = 1, \cdots, U \] \hspace{1cm} (4.1)

\[ \gamma_u(t) = \lambda_u \exp(w_u s(t)), t \in [0, T] \] \hspace{1cm} (4.2)

Here, \( \lambda_u \) represents the baseline arrival rate, while the elements of \( w_u \) indicate how relevant each of the \( K \) sources is to user \( u \). Our goal is to estimate both these user specific parameters as well as the latent sources \( s_k(t) \) from the set of arrival times \( \{y_{u,j}\} \).

### 4.2.1 Binary semi-Markov Jump processes

In our application, we model each of the \( K \) latent sources as a binary signal, switching on and off depending on whether the associated external characteristic is active. While it is simplest to model the latent functions, \( s_k(t) \), as Markov jump processes (MJPs), the resulting memoryless property can be inappropriate, allowing unnecessary switching between states. Thus, we model these as binary semi-Markov Jump Processes (bsMJP) (Feller, 1964). Realizations of a bsMJP are right-continuous, piecewise constant binary functions where, unlike an MJP, the rate of state transitions vary with the time since the last transition. This is formalized by a hazard function \( h^{(0)}(\nu) \), giving the rate of transition from state 0-to-1, \( \nu \) time units after entering state 0 (\( h^{(1)}(\nu) \) is similarly defined, and we do not allow for self-transitions). For an MJP, \( h^{(s)}(\nu) \) is a constant, resulting in a memoryless property; in contrast, by making \( h^{(s)}(\nu) \) take small values for small values of \( \nu \), we can discourage the system from leaving a state it has just entered. In our applications, we assume \( h \) belongs to the Weibull family, with

\[
h^{(s)}_k(\nu) = \frac{\beta_k^{(s)}}{\mu_k^{(s)}} \left( \frac{\nu}{\mu_k^{(s)}} \right)^{\beta_k^{(s)} - 1}
\] \hspace{1cm} (4.3)
This corresponds to the interval length between two state transitions following a Weibull distribution

\[ Pr(\nu|\beta_k^{(s)}, \mu_k^{(s)}) = \exp\left(-\left(\frac{\nu}{\mu_k^{(s)}}\right)^{\beta_k^{(s)}} \frac{\beta_k^{(s)}}{\mu_k^{(s)}} \left(\frac{\nu}{\mu_k^{(s)}}\right)^{\beta_k^{(s)}-1}\right) \]

\( \beta_k^{(s)} \) and \( \mu_k^{(s)} \) are the shape and scale parameters of the Weibull distribution for state \( s \). Setting \( \beta_k^{(s)} \) to 1 recovers the exponential distribution (and our switching process reduces to an MJP).

The hazard functions govern the dynamics of the bsMJP; we also need an initial distribution \( \pi_k \) over states. Then, we have

\[ s_k(\cdot) \sim \text{bsMJP}(\pi_k, h_k^{(1)}(\cdot), h_k^{(0)}(\cdot)) \quad (4.4) \]

### 4.2.2 Number of Latent Sources

In most applications, the number of latent sources is unknown, and must be inferred. It is desirable to place a prior on this quantity, and try to infer it from the data. A more flexible approach is a nonparametric solution: allow for an infinite number of potential sources, with each user influenced by a finite, and unknown, subset of them. This corresponds to a binary matrix with infinite columns, and with element \((u,k)\) indicating whether or not the bsMJP \( k \) is relevant to user \( u \). A popular prior on the resulting binary association matrix is the Indian Buffet Process (IBP) (Griffiths and Ghahramani, 2006), however, despite its elegance, inference of the IBP is complicated and the resulting Markov chain Monte Carlo algorithm mixes poorly as it explores the combinatorial space of binary matrices. This raises a need for alternate approaches to flexibly model the unobserved latent structure.

Here, inspired by Bayesian shrinkage estimation for sparse regression problems (Polson and Scott, 2012), we control complexity of the factor loading matrix using a multiplicative gamma process (MGP) shrinkage prior (Bhattacharya and Dunson,
2011). Rather than placing a spike-and-slab prior on the columns of $\mathbf{W}$, we model each row vector of the factor loading matrix ($\mathbf{w}_u \in \mathbb{R}^K$) as a collection of Gaussian variables increasingly concentrated around the origin. Specifically,

$$
\mathbf{w}_u \sim \mathcal{N}(0, \Lambda^{-1}), \quad \Lambda = \text{diag}(\tau_1, \tau_2, \cdots)
$$

$$
\tau_k = \prod_{l=1}^k \xi_l, \quad \xi_l \sim \text{Ga}(\alpha, 1)
$$

By choosing $\alpha > 1$, the $\{\xi_l\}$ are on average larger than 1, encouraging $\tau_k$ to increase with $k$. This in turn causes the amplitudes of the $\mathbf{w}_{uk}$ to shrink close to (while not exactly equaling) 0. The construction results in stochastically ordering the latent sources, with early components having large variance and corresponding to sources relevant to most users. As $k$ increases, these weights are shrunk more and more strongly towards zero, while still allowing a few to escape this pull (allowing us to model sources specific to certain users). Thus, we potentially obtain an infinite number of latent sources $\{s_k(t)\}$, stochastically ranked by their contribution to the rate function $\{\gamma_u(t)\}$. In practice, we truncate at a sufficiently large $K$.

### 4.2.3 Miscellaneous variables

Finally, we describe prior distributions over the remaining variables. We model the base Poisson rate $\lambda_u$ of each user as independently drawn from $\text{Ga}(c, d)$. We construct conjugate hyperpriors for $\beta_k^{(s)}$ and $\mu_k^{(s)}$ after a variable transformation outlined in Section 4.4.

### 4.3 Related Work

Early work analyzing arrival data did not exploit correlation across multiple arrival streams (Daley and Vere-Jones, 1998; Kass and Ventura, 2001; Riehle et al., 1997). The work in Smith and Brown (2003) introduces a single first-order autoregressive latent state model with a proposed EM-algorithm to estimate parameter values from
multi-user arrival data. A more complex hidden Markov Model (HMM) approach using multiple latent states is presented in Escola et al. (2011). In contrast to the work in this chapter, the time evolving stimulus has a simple structure and is explicitly known.

Perhaps most similar to the work in this chapter is the Gaussian process factor analysis (GPFA) approach of Yu et al. (2009). They modeled the intensities as a series of correlated neural spike trains by linearly transforming a small set of latent Gaussian processes. By contrast, the binary switching signals in our model captures specific aspects of the latent structure, and our shrinkage approach allows us to infer the number of sources. Finally, the simpler structure of the bsMJP means our model is scalable to longer observation intervals. Inference in Yu et al. (2009) required a variational approximation to the complicated posterior, while no such approximations are needed here.

Our modeling also relates to previous work on finite state systems in discrete and continuous time. We generalize the Markov-modulated Poisson process (MMPP) (Scott and Smyth, 2003) by allowing correlated Poisson intensities, and by allowing more flexible (i.e., non-exponential) holding times. This latter point of semi-Markovianity has been a topic of recent interest in the discrete time-series modeling community (Fox et al., 2011; Johnson and Willsky, 2013), although it fits more naturally in the continuous-time setting. While we took a shrinkage approach to coupling the latent sources, our ideas easily extend to genuine, truncation-free nonparametric models based on the IBP. Such ideas are related to the infinite factorial HMM (Van Gael et al., 2009), a discrete-time Markov model with infinite, independent latent sources, and provide alternatives to the Dirichlet process-based infinite state MJP models developed in the continuous-time literature (Teh et al., 2011; Saeedi and Bouchard-Côté, 2011).
4.4 MCMC for bsMJs

A challenge for more widespread application of continuous-time models is the problem of posterior inference. For our model, the central problem reduces to sampling from the posterior over bsMJP paths given the Poisson arrival times, \( \{y_{u,j}\} \). We are aware of two general approaches to MCMC sampling for such models: the particle MCMC approach of Andrieu et al. (2010); Golightly and Wilkinson (2011) and a thinning-based approach of Rao and Teh (2012, 2013). Here, we adapt the latter to our problem.

Observe that a sample path \( s_k \) of the bsMJP is entirely determined by the set of transition times \( \phi_k = \{\phi_{k,1}, \cdots, \phi_{k,n_k}\} \), and the states evaluated at these times \( \{s_k(t), t \in \phi_k\} \) (for the binary sMJP, the latter are redundant given the initial state). Also, recall that \( \nu \) time-units after entering state 0, the rate of transitioning from state 0 to 1 is given by \( h_{k}^{(0)}(\nu) \) (respectively, \( h_{k}^{(1)}(\nu) \) for state 1). Typically self-transitions are not allowed, and have rate equal to zero.

We now define an equivalent continuous-time system but with self-transitions, occurring with constant rates \( \Omega_{k}^{(0)} \) and \( \Omega_{k}^{(1)} \) for states 0 and 1. These self-transitions will serve as auxiliary variables facilitating the easy resampling of new state values. Our approach simplifies Rao and Teh (2012, 2013), where a multiplicative bound on the transition rates resulted in self-transition rates depending on the current holding time.

For state \( s \), candidate transition times (whether self-transitions or not) are now drawn from a hazard function \( H_{k}^{(s)}(\nu) = h_{k}^{(s)}(\nu) + \Omega_{k}^{(s)} \). We sample these candidate events sequentially. Thus, assume \( l \) time units have passed since the last state transition (when we entered state \( s \)). Then, we sample the next candidate event-time from \( H_{k}^{(s)}(\cdot) \), conditioning on it being larger than \( l \). Assuming this is \( l + \Delta \), we advance time by \( \Delta \), and assign an actual transition out of state \( s \) with probability \( \frac{h_{k}^{(s)}(l+\Delta)}{H_{k}^{(s)}(l+\Delta)} \).
(otherwise this event is treated as a self-transition). After updating $l$, we repeat this process until the current time exceeds $T$. Algorithm 1 in the appendix gives details of this generative process. It is easy to show that discarding the self-state transitions corresponds to a draw from the original bsMJP.

We use the construction above to define a Markov operator over bsMJP paths, with the desired posterior as its stationary distribution. Denote the set of self-transitions as $\tilde{\phi}_k$, with the set of actual transitions given by $\phi_k$. From our construction of the previous paragraph, it follows that given the current bsMJP trajectory $s_k$, the set $\tilde{\phi}_k$ is conditionally an inhomogeneous Poisson with piecewise-constant rate $\Omega_{k}^{(s_k(t))}(t)$. Thus, when the system is in state $s$, the Poisson rate is $\Omega_{k}^{(s)}$, and we can easily reconstruct $\tilde{\phi}_k$.

Denote all the candidate times as $\Phi_k = \phi_k \cup \tilde{\phi}_k$. Having introduced $\tilde{\phi}_k$, we sample a new path, now restricting ourselves to paths that change state at some subset of $\Phi_k$ (rather than searching over all paths with all possible transition times). Consequently, sampling a new path conditioned on $\Phi_k$ amounts to reassigning labels “transition” or “self-transition” to each of the elements of $\Phi_k$. Our problem now reduces to sampling a trajectory of discrete-time model, and can be carried out efficiently using dynamic programming algorithms such as forward filtering backward sampling (FFBS) (Frühwirth-Schnatter, 1994). Note that this step accounts for the Poisson observations. In particular, given two successive candidate transition times, $t_1$ and $t_2$ in $\Phi_k$, the likelihood the system remains in state $s$ over this interval equals the probability of the subset of Poisson observations $\{y_{u,j}\}$ falling in $[t_1, t_2)$ under state $s$. This follows a Poisson distribution and is easy to calculate.

Overall, sampling proceeds by alternately sampling a set of thinned events $\tilde{\phi}_k$ given the current trajectory, and then a new trajectory given $\Phi_k = \phi_k \cup \tilde{\phi}_k$. We leave the details of this for the appendix.
4.5 Model inference

We now describe the overall Markov Chain Monte Carlo (MCMC) algorithm. We wish to infer the posterior distribution over \{\{w_u\}, \{s_k(\cdot)\}, \{\lambda_u\}, \{\xi_k\}, \{\beta_k^{(s)}\}, \{\mu_k^{(s)}\}\}. Our algorithm is a Gibbs sampler, sampling each variable conditioned on the remaining variables. For ease of notation, we use \(p(\cdot | \sim)\) to denote the posterior distribution of one variable conditioning on all other variables.

**Inference of Latent Sources** \((s_k(\cdot))\): We cycle through all \(K\) sources, resampling each bsMJP path \(s_k\) conditioned on all other variables. To do so, we use the algorithm of the previous section: for each source, conditionally introduce the Poisson distributed thinned events \(\tilde{\phi}_k\), and then conditionally resample the new trajectory using the FFBS algorithm.

**Inference of Factor Loadings** \((w_u)\): The Gaussian prior on \(w_u\) and the non-linear likelihood in Equation (4.2) result in a nonstandard posterior distribution. While a simple Metropolis-Hastings (MH) approach is to perturb the current value \(w_u\), proposing a new value from a normal variable drawn with mean centered at \(w_u\), we instead use the following proposal distribution tailored to our problem.

Noting that \(p(w_u|\sim)\) is log-concave, we use Newton’s method to find the MAP \(\hat{w}_u\) with gradient and Hessian

\[
g(w_u) = \sum_{j=1}^{N_u} s(y_{u,j}) - \int_0^T \exp(w_u s(t)) \lambda_u s(t) dt - \Lambda w_u^T
\]

\[
H(w_u) = -\int_0^T \exp(w_u s(t)) \lambda_u s(t) s(t)^T dt - \Lambda
\]

respectively. Here, \(N_u\) is the number of arrivals observed for user \(u\). Because latent sources, \(s_k(t)\), are simple binary functions, the integrals above reduce to finite summations and are easily computable. Thus, at each iteration of the sampler, we
propose \( w_u^* \sim \mathcal{N}(\hat{w}_u, qH^{-1}(\hat{w}_u)) \), where \( q \) is a tuning parameter. The new proposal \( w_u^* \) is accepted with probability

\[
\min\{1, \frac{Pr(w_u^* | \sim)\mathcal{N}(w_u^{(old)}, \hat{w}_u, qH^{-1}(\hat{w}_u))}{Pr(w_u^{(old)} | \sim)\mathcal{N}(w_u^*, \hat{w}_u, qH^{-1}(\hat{w}_u))}\}
\]

We set \( q \) to 5 in our experiments, and the high acceptance rates suggest this is an efficient approach.

**Hyperparameter inference** \((\beta_k^{(s)}, \mu_k^{(s)}, \xi_k)\):

By applying the following transformation: \( \beta' = \beta, \mu' = \mu^{-\beta} \), we obtain a new representation of the Weibull distribution with p.d.f. and c.d.f.:

\[
f(\nu; \beta', \mu') = \beta'\mu'\nu^{\beta'-1}\exp(-\mu'\nu^\beta) \quad (4.6)
\]
\[
F(\nu; \beta', \mu') = 1 - \exp(-\mu'\nu^\beta) \quad (4.7)
\]

Using these representations, a gamma prior on \( \mu' \) is the conjugate, and with hyperparameters \((e, f)\), we can directly sample \( \mu' \) from the posterior,

\[
p(\mu_k^{(s)} | \sim) \propto \text{Ga}(\mu_k^{(s)}; e + \sum_i \mathbb{I}(\phi_{k,i} = s), f + \sum_{i:s_k(\phi_{k,i}) = s} \Delta_{k,i+1}^{(s)})
\]

Here \( \Delta_{k,i+1} = \phi_{k,i+1} - \phi_{k,i} \).

The components of the MGP shrinkage prior, \( \{\xi_k\} \), can be directly sampled from the conditional posterior:

\[
p(\xi_k | \sim) \propto \text{Ga}(\alpha + \sum_{u=1}^{U} \frac{U(K - k + 1)}{2}, 1 + \frac{1}{2} \sum_{u=1}^{U} \sum_{l=k}^{K} \sum_{m=1, m \neq k}^{K} w_i^2 \prod_{m=1, m \neq k}^{K} \xi_m)
\]

We place a uniform prior for \( \beta_k^{(s)} \) on a set of discretized grid points and sample from the candidates according to their posterior mass. Other variables, including \( \{\lambda_u\} \), are straightforward to sample.
4.6 Experiments

We consider two synthetic datasets of arrival data to study our model and sampler, before considering a real-world biometric application. For the latter, we assess the model’s ability to capture and explain observed explicit user feedback (i.e., user ratings). As a baseline, we used the kernel methods analyzed in Zhang and Kou (2010) to estimate the inhomogeneous Poisson rates, and thus the underlying model parameters (e.g., instantaneous rate, factor loadings, etc.). We found that with a carefully chosen kernel width, the kernel shape does not matter much, and for simplicity, we used a uniform kernel (giving a time-discretized binning method). An oracle choice of the best bin size was used after testing multiple sizes.

4.6.1 Synthetic Experiments

For our first dataset, we generated $K = 3$ bsMJP paths over an interval of length $T$ with initial state distribution $\pi_k = [0.5, 0.5]^T$ and Weibull hazard rates. Weibull shape $\beta_k^{(s)}$ and scale parameters $\mu_k^{(s)}$ were uniformly drawn from $[1, 5]$ and $[50, 100]$, while each row of the loading matrix $W$ was independently generated from a standard normal, $\mathcal{N}(0_K, I_K)$, where $0_K$ is a $K$-dimensional vector of zeros, and $I_K$ is the $K \times K$ identity matrix. The columns of $W$ were reordered, so that the energy contained in $w_k$ decreased monotonically. Our observations consisted of $U = 12$ sequences of event arrivals, each with a base Poisson rate $\lambda_u$ drawn from a truncated Gaussian distribution centered at $\lambda_0$ with small variance ($\ll \lambda_0$). As explained below, we varied both the average base rate, $\lambda_0$, and the observation time length, $T$. For inference, the fixed hyperparameters of the sampler were set as: $\alpha = 3, c = d = e = f = 10^{-3}$, and $\pi_k = [0.5, 0.5]^T$. We ran 5000 MCMC iterations of our MCMC sampler, discarding the first 2000 as burn-in, with posterior samples collected every 5 iterations. The running time of a typical trial (with $T = 1000$ and about 120 event arrivals for each
Table 4.1: Effective sample size (ESS) and autocorrelation function (ACF) of the base rate ($\lambda_u$), number of transitions of source $s_k(t)$ ($n_k$), and the factor loadings ($w_{uk}$)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>$\lambda_u$</th>
<th>$n_k$</th>
<th>$w_{uk}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESS/iteration</td>
<td>0.046</td>
<td>0.278</td>
<td>0.161</td>
</tr>
<tr>
<td>ESS/second</td>
<td>0.076</td>
<td>0.467</td>
<td>0.266</td>
</tr>
<tr>
<td>ACF (LAG 5)</td>
<td>0.609</td>
<td>0.101</td>
<td>0.225</td>
</tr>
<tr>
<td>ACF (LAG 10)</td>
<td>0.415</td>
<td>0.049</td>
<td>0.100</td>
</tr>
<tr>
<td>ACF (LAG 50)</td>
<td>-0.047</td>
<td>-0.016</td>
<td>-0.039</td>
</tr>
</tbody>
</table>

user) was about 3000 seconds with unoptimized Matlab code on a computer with 2.2GHz CPU and 8GB RAM. To evaluate the mixing behavior of the sampler, we use R-coda Plummer et al. (2006) to compute the effective sample size (ESS), as well as Markov chain autocorrelation functions (ACF) of various model parameters. Table 4.1 shows these statistics, with the ACF shown for the parameters $\lambda_1$, $w_{11}$ and $n_1$, the number of transitions of the first latent source. These numbers are typical of MCMC samplers, and show the sampler mixes well

**Instantaneous Rate Estimation:** One of the main advantages to the proposed model is the ability to exploit correlations across streams of observed arrival data. This is especially important when the base arrival rate of each user is quite low. In this experiment, we examine the ability to accurately recover $\{\gamma_u(t)\}$, the instantaneous arrival rate of each user, choosing the mean of base rates $\lambda_0$ from values in $\{0.01, 0.02, 0.05, 0.10, 0.20\}$. We keep the observation time length constant at $T = 2000$.

We measured the estimation error of the instantaneous rate by discretizing the interval $[0, T]$ using $N = 1000$ evenly spaced grid points. We compute the posterior mean estimation error at each grid point, normalizing with respect to that point’s true rate, and record the average normalized error for 15 repeats. Rate inference is

---

1 Code available at [http://people.duke.edu/~wl89/](http://people.duke.edu/~wl89/)
performed using both the proposed model, and the binning approach of Zhang and Kou (2010) using bandwidth values of 1, 3, 10 times the inverse of mean arrival rate. The results are shown in Figure 4.1.

For very low base arrival rates (i.e., $\lambda_0 = 0.01$) all methods perform similarly. As $\lambda_0$ increases, our model performs significantly better than the competing binning method. For a mean base rate of $\lambda_0 = 0.05$, we find that the error rate is less than half the error rate of the best choice of binning bandwidth.

**Factor Loading Matrix Inference:** As Figure 4.1 suggests, at low arrival rates there is not enough information to recover the instantaneous Poisson rates (and thus the state of the latent sources). This is shown in the left plot of Figure 4.2: for base rates $\lambda_0 = \{0.01, 0.02, 0.10\}$, as $T$ increases (taking values in $T_{\text{cand}} = \{200, 500, 1000, 2000, 5000\}$), the error in the latent sources (estimated over a grid as in the previous section) increases slightly: this is because the ‘number of variables’ in a bsMJP path increases with $T$. 

![Figure 4.1](image-url)
In such situations, it is still of interest of estimate parameters like the factor loading matrix $W$: even if we cannot reconstruct exactly what a user had in mind at any previous time, we would still like to characterize their behaviour to make predictions in the future. The right plot shows that the estimation error of $W$ decreases monotonically as the observation interval increases, implying that these parameters can be recovered even if the posterior distribution over latent sources never concentrates. Here, for each posterior sample, $\hat{W}$, the estimation error with respect to the true factor loadings, $W$, is computed as $\frac{||\hat{W} - W||_2}{||W||_2}$.

**Deviation from Markovianity:** Setting the Weibull shape parameter $\beta$ to 1 recovers the exponential distribution, reducing the latent sources to memoryless binary MJPs (bMJPs). To show the flexibility afforded by this parameter, we consider latent sources that are square waves, switching between ‘1’ and ‘0’ at a fixed frequency. Figure 4.3 compares inferences over a latent source using a bMJP and a bsMJP prior. We see that the state intervals inferred by bMJP are more irregular, showing unnecessary switching between states. For the bsMJP, we placed a uniform prior on $[1, 5]$ on the shape parameter, allowing the model to estimate the state persistence. Figure 4.3 shows the posterior over this parameter places significant mass
Figure 4.3: Left: inferred sources using bsMJP and bMJP models. For each subplot, the first row shows the truth, the second the inferred source using bsMJP, and the third, that using bMJP. Right: posterior distribution of a shape parameter inferred using bsMJP away from 1, forcing more regular state holding times.

Latent Factor Number Estimation: In real world applications, the number of latent sources is usually unknown a priori. Our MGP shrinkage prior from Section 4.2 allows us to infer the number of dominant sources. Again, we vary the observation length from $T_{\text{cand}} = \{500, 2000, 5000\}$, set the base arrival rate $\lambda_0 = 0.10$, and set the true number of latent sources as $K = 3$. When doing inference, we truncate the number of sources, picking a large enough number $K_+ = 10$ to avoid under-fitting. Though the sampler is not sensitive to $\alpha$, a setting $\alpha \to 1$ leads to a higher chance of sampling precision sequences which are not monotonically increasing. As mentioned before, $\alpha$ is set as 3 in our experiments. For each posterior sample, we identify the relevant sources by thresholding the associated weight-vector $w_k$, picking the smallest collection of weights containing 90 percent of total energy of $W$. Figure 4.4 demonstrates the behavior of the posterior distribution with respect to the inferred number of latent sources. We find that as an increasing number of observations are available, the posterior mass quickly concentrates around the true value $K = 3$. 

80
Figure 4.4: Posterior distribution of number of latent sources

4.6.2 Skin Conductance Biometrics

Finally, we apply our model to a dataset from a real-world biometrics application. This dataset was collected with 10 volunteers (users) watching a feature-length film while wearing Galvanic Skin Response (GSR) sensors (the Affectiva Q-Sensor, Affectiva (2012)) measuring skin conductance. Using a current state-of-the-art decomposition approach from Silveira et al. (2013), we extract user arousal responses from the observed skin conductance signals to obtain arrival data for each user across the two-hour, eighteen minute film. As shown in Figure 4.5, a typical user has about 150 event arrivals during the recording, similar to the $\lambda_0 = 0.02$ scenario in the synthetic experiments.

Below, we analyze users’ arousal level based on their GSR signals. We also analyze the similarity between users and explore dominant latent events in the movie using our model. We validate our analyses using two types of explicit feedback obtained from this experiment. First, all the users were asked to recall 10 scenes from the film and rate their arousal intensity from 1 to 10 for each scene (“1” being “tedious” and “10” representing “exciting/tense”). Second, the general rating of the movie from
1 to 5 was also collected, indicating the user’s overall opinion of the film (\{“Poor”, “Fair”, “Good”, “Very Good”, “Excellent”\}).

We estimate the rate function, factor loading matrix, and latent sources using MCMC posterior samples. As discussed in Section 4.4, a non-informative prior $\text{Ga}(10^{-3}, 10^{-3})$ is put on $\lambda_u$ and $\mu^{(s)}_k$, and uniform prior on $[1, 5]$ is put on $\beta^{(s)}_k$. The maximum number of latent sources is set as $K_+ = 10$. Hyperprior $\alpha = 3$, results in factor loading matrices $W$ where on average about $K = 3$ columns contain 95% of total energy. Producing 10000 samples took about 12000 seconds.

**Instantaneous Rate Inference:** To qualitatively show the benefit of sharing information across users, we ran our model using both multiple traces ($U = 10$) and a single trace ($U = 1$) (both with a maximum of $K = 10$ latent sources). The middle and bottom plots of Figure 4.5 show posterior mean of the estimated rate function

---

**Figure 4.5:** Extracted arousal events from one user. Top: event arrivals; middle: jointly estimated rate. Bottom: rate inferred from single user’s trace.
(along with an uncertainty interval equal to one posterior standard deviation) for the shared and separate cases. We see that using multiple trace information gives estimates of the rate function that are piecewise stable, unlike the single trace case where the estimated rate function tends to follow the empirical Poisson rate. The inferred hazard function tends to be more parsimonious in terms of state-changes as well, as it is contrained to explain more data. Such information is useful to tease apart responses to common stimulus from user-specific responses, and we look at this in more detail at the end of this section.

To quantitatively measure our rate estimation, we correlated our estimated rate values with the explicit arousal feedback for the 10 scenes. We transformed the recalled arousal intensity (1-10) into binary labels by comparing each to the average arousal intensity of each user (such that each user has 5 scenes with a “0” class label and five scenes with a “1” class label). Using these binary labels as ground truth, we compare the instantaneous rate from the posterior mode estimate of our proposed method against that derived from the binning approach and a Markovian version of our methodology using bMJP. For all approaches, we evaluate the inferred rate function at time points corresponding to the 10 recalled scenes for all users and then inferred the binary labels at those time points by thresholding. Varying the threshold, we plot the ROC curves in the left plot of Figure 4.6.

As shown in the figure, the instantaneous rate inferred by the proposed model conforms to the user explicit feedback better than the rate estimated via the binning approach and the simplified model with bMJP. Specifically, our proposed algorithm is able to correctly classify almost 40% of the user explicit scene ratings with no false alarms, while the binning approach only classifies 10% of the scenes with no false alarms.

**Factor Loading Matrix Inference:** Each user has their own factor loading vector which can be used to calculate the distance between pairs of users. Thus,
we compute the pairwise Euclidean distance between users using the posterior mode estimate of the factor loading matrix. We then test how well this user-similarity metric predicts the user ratings. Using all 45 possible pairs of users, we plot two sets of ROC curves: in the first, we compare pairs of users with the same rating (e.g., both users rate the film at “4”) versus users that differ by a single rating point (e.g., “3” and “4”); in the second, we compare pairs of users with the same rating versus users that differ by two ratings points (e.g., “3” and “5”). As illustrated in the right plot of Figure 4.6, the proposed method does well with predicting user rating similarity, with the ability to classify over 55% of the users with the same rating from the set of users two ratings apart, with no false alarms.

![ROC curves](image)

**Figure 4.6:** ROC curves predicting (left) user arousal intensity, (right) user rating similarity from loading vectors \( \mathbf{w}_u \)

**Latent Source Inference:** Finally, we analyze the dominant latent sources underlying the observed arousal responses returned by our inferences. The left plot of Figure 4.7 shows the posterior distribution over the possible number of dominant sources, defined as the minimum number of columns of the factor loading matrix containing 90 percent of total energy. The posterior mass concentrates around 4 and 5, and we plot the 5 dominant latent sources in the right plot of Figure 4.7. The first source is an offset of the baseline Poisson rate. We found the following 4 sources had fairly clear interpretations. For the second source, the elements in the corresponding
column of the factor loading matrix are all positive, indicating this factor enhances arousal intensity. Specifically, this is activated at around 20 minutes for scenes about a plane crash, around 55 minutes and 65 minutes for key turning points of the plot, and around 115 minutes and 125 minutes for a climax and a surprising denouement respectively. Taking the third source as another example, both positive and negative factor loadings exist among all users, indicating this factor enhances arousal intensity for some of users but suppresses it for others. This is activated for the scene when the main actor first meets the main actress, and for the occurrence of their last dialogue. Such information can be used along with user information to better understand users, the stimulus, and the interaction between the two.

4.7 Discussion

There are a number of variations and extensions to our modeling choices worth exploring. While we placed a multiplicative gamma shrinkage prior on the factor loading matrix, an alternative is to construct similar priors using the framework of Lévy processes Applebaum (2004); Polson and Scott (2012). Similarly, we can allow added structure to the loading matrix $W$ (e.g., clustering its rows, to infer clusters of users), or allow $W$ to vary with time (modeling the evolution of a users
tastes). Another important extension incorporates user-covariates like age, sex, or profession. Ultimately, the goal is not just to understand users and stimuli, but to use models like ours to adaptively modify the stimulus by monitoring user response. This is central to applications ranging from human prosthetics and brain-computer interface to recommendation and targeted advertising.
4.8 Appendix

4.8.1 Generative process for bsMJP

**Algorithm 3** Generative process for a K-dimensional bsMJP path in $[0,T]$

**Require:** Hazard function of each state and each latent feature $h_{0k}(\cdot), h_{1k}(\cdot), k = 1, \ldots, K$, constant hazard rates $\Omega_{0k}, \Omega_{1k}$, and initial state distribution $\pi_0$.

**Ensure:** A K-dimensional sMJP path $\{\phi_k, s_k(\phi_k)\}$

1. while $k \in \{1, 2, \ldots, K\}$ do
2. Initialize $l_0 = 0, i = 0, \phi_k,0 = 0, \phi_k = \{\phi_k,0\}, s_k(\phi_k,0) \sim \pi_0$,
3. while $\tilde{\phi}_{k,i} < T$ do
4. increment $i$
5. Sample $\Delta_i \sim H_{\tilde{s}_k(\phi_k,i-1),k}(\cdot)$. Set $\tilde{\phi}_{k,i} = \phi_{k,i-1} + \Delta_i$.
6. Draw $\delta \sim \text{Unif}(0,1)$
7. if $\delta < \frac{h_{\tilde{s}_k(\phi_k,i-1),k}(l_{i-1}+\Delta_i)}{h_{\tilde{s}_k(\phi_k,i-1),k}(l_{i-1}+\Delta_i)+\Omega_{\tilde{s}_k(\phi_k,i-1),k}}$ then
8. Set $l_i = 0, s_k(\tilde{\phi}_{k,i}) = 1 - s_k(\phi_{k,i-1}), \phi_k = \phi_k \cup \{\phi_{k,i}\}$
9. else
10. Set $l_i = l_{i-1} + \Delta_{i+1}, s_k(\tilde{\phi}_{k,i}) = s_k(\tilde{\phi}_{k,i-1})$
11. end if
12. end while
13. $\phi_k = \phi_k \cup \{T\}, \{\phi_k, s_k = \tilde{s}_k(t), t \in \phi_k\}$ is a generated bsMJP path.
14. end while

4.8.2 Graphical model

![Figure 4.8: The graphical representation of the proposed model](image-url)
In this thesis, time series and sequences are thoroughly studied from two aspects, representation learning and prediction making. A detailed review summarizing existing approaches is provided. We observe that time series have been studied for decades and various models/learning methods are available, but adapting the existing methods to high-dimensional and noisy scenarios remains a challenge. The study on sequences/point processes is relatively new and improvements have been made in recent years. In this thesis, the main contributions are summarized as follows.

1. For time series representation, especially in high-dimensional and noisy scenarios, we present an idea to define the similarity between segments of the time series and learn a low-dimensional representation via non-linear dimension reduction. We assume that the time-varying distributions, rather than the samples themselves, are driven by few underlying controlling processes, yielding a low-dimensional smooth manifold in the domain of the distribution parameters. Further, we propose an efficient diffusion map approach based on distributional information of time-series data, which preserves the geodesic dis-
tances between samples on a statistical manifold, and uncovers an underlying process that consists of the controlling factors. We also complement the analysis with the diffusion maps approach based on the distributional information embodied in time series dynamics.

2. To learn time varying distributions, we propose a class of Bayesian models with various prior specifications such as autoregressive processes and Gaussian processes. We then apply the proposed models to two applications, music analysis and analysis of EEG recordings, and demonstrate how to specify the model adapting to different domains with prior knowledge.

3. For time series classification problems, we focus on the classifier design optimizing the performance in the low false alarm region. A partial AUC optimization approach is proposed, where the temporal information is incorporated by introducing hidden variables. We use the maximum margin approach to learn the model parameters defining the constraints between hidden variables with the largest discriminative power. The connections are built with other classification approaches including the HMM-based classifier and HCRF. Simulation results demonstrate the effectiveness of optimizing the performance in the low false alarm rate regions.

4. For representation of correlated point processes, we propose a continuous process model, which decomposes the point processes into a factor loading matrix and a small set of binary latent source functions. The factor loading matrix demonstrates the relationship among the point processes; meanwhile, the latent source functions (with on/off states) provide insights on how the point processes are evolving as a whole.

5. To infer the binary latent source functions, we propose an efficient method to
sample the continuous time binary state functions. For the binary state transition patterns, we allow flexible forms of hazard functions, to model various distributions over state durations (e.g., some states favor stickiness and some states tend to have a constant duration). We can also infer the number of latent source functions using the multiplicative Gamma process and improves interpretability.

Based on the work presented in this thesis, a few interesting directions are inspired, listed as follows.

1. In the line of representation learning and prediction of time series, capturing higher order statistics is an interesting and challenging direction to explore. In this work, we only allow first order correlation in the time domain, which can be relaxed to $p$-th order. How to infer the $(p + 1)$-way transition tensor in an $p$-th order HMM in an efficient manner remains an open problem (Sarkar and Dunson, 2015).

2. In most time series models, temporal correlation is incorporated as temporal smoothness, as also assumed in Chapter 2 and 3 in this thesis. But in reality, smooth evolving and occasional jumps are both observed in a series. The frequency, type, and amplitude of jumps are often informative in both representation learning and discriminative tasks. How to model such patterns without adding much complexity is an interesting problem to explore.

3. In Chapter 3, an algorithm designed for a semi-convex problem is proposed. Such non-convex objective is very common in time series and sequence modeling due to the introduction of hidden variables. Also because the number of hidden variables scales linearly with time, computation cost is often a concern.
Developing efficient incremental learning algorithms is a meaningful direction to explore.

4. In Chapter 4, we propose a method to model correlated arrival sequences (synchronized sequences of multiple subjects), inferring a factor loading matrix and a small set of latent source functions. We impose a sparse prior on the factor loading matrix; it is also interesting to add a structural prior to the loading matrix (e.g., clustering its rows, to infer clusters of subjects), or allow the loading matrix to vary with time (modeling the evolution of subjects). Another important extension is to incorporate side information of the subjects; e.g., for subjects that are people, covariates like age, sex, or profession can affect the loading matrix via a mapping function.

5. For point process modeling, we only focus on single-type event traces in this thesis. Exploring marked point processes remains a challenging problem, where categorical or continuous variables are associated with each arrival event. Efforts have been taken on modeling point processes with different event types. However, in many cases, each event is associated with a high-dimensional and possible sparse vector. How to model those variables and conduct joint analysis with the arrival sequences in a principled way is an open problem.
Bibliography


West, M. (1996), Bayesian Forecasting, Wiley Online Library.


Biography

Wenzhao Lian was born in Handan, China, on August 26th, 1989. In 2015, he got his Ph.D in Electrical and Computer Engineering, working with Dr. Lawrence Carin in the machine learning group at Duke University. He also got a Master’s degree in Statistics Science at Duke University in 2015, supervised by Dr. David Dunson. During his PhD, he did a few summer internships, including Yahoo Labs Sunnyvale in 2015, Microsoft Research Redmond in 2014, and Technicolor Research Bay Area in 2013. He got his Bachelor’s Degree in Electrical and Computer Engineering in Shanghai Jiao Tong University in 2011. In the year of 2013 and 2010, he went to Department of Mathematics at Yale University and Department of Electrical and Computer Engineering at University of Virginia as a visiting student, respectively. His current research interest includes machine learning and statistics, with a specialty in time series and point processes.

PUBLICATIONS


3. E. X. Chen, **W. Lian**, L. Carin, D. Brady, ”Task-driven Adaptive Sensing on
Quadrupole Mass Filter Systems for Classification,” Computational Optical Sensing and Imaging, 2015


103