SEMIPARAMETRIC TEMPORAL CLUSTERING

by

SUZANNE TAMANG

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2013

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction for the dissertation requirement for the degree of Doctor of Philosophy

Professor Simon Parsons

Date

Chair of Examining Committee

Professor Robert Haralick

Date

Executive Officer

Professor Noemie Elhadad, Columbia University

Professor Susan Epstein, Hunter College

Professor Heng Ji, Rensselaer Polytechnic Institute

Professor Andrew Rosenberg, Queens College

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

SEMIPARAMETRIC TEMPORAL CLUSTERING

by

Suzanne Tamang

Adviser: Professor Simon Parsons

Although temporal data provides critical context for many real-world reasoning tasks, incorporating the temporal dimension into an analysis can present methodological challenges. Traditional methods from statistics are limited in their ability to process noisy, large-scale secondary data sources. Data mining approaches are better suited for these types of problems, but have primarily focused on static data sets. However, few real-world data sets are static, or measure stationary phenomena; rather, they are dynamic.

To facilitate the meaningful use of abundant, unlabeled temporal data, I develop a new temporal clustering method that can assist in the preprocessing, exploration, and discovery of new knowledge from secondary data sources that are subject to arbitrary sampling schemes, and contain observation sequences of different durations. My approach builds on the semiparametric time series clustering framework, which has demonstrated clear benefits over fully parametric, or fully non-parametric methods. The framework combines beneficial parametric assumptions, such as the Markov or hidden-state assumption, to model temporal systems,

with a more agnostic, nonparametric approach for clustering the embedded models.

Using digital health data as a case study, I broaden the range of scenarios for which semiparametric clustering can be successfully applied. Specifically, I develop a method to use a state-of-the-art continuous-time Bayesian network to more naturally represent temporal information, addressing limitations of discrete-time methods. Also, as an alternative to spectral methods I pair model-based abstraction with a nonparametric Bayesian clustering technique that allows k to be expressed as a function of the size and complexity of the patient population, avoiding the requirement to prespecify the number of clusters using a heuristic.

To demonstrate the ability of this approach to produce meaningful results, clusters are evaluated using intrinsic and extrinsic validation. In addition, I compare cluster assignments with that of temporal clustering systems reported in the research literature, showing a 20% relative improvement over the best system's performance and recognizable differences among the patient clusters that are detected.

Acknowledgements

I would first like to thank the many people that have been involved with the completion of this dissertation. Without them this accomplishment would not have be possible. My advisor, Professor Simon Parsons, is not only a remarkable researcher, but also an incredible mentor. He provided me with both the flexibility to pursue my own interests, and the structure that is required to bring closure to a dissertation. In combination with my thesis advisor, I would like to express the deepest appreciation to my committee members. Their insightful comments and constructive feedback have been invaluable.

I am grateful to the Department of Biomedical Informatics at Columbia University for providing me access to clinical data, and for their help in shaping the preliminary research ideas that are further developed in my dissertation. I am also grateful to Brooklyn College, and the Macualay Honors College. Not only did CUNY fellowships provide me with financial support to continue my studies, but as a product of the New York City public education system, the opportunity that these fellowships afforded me to teach New York City students has been especially rewarding.

I thank my grandmother, who bought me my first computer. My mother for never ceasing to keep things interesting, and loving me unconditionally.

Lastly, I must express my sincere gratitude to Rave Harpaz. Of all the things to be grateful for, I can't imagine anything more wonderful than to share what you love, with the one you love.

Contents

List of Tables

List of Figures

List of Abbreviations

2-TBN 2-slice Temporal Bayes Net 35

ALB Albumin Test 86

BC Bayesian Clustering 136

BN Bayesian Network 29

ChE Cholinesterase Test 86

CT Continuous-Time 2

CT-BN Continuous-Time Bayesian Network 5

D-BIL Bilirubin 86

DAG Directed Acyclic Graph 34

DBN Dynamic Bayesian Network 35

DP Dirichlet Process 51

DPGMM Dirichlet Process Gaussian Mixture Model 55

DPMM Dirichlet Process Mixture Model 51

DT Discrete-Time 2

EHR Electronic Health Record 15

GMM Gaussian Mixture Model 53

HAI Hepatitis Activity Index 109

HMM Hidden Markov Model 1

HVB Hepatitis Virus B 95

HVC Hepatitis Virus C 95

MSM Multi-state Markov Model 5

MDS Multidimensional Scaling 154

PGM Probabilistic Graphical Model 4

SC Spectral Clustering 136

SSM State Space Model 30

ZTT Zinc Turbidity Test 86

Chapter 1

Introduction

The increasing availability of large data sets that track biological, behavioral and other phenomena, provides new opportunities for temporal learning. However, the secondary function of many real-world data sources as a research tool presents challenges for their analysis. To derive meaningful use of abundant, unlabelled temporal data, I develop a novel *temporal clustering method*. In contrast to traditional methods, this approach can be used for new knowledge discovery from noisy, heterogenous data sources. In addition, it can be used to cluster multivariate temporal sequences that are irregular in duration, irregularly sampled and subject to interval censoring.

My approach builds on the framework of semiparametric temporal clustering framework described by Jebara et al. [34], which couples nonparametric spectral clustering with parametric Hidden Markov Model (HMM)s. Based on the premise that parametric assumptions such as the Markov property or hidden state

representations are beneficial for modeling temporal data, and that it is advantageous to remain nonparametric and agnostic about the form of group structures in a temporal dataset, they apply their unsupervised learning method to motion capture, sign language and handwriting data sets. When compared to fully parametric mixture modeling, and a fully nonparametric method that pairs spectral clustering with nonparametric time series kernels, their results show consistent gains in performance.

Despite the ability of Markov models and their variants to capture temporal semantics, the Discrete-Time (DT) assumption, which requires that each measurement is represented by a series of values with uniform intervals indicated by the smallest time unit in the data set, Δt, makes DT methods less appropriate for modeling real-world data that is poorly structured and incomplete. For each time unit, missing data values must be imputed or provided using a default, forcing the representation of information without support. Also, DT models are less appropriate for modeling complex dynamic processes that may evolve in non-linear time versus linear time.

In this thesis, I describe a new method for clustering temporal data that is not well-suited for DT abstraction models. Specifically, I develop a technique that can be used to embed Continuous-Time (CT) Markov models for semiparametric temporal clustering. This approach can be used to generate a temporal representation that does not have the the limitations of discrete time approaches. Also, I describe how CT embedded models can be used to model multivariate data sets in which the individual variables may not be observed simultaneously. Lastly, in-

stead of using more popular spectral methods to cluster the embedded models, I propose a nonparametric Bayesian technique that is not limited by the requirement to prespecify the number of clusters, k, in advance.

1.1 Motivation

Clustering is a pervasive and natural human activity that affects knowledge representation and discovery [26]. Typically, we use it to group similar objects together, and establish criteria that are useful for their definition. Computational clustering algorithms can help reveal inherent group structure, or *clusters*, among observations without requiring the use of class labels for learning. For large data sets of high dimension, and where the distinguishing characteristics for establishing class boundaries are unknown, clustering is particularly useful for exploratory analysis, can enable the systematic examination of a data set, and can be used as a preprocessing step with the aim of enriching the quality of a data sample by filtering noisy or less informative examples.

For phenomena that evolve over time, the magnitude and direction of changes, and when changes occur, can provide critical context for reasoning. However, temporal modeling is challenging. Not only are there resource issues associated with the increased size of a temporal data set, but modeling choices related to the *representation of temporal granularity* and *sequential dependencies* must be considered. Also, in contrast to a primary data source, a secondary data source is more likely to contain noise, measurement sequences that are *irregular* in length, and to

reflect a variety of sampling schemes. These are all aspects that are problematic for existing approaches to temporal analysis.

1.2 Approach

A variety of learning algorithms have been developed to address the limitations of traditional approaches for modeling real-world time series sequences. Regardless of the task, the first step involves abstraction, a process of transforming the raw data into a high-level representation that facilitates description and comparison among sequences, typically to retain only information which is relevant for a particular purpose.

Abstraction techniques based on modeling a problem as a probabilistic graph, or Probabilistic Graphical Model (PGM), have been widely applied for sequential learning. The parametric Markovian assumption is an important simplifying assumption that captures temporal dependencies between time slices and allows for tractable inference. Also, the model probabilities can be updated to reflect new information.

For clustering time series, the *semiparametric clustering framework* [34] pairs parametric abstraction to model each singleton time series with a subsequent nonparametric clustering step that operates on the embedded models. Using embedded HMMs with spectral clustering, researchers have reported state-of-the-art performance on a variety of data sets [24, 23, 34, 77]. PGMs, specifically Markov and HMMs are used to exploit some parametric knowledge about the data, and si-

multaneously utilize nonparametric principles to remain more agnostic about the forms time series can adopt.

Using digital health data as a case study, my thesis extends Jebara et al.'s semiparametric clustering approach for time series, with the aim of broadening the range of scenarios for which the framework can be successfully applied. By default Markov models make discrete-time assumptions. Although these are appropriate for modeling many temporal data sets, they are not well-suited for processes that that are observed by arbitrary sampling schemes, or that may evolve on a non-linear trajectory.

To more naturally represent temporal data, and avoid forcing the representation of temporal measurements without support, the specific models I use to abstract temporal sequences for embedded clustering are based on finite state *Continuous-Time (CT) Markov processes*. Specifically, to learn each patient's model, I use Continuous-Time Bayesian Network (CT-BN)s [45, 46], which can be viewed as a CT extension of dynamic Bayesian networks. Instead of the characteristic transition matrix, P, that is associated with DT Markov models, I use the CT model equivalent, the intensity matrix, commonly symbolized as Q, to develop a method for clustering CT models.

It is important to note that CT-BNs share a common theoretical foundation in stochastic process theory with Multi-state Markov Model (MSM)s and were also motivated by the limitations of DT models. MSMs are an instance of CT-BNs that are used by biostatisticians and epidemiologists for modeling disease dynamics. In contrast to the use of CT models for learning patient-level temporal

models, MSMs are not used to model singleton time series, rather population-level dynamics. Typically, they are applied in the context of survival analysis, and on relatively small patient samples that are interval censored. Recent applications of MSMs for disease modeling include HIV [25, 51], dementia [55], lung disease [69], cancer [72], and various other medical conditions [33].

In addition to CT abstraction for the purpose of embedding, I extend the clustering step to the nonparametric Bayesian setting, which allows the number of clusters, k, to be expressed as a function of the size and complexity of the data. A major limitation of spectral methods is the need to establish k using a heuristic $a\ priori$. Also, research has shown that humans employ multiple strategies for finding k, and even on simple data sets the number of possible interpretations can be high [38]. For describing the phenotypes of complex patient types, an approach that can permit the interpretation of multiple clustering views can be beneficial.

Lastly, the reasons described above motivate this work but it is important to note the following. Although unsupervised learning methods have many successful applications, one hindrance to their practical adoption is the difficulty of interpreting results. Many clustering metrics are based on a mathematical interpretation of clustering as a partitioning problem that is independent of the clustering algorithm [11, 26, 38]. For this reason, I evaluate my method using established validation metrics, and use visual tools to help communicate any clinical significance.

1.3 Contributions

The central contribution of this thesis is to introduce a new learning method for the exploratory analysis of non-canonical time series data. Although, clinically significant work applying exploratory techniques to clinical data has been demonstrated [42, 56], it has focused on monitoring patients in the critical care setting, where data is sampled at a high frequency and the clinical concerns are more immediate. In contrast, chronic disease progression can take months or years to manifest, and longer-term trends can have increased importance.

More specifically, the main contributions of this thesis are:

- I introduce a temporal clustering method designed to utilize the temporal data that serves a secondary purpose as a research instrument.

- I describe how continuous-time model based abstraction can be used to provide a principled approach to regularizing time series that are variable in length, subject to arbitrary sampling schemes, and reflect irregular intervals.

- I develop a continuous-time abstraction method for clinical data that more directly models observations, making an important connection between continuous time disease models in biostatistics and continuous-time Bayesian networks from computer science.

- I extend the semiparametric framework to the nonparametric Bayesian setting. This no longer requires that k is fixed, or a required parameter for clustering.

- I demonstrate that my clustering approach can produce meaningful clusters from noisy, incomplete patient data and evaluate results on intrinsic and extrinsic validation methods.

In addition, I evaluate semiparametric Bayesian clustering and show that my clustering approach improves on state-of-the-art performance by comparison with a benchmark system on gold-standard results. To assess the relative performance to alternatives, I compare nonparametric Bayesian clustering with spectral methods for clustering patient models.

Lastly, I facilitate the interpretation of semiparametric Bayesian clustering results by normalizing cluster-level characteristics so that group signatures can more easily be in terms of "instantaneous risk" using a visualization I call the Q-matrix Plot, and I discuss the clinical significance of clustering results, and what findings have a potential translation to clinical practice.

1.4 Overview

This work extends applications of semiparametric clustering methods for temporal data and describes two experiments using clinical data sets to evaluate performance.

The background on temporal analysis methods, including semiparametric clustering is discussed in more detail in Chapter 2. I describe a variety of methods for temporal abstraction in Chapter 3. Their relationship to clustering, and a comparison of clustering methods are provided in Chapter 4. My novel contributions

begin in Chapter 5, where I discuss continuous-time representations for patient level disease modeling, the integration of nonparametric Bayesian clustering. I evaluate the performance of this new approach to semi-parametric clustering on two data patient-level data sets, including a hepatitis data set and a glucose data set in Chapters 6 and 7. Finally, a summary of this work, the key conclusions and next steps appear in Chapter 8.

1.5 Publications

The work described in this thesis has appeared in several publications:

- Tamang S and Parsons S. (2013a). Semiparametric Clustering Methods for Modeling Chronic Disease Dynamics. *In Proceedings of the International Conference on Machine Learning, Workshop on the Role of Machine Learning in Transforming Healthcare.*

- Tamang S and Parsons S. (2013b). Unsupervised Modeling of Patient-level Disease Dynamics. *In Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium, Workshop on Data Driven Wellness.*

- Tamang S and Parsons S. (2011). Using semi-parametric clustering applied to EHR time series data. *In Proceedings of the Knowledge Discovery and Data Mining, Workshop on Data Mining for Medicine and Healthcare.*

In Chapter 7, I describe preliminary work for this thesis using HMM abstraction (Tamang and Parsons, 2011). The main contribution of this work was that it demonstrated the appropriateness of spectral clustering by comparing results with k-means clustering and of model-based abstraction, which performed better than feature based abstraction using aggregate time series statistics. This finding was produced regardless of the abstraction method used and based on intrinsic validation metrics.

In this thesis, I extend semiparametric clustering to continuous-time model abstraction and the nonparametric Bayesian setting (Tamang and Parsons, 2013a; 2013b). This approach is detailed in Chapter 5, and referred to *semiparametric Bayesian clustering*. To evaluate performance I conduct two experiments to assess the performance of this method and that appear in Chapters 6 and 7.

Chapter 2

Preliminaries

Time is undoubtedly one of the most mysterious concepts. We may learn to 'tell time', but our understanding is so natural, and empirically enforced that we never need to explicitly be be told or reminded of its basic qualities. However, when analyzed more critically, time is paradoxical, and its nature, construction and impact is understood more abstractly than tangibly.

In this chapter, I define some basic terms, related to temporal analysis, that are used by various disciplines to describe time-related data, highlight the main challenges for automating temporal inference, and describe the key characteristics of model-based clustering, of which semiparametric temporal clustering is a new and evolving instance.

2.1 Temporal Analysis

Depending on which book you may open on temporal analysis methods, different names for time-related observation data may be used. For that reason, I define some key terms used in the literature, and clarify how they are used in my thesis.

This work focuses specifically on *temporal data*, which, when measured uniformly, are a special form of sequential data generally referred to as *time series*.

Sequential data is the most minimal of temporal types, conveying only that the measurements are in order. For example, sequential data can exist in the form of word order in a sentence, or it can exist in a more explicit temporal form that is generated by a process. Therefore, all time series are sequential, but not all sequential data are time series.

When temporal data is related to the same entity, it is sometimes referred to as *panel data*, or *longitudinal data*. For example, a cohort study that tracks participants at 6 months and one year after baseline measurements are recorded may be considered panel data by an economist and longitudinal data by a biostatistician. Sometimes panel data provides the additional meaning that data is interval censored.

To model temporal data, critical assumptions are made by automated reasoning methods:

- a underlying characteristic of unidirectionality, that defines time as an infinite, forward moving trajectory, and
- that information about the past can be propagated along this temporal tra-

jectory, or a quality of persistence that allows information to travel through time and contribute to a history.

These two assumptions are perhaps the most basic, and are consistent with an enormous amount of evidence that can be observed in both our internal and external environments.

Methods for modeling temporal data vary most dramatically not in a rejection of these basic tenets, but rather the assumptions that facilitate data description and tractable analysis. Although many are known to be false, (e.g. the assumption that the a temporal process can be approximated by modeling one snapshot), these important simplifying assumptions are either required for any analysis at all, or are assumed to present a limited effect on results.

2.2 Challenges for Temporal Learning

Recently data technologies have provided humans with the potential to mine massive data sets with complex temporal patterns. However, the growth of temporal data has outpaced the development of temporal learning algorithms to process them effectively.

Although existing data mining approaches have proven their ability to reveal unseen patterns and summarize data in meaningful ways, there is an unmet need for high performance techniques specific to temporal data. Temporal mining is a special case of data mining that seeks to address the methodological issues presented by real-world databases that are temporal in nature. Specifically, temporal

mining is appropriate for data sources that are large-scale, variable in duration, incomplete, irregularly sampled and subject to other types of noise. Tasks typical of these methods include: data characterization and comparison, clustering, classification and other types of pattern analysis.

Temporal Clustering

When little prior knowledge about group structure is known and a collection of observations is too large for a human to peruse, unsupervised-learning algorithms, also known as unsupervised classification, can be useful.

At the minimal level, automated clustering can be viewed at preprocessing with the goal of improving the performance of a system. For example, in a collection of patients with a potentially lethal disease, clustering can help screen patients with minimal risk. A consequent processing step can focus on the set of patients more likely to develop a terminal condition.

Temporal clustering methods are important for several reasons:

- time series data can capture important characteristics of complex dynamic processes, but are problematic for many traditional data mining and temporal analysis techniques,
- clustering methods have been shown to provide a powerful tool for allowing researchers to examine relatively unexplored data repositories that are too large to manually peruse,
- for many new applications, labeled data does not exist and the cost of label-

ing training examples can consume too many resources to allow for supervised learning,

- clustering can detect group structure in data that are not predicted by the researchers current knowledge

2.2.1 Case Study: Longitudinal Patient Data

Temporal data provides important clinical context for the diagnosis, prognosis and treatment of many diseases [5, 6, 36, 50, 66, 69, 70]. Historically, following a cohort of individuals entailed the design of customized measurement tools to monitor patients for a research study and explicit checkpoints for data collection.

One new source of longitudinal patient data that is receiving increased attention by researchers is digital health data collected in institutional Electronic Health Record (EHR)s. These are information rich records of a patient's medical history, and typically contain various types and amounts of data, including demographics, clinical notes, allergies, medications, vital signs, previous aliments and diseases, immunizations, laboratory data and radiology reports. These can be used to provide a population-based view of phenotypic responses to the progression of disease and treatment.

Advocates of these systems hope that the richness of available patient data contained in these collections will enable a feedback loop of new knowledge discovery and translation to practice, supporting the engineering of improved healthcare systems. There is an urgent need to demonstrate the benefit of maintaining

petabytes of patient data, and an important role for probabilistic learning algorithms that can assist in the discovery new temporal knowledge from these noisy, heterogenous, fragmented data collections.

The specific type of longitudinal data used for our experiments consists of indicators for chronic disease, which for a disease like hepatitis may take decades to progress. As a case study, I focus on two types of disease-related observations. The first data set contains temporal indicators for chronic hepatitis patients that at are at risk of fibrosis of the liver and other disease related medical conditions. The second data set is related to diabetes and provides temporal indicators for glucose management. The broad goal is to cluster patients into disease related risk types based on temporal indicators that can help to describe key clinical characteristics and inform treatment decisions by clinicians.

2.3 Model-based Clustering

Using digital health data as a case study, my work extends the settings for which semiparametric clustering, a type of model-based clustering can be applied. In this section, I describe the conceptual model for model based clustering and provide an detailed description of semiparametric clustering.

Figure 2.1 shows the general steps of model-based time series clustering, which involves: *preprocessing* of the raw data, *temporal abstraction* to approximate the time series in a more concise form, and a *clustering* algorithm that operates on the models, or abstractions, instead of the raw time series measurements.

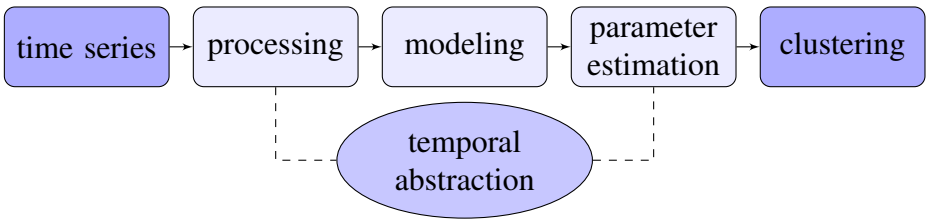

Figure 2.1: General time series clustering framework

There have been various methods proposed for model-based abstraction in the recent literature and applications of probabilistic graphical models (PGMs), specifically Markov models and their variants, for clustering sequential processes is an increasingly active active area of research [22, 34, 48]. Applications can be traced back to early work by Smyth [61], who first described a generalization of the standard mixture model approach using HMMs for clustering sequences.

One aspect of modeling temporal phenomena as a probabilistic graph that makes them particularly attractive for modeling sequential phenomena is the expressive language that can be applied to describe a process. Conditional dependencies can be represented with one time slice to indicate conditional distributions associated with the process, and between time slices to reflect temporal dependencies.

Also, a benefit for new problem adaption is that problem semantics and algorithmic components are loosely coupled. That is, when an analysis for a new data set is required, a model must be created to reflect problem's semantics, but all of the algorithmic parts lend themselves to immediate reuse.

2.3.1 Clustering with Mixtures of HMMs

The use of Markov-based approaches to model time series has been shown to produce high performance results in several domains. The beneficial parametric assumptions they make are used for approximating temporal dependencies and variable values. Almost all existing model-based methods use a Markov chain, many with extensions for hidden states, to represent the temporal dynamics of observation sequence. Among the various techniques that appear in the research literature on Markov-model abstraction, the key distinction is the model type and how they are used for clustering.

One of the most popular model-based clustering approaches in the literature, describes a time series dataset as generated by K-HMM components. It is commonly referred to a clustering with mixtures of HMMs. For example, we can describe a dynamic process using:

$$f_K(O) = \sum_{k=1}^{K} f_k(O|\theta_k)p_k$$

where O is a time series, p_k is the weight of the kth HMM and $f_k(O|\theta_k)$is the probability that the series was generated by the component model f_k with parameter θ_k. The clustering task is defined as an extension of a standard mixture model task, and the problem is to define the best K models that define the population [58, 61, 65, 78].

2.3.2 Semiparamertic Clustering

One more recent approach to model-based abstraction that has been applied for a variety of time series modeling tasks [24, 23, 34, 67, 77] is the general framework of *semiparametric clustering*. As practiced, this approach pairs a fully parametric maximum likelihood estimation approach to abstract each singleton time series with a nonparametric clustering step that uses embedded models as input. This framework provides a principled approach to transforming variable length temporal sequences into a succinct representation that facilitates their use by traditional multivariate learning algorithms.

Based on a literature search, the first implementation of semiparametric clustering pairing Markov model based abstraction with spectral clustering methods was published by Yin and Yang [77]. However, the theoretical justification for the hybrid method, and a comparative evaluation with fully parametric and fully nonparametric alternatives is provided by Jebara et al. [34]. Both works, show better performance relative to clustering with mixtures of HMMs, and in the case of the latter, which used a probability product kernel to construct the similarity matrix for clustering, reports results comparable to that which was achieved with supervised learning.

Overview

Figure 2.2 shows an overview of a *semiparametric clustering* framework for temporal modeling. As eluded to earlier, the name of the method describes the two

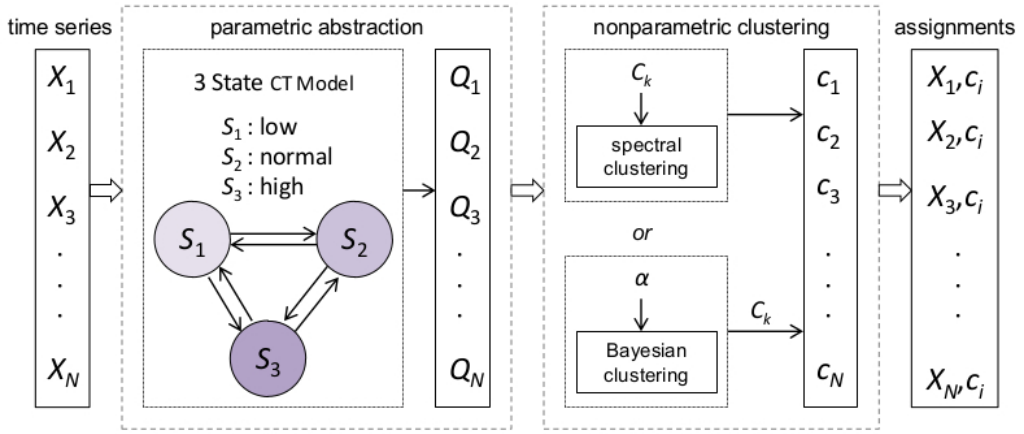

Figure 2.2: An overview of semiparametric clustering

key components to the technique, which provides a principled way of transforming and clustering arbitrarily sampled, irregular length observations and still provides tractable inference.

Step 1 The first step entails temporal abstraction using a parametric model to transform a set of raw singleton time series for N observations, $X_1, X_1, ..., X_N$, into a more manageable form for traditional multivariate clustering algorithms. It entails learning each patients model-specific parameters, $Q_1, Q_2, ..., Q_N$, from the time series observations typically using expectation maximization (EM) [16] based methods such as the forward-backward algorithm.

Step 2 The second step is nonparametric clustering. Spectral methods are typically used in the semiparametric framework, which requires determining the number of clusters, k, in advance. The models, $Q_1, Q_2, ..., Q_N$, instead of

raw time series sequences, $X_1, X_1, ..., X_N$, are embedded for nonparametric clustering.

Chapter 3

Temporal Abstraction

In this chapter, I discuss popular methods for representing temporal data for use in probabilistic learning applications, or methods known as *temporal abstractions*. More generally, *abstraction* is defined as the process of transforming a concept or an observable phenomenon in to a more succinct form, typically to retain only information which is relevant for an application.

For the purpose of inference, the goal of temporal abstraction is to provide a concise, high-level description of a sequential process that facilitates description and comparison among sequences, while simultaneously preserving the information contained in the raw data. Abstractions provide the benefit of being more comprehensible to humans than raw data, and facilitateS subsequent analysis by knowledge discovery and data mining methods.

3.1 Representations

Regardless of the temporal modeling task, abstraction is almost always the first step after the data has been preprocessed. Most all temporal analysis methods integrate an abstraction technique. These techniques can be used on their own for descriptive purposes or in conjunction with a subsequent processing step.

Some researchers [39] argue that abstraction is the single most important step for temporal analysis. That is, if the quality of the approximation is high, the results that are achieved will be similar to what would be achieved using the raw data. If abstraction is poor, dynamics are not sufficiently captured and can lead to false findings.

True or not, it is obvious that the choice of high-level representation is a key decision in any temporal mining framework, and especially important when data is irregularly sampled and noisy. In this case, abstraction can help to mitigate the impact of noise, and transform uneven length observation lengths into a uniform format that allows for further processing.

One way to describe an abstraction approach is in terms of how a time series is input for a learning task:

Raw Data In general, raw data it not feasibly processed by clustering methods. For small, structured, or synthetically derived data, clustering on the raw data sequence is possible.

Feature-based Many feature extraction methods are simple in nature and help to characterize the shape of the trajectory that is monitored. They aim to

measure a quality about the observation sequence instead of the underlying data generation process and use the similarity among features for clustering. Also, features tend to be domain, task and data set specific, and are hard to generalize. For example, a feature that works well for one data set may be irrelevant to another application.

Model-based Working under the assumption that each time series was generated by an underlying process that can be modeled, the similarity between different models serves as the basis for time series clustering. There have been various methods proposed for model-based abstraction in the recent literature and applications of Markov and hidden Markov modes for temporal modeling is an increasingly active area of research [22, 34, 61].

3.1.1 Subsequence and Whole Sequence Approaches

In addition to abstraction types, we can also describe an abstraction method in terms of the information represented in the approximated values. In the past, work with subsequences was pervasive. However, a study by Keogh et al. [40] that showed the limitations of subsequence time series (STS) based approaches resulted in a shift towards whole sequence analysis.

The main claim of [40] that "clustering of time series sequences is meaningless" was relevant to sliding-window abstraction methods. The study replicated numerous published studies presenting additional results that invalidated previously published work (including the authors' own). The work showed that subse-

quence time series clustering methods that had been published to that point (2002) generated output that was independent of the input.

Although there has since been work that addresses the shortcomings noted in this key paper, for many domains and applications these findings still hold relevance. They point to the limitations of sliding window methods, their problem with trivial matches, the limitations of distance metrics for determining the similarity of sequences, and the challenges posed by developing new ones.

3.2 Moving Average Models

The most widely used family of techniques for modeling time are based on autoregressive techniques, known collectively as autoregressive moving average or ARMA models [4]. These techniques make beneficial assumption to model the dependencies between adjacent observations time series, but don't directly represent problem semantics.

The name ARMA comes from the combination of autoregressive (AR) features with moving average (MA) models to from ARMA(p, q) models, and the models use a predetermined, fixed size temporal window that slides along the entire duration of the sequence. Individual time points are transformed into a series of subsequence measurements based on the data observed in each time window.

Autoregressive (AR) models assume that a current value depends on previous periods p and white noise ϵ_t. For a model of order p, with model parameters ϕ at

time t $AR(p)$ is defined as:

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t \tag{3.1}$$

A *moving average* model of order q, $MA(q)$ with parameters θ and white noise ϵ_t is defined as:

$$y_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{3.2}$$

ARMA models combine the two model types $AR(p)$ (Eq.3.1) and $MA(q)$ (Eq.3.2) to define $ARMA(p, q)$ by:

$$y_t = \epsilon_t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

The strengths of ARMA techniques for time series modeling includes their ease of interpretation, prediction quality, and their ability to extend to multivariate settings. However, they are limited by the use of a fixed length temporal window, their inability to compare variable length time series, and other model features that make them more appropriate for canonical time series, but not the detection of complex temporal patterns from more methodologically challenging temporal data sets.

Additional requirements of ARMA models are typically that the developer must have extensive knowledge and experience with the process that generated the data, and the technique requires a substantial amount of data preprocessing and parameter tuning.

3.3 Spectrum Analysis

Some well-known approaches to feature extraction that have continued to be enhanced since their introduction are spectral techniques. These are borrowed from signal-processing and include Discrete Fourier Transform (DFT) [27], Discrete Wavelet Transforms (DWT) [59], and Piecewise Linear Approximation (PLA) [17].

The basic idea is to view a sequence as a signal and then to decompose this signal into its characteristic parts. This type of analysis is most applicable when a process is best described as the sum of individual frequency components. For example, the QRS complex, P wave and T waves that are detected by an electrocardiogram.

3.4 Graphical Models

An increasingly popular type of model of which certain types can provide useful parametric assumptions for modeling time are Probabilistic Graphical Model (PGM)s. Key modeling concepts are represented by random variables, "nodes", for which a graphical structure represents the conditional dependencies between nodes as "edges" which can be undirected or directed to indication the flow of information.

The graphical model formalism provides general algorithms for the calculation of marginal and conditional probabilities for a set of observations generated by various types of processes and there are many methods devised to construct and utilize them to provide new information while controlling computa-

tional costs [35]. Also, these techniques provide the flexibility for modeling complex problems, and each graph's semantics are separate from the algorithms that can be applied to the structures [37]. Since the algorithmic components can be reused more easily, and show good performance on very simple problem models, these techniques have already demonstrated success for many types of supervised and unsupervised learning in across seemingly unrelated domains.

An example of a graphical model with four random variables is shown in Figure 3.1, which is a directed acyclic graph (DAG) with the conditional independence relations in $P(A, B, C, D)$

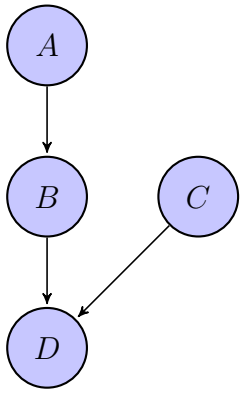

Figure 3.1: Graphical model with four random variables and directed edges

For temporal abstraction, PGMs can be used to transform the sequence into a new representation, and avoid the need to align and compare two sequences directly using distance metrics. Also, they more naturally capture temporal correlations among neighboring states, and the importance of recent events relative to those in the distant past, and can model whole time series sequences [22, 32, 37] .

To model the dynamics of a phenomena, the graph of the underlying process is instantiated for the time slices, and edges are placed between these slices oriented in the direction of the temporal trajectory.

3.4.1 Graphical Model Types for Temporal Data

Probabilistic Graphical Model (PGM)s come in a variety of types. My work focuses on modeling the temporal aspect of data, and for that reason I limit further discussion to the use of graphical models in this context.

Types of PGMs are not mutually exclusive, and often many well known models for temporal learning can be viewed as special classes or instances of a broader category. Also, the same type of model can be referred to by different names in various research communities. For example, a Markov model, also called a Markov chain, is also a Bayesian Network (BN) [15] that represents a first-order Markov process.

To describe some of the differences between popular temporal models, I define the common types of models and the properties that are key to understanding how to construct an informative but concise abstraction of a system's temporal dynamics using the language of PGMs. Also, the exact same model may be referred to differently, even by researchers in the same field. By providing this clarification, I hope to help eliminate some of the terminological baggage that can cause confusion for others reading the literature on PGMs, and that may not immediately see the connections or commonalities.

Examples of models are shown here with model states indicated by a lighter

color, and observations appear darker. Conditional dependencies are indicated with arrows. An example of this is shown in Figure 3.2, which shows a dependence relation between the state Q, and an observation, O, at time i.

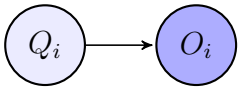

Figure 3.2: Examples of a state, Q, and an observation, O, at time i

State Space Model (SSM)

A SSM is a stochastic dynamical system that assumes an underlying state, or set of states, that are associated with a temporal phenomena and generates observations over time. An SSM requires the definition of state transition probabilities, $P(Q_t|Q_{t-1})$, indicating the probability from transiting from one state to any other, the observation probabilities, $P(Y_t|Q_t)$, indicating the probability of an observation given the current state value, and the initial state probability distributions π. The state's value can be continuous or discrete. A well studied SSM is the *Kalman filter*, which has continuous valued states.

Markov Models

The *Markov property* states that the conditional probability distribution of a future state Q_{t+1} is conditionally independent of states Q_i, where $i < t$, given Q_t. In other words, given full knowledge of the current state, the future and the past are independent.

In general, if a simple but informative state description for a dynamic process can be defined, a *Markov assumption* is generally a reasonable approximation of the dependencies in a distribution [37]. It is one of the most important simplifying assumptions for temporal PGMs in that it can allow for tractable inference even for complex processes.

A system for which the Markov assumption holds is considered a *Markovian system*, and the underlying process is a *Markov process*.

A *Markov model* is a SSM with a Markovian assumption, and by default such models are discrete-time representations. Q_t is represented by a single discrete random variable. Figure 3.3 shows a Markov model with 3 states Q_1, Q_2, Q_3. It is also an example of a *Markov chain* or a Bayesian network representing a first-order Markov process.

For the purpose of inference, the probability tables are known, or can be learned with techniques such as the forward-backward algorithm [52] which is detailed in Section 3.4.3.

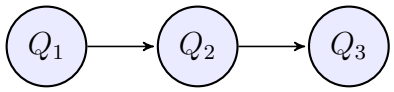

Figure 3.3: A Markov model for time series of length $T = 3$

Hidden Markov Model (HMM)

Hidden Markov Model (HMM)s are extensions of Markov models that are used when model states are latent but for which correlated indicator variables exist. For

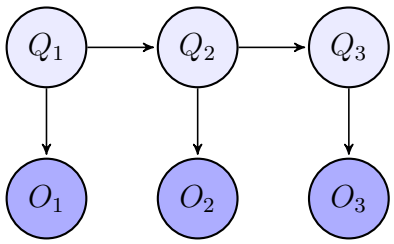

Figure 3.4: A HMM for time series of length $T = 3$, where at any time $t \in \{1, ..., T\}$ the value of a hidden model state is Q_t and O_t is the corresponding observation.

HMMs, Q_t is represented by a single 'hidden' discrete random variable.

Figure 3.4 shows a HMM with 3 hidden states Q_1, Q_2, Q_3 that can assume any of the M possible values and are correlated with the set of observations, O_1, O_2, O_3. Like states, the values of observations come from a pre-defined set. For the purpose of inference, these probability tables are known, or can be learned with optimization techniques such as the forward-backward algorithm [52]. A more detailed description of HMMs are provided by Rabiner [52].

For the random variables in the temporal model, given the values of the variables in the previous model state the distribution of trajectories can be defined as the product of the conditional distributions and calculated using the chain rule. More formally:

$$P(Q_T) = \prod_{t=1}^{T} P(Q_0)P(Q_{t+1}|Q_t)$$

HMM Variants

Autoregressive HMMs are used for modeling when the current state of the model may not always capture all the information available in previous states. For time series data it can be helpful to assume that an observation at time $t-1$ will be an indicator of the observation at time t, especially when the signal is noisy. This property can be incorporated into the model by using a directed edge between O_{t-1} and O_t as shown in Figure 3.5. The conditional distribution for the autoregressive HMM described above is $P(Q_t|Q_{t-1}) = P(Q_t|Q_{t-1}, O_{t-1})$.

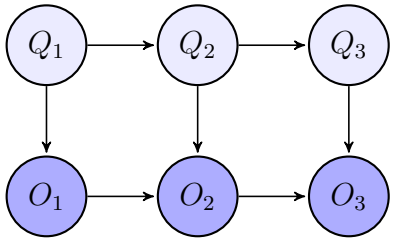

Figure 3.5: Autoregressive Markov model

Other popular HMM variants include *factorial HMMs*. When other sources of information are independent, new states and variables can be used to distribute the hidden state, Z, among covariates. Figure 3.6 shows the graphical model for a two state factorial representation. The conditional distribution for a factorial HMM when each state variable $n \in \{1, ..., N\}$ is inferred using the formula:

$$P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_t^{(i)}|Z_{t-1}^{(i)})$$

The last HMM variant I note is the semi-Markov HMM. Relaxing the Markov

assumption provides many new opportunities to model temporal dependencies. Some examples of these types are n-grams, mixed memory HMMs, and sliding HMMs with extended horizons. These allow for the inference of the current state to be conditioned on n previous states, where $n > 1$. For example, a second order Markov model defines $P(Q_t|Q_{t-1}, Q_{t-2})$, and a third order model requires $P(Q_t|Q_{t-1}, Q_{t-2}, Q_{t-2})$.

Bayesian Network

Bayesian Network (BN)s, or Bayes nets, are a broad class of PGMs that are represented as Directed Acyclic Graph (DAG)s, where nodes represent random variables of interest, edges represent informational or causal dependencies among variables, and each node is conditionally independent of its non-descendants given the parents. Since they allow for a richer structural versatility than the models discussed so far, Bayes nets allow for more complex models to be represented.

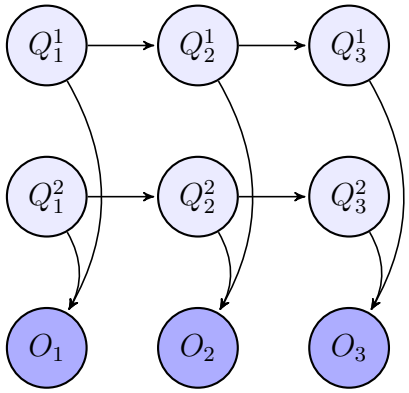

Figure 3.6: Factorial HMM with 2 hidden states

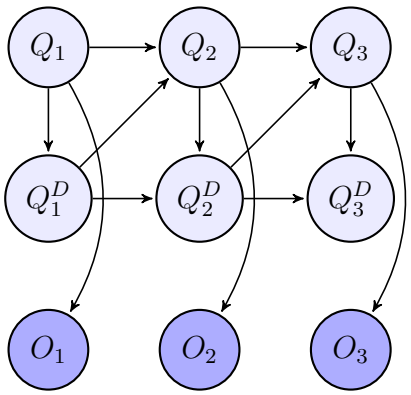

Figure 3.7: Semi-Markov HMM

More formally, let X_i be a node in the BN, X, over the variables $X_1, ..., X_n$ and let Pa(X_i) be the parents of X_i. Based on the independence assumptions, we can define the joint probability distribution $P(X)$ using the chain rule for BNs as follows:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i)|Pa(X_i)$$

3.4.2 Dynamic Bayesian Networks

For modeling time series, we can construct a Dynamic Bayesian Network (DBN) out of a BN, where this BN functions as a *template*, by instantiating the set of template variables for each time slice. The set of template variables are repeated for each point in the series and directed edges are added between time slices in the direction of time to reflect the template variables that interface between slices. In a 2-slice Temporal Bayes Net (2-TBN) edges are of two main types: *inter-time-slice edges* that interface between time slices or *intra-time-slice-edges* that reflect

the conditional dependencies among variables and within a single time slice.

DBNs formalize temporal dynamics to represent a time series of length T as a collection of discrete, ordered points corresponding with a specific model state Q that can assume one of M possible values so that for any t, $Q_t \in \{1, ..., M\}$.

DBNs extend the basic HMM model in that they are able to capture richer semantic information than HMMs, allowing temporal phenomena to be modeled more accurately. For example, in a time slice DBN multiple model states may be represented; i.e., a DBN time slice $Z_t^{(i)}$ permits a set of n states where $i \in \{1, ..., n\}$. Also, a DBN permits a set of corresponding observations for i that can assume discrete or continuous values.

The corresponding conditional distribution for the transition model of a 2-TBN makes two simplifying assumptions: (1) a dynamic system over the template variables X satisfies the Markovian system and (2) a Markovian dynamic system is time invariant, so that $P(X_{t+1}|X)$ is the same for all values of t. Since time is an infinite trajectory, these assumptions allow for modeling the state transition model more compactly.

Let Z be a set of n persistent template variables that transfer temporal information in a 2-TBN. We can represent the conditional distribution as:

$$P(Z_{t+1}|Z_t) = \prod_{i=1}^{n} P(Z_{t+1}^{(i)}|Pa(Z_{t+1}^{(i)}))$$

For example, Figure 3.8 is a simple DBN model for monitoring a patient's disease acuity. The variables are defined as: A=Access to care, C=Compliance,

D=Disease management, R=comoRbidities. The model's structure conveys problem knowledge such as the level of compliance with treatment (C) is conditionally dependent on access to care (A). Assuming all nodes are persistent, we can represent the conditional distribution based on the formula:

$$P(Z_{t+1}|Z_t) = P(Z^A_{t+1}|Z^A_t)P(Z^C_{t+1}|Z^A_{t+1}Z^C_t)P(Z^D_{t+1}|Z^C_{t+1}Z^D_t Z^R_{t+1})P(Z^R_{t+1}|Z^R_t)$$

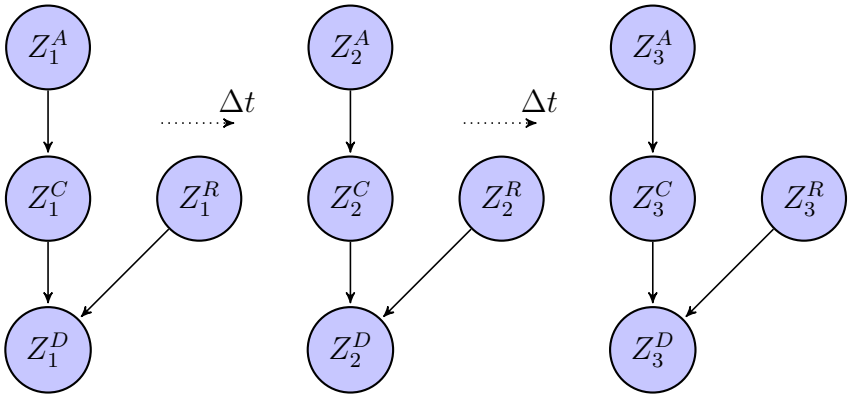

Figure 3.8: Dynamic Bayesian network (DBN, $T = 3$)

Connections: Bayesian Networks and Markov Models

The connection between BNs and Markov models can be described as follows. A simple Markov model can be viewed as a BN specifying the conditional independence relations consistent with a Markovian system. Although they can be used to model time, BNs are defined by instantiations of a template BN. At each time slice, the parents of the Markov model's state are not apparent. The template that is repeated is not a BN. However a HMM, which has a parent node at each time

slice is a simple DBN or a BN specifying the conditional independence relations for a HMM.

Making the connection between HMMs and BNs allows one to relate them to more complex models, and integrate general solutions that have been developed for construction, inference and learning in BNs [62, 63].

3.4.3 Markov Processes

For abstracting temporal sequences, hidden and Markovian assumptions are used for modeling. In this section I further describe the theoretical aspects related to a process that are assumed to be approximately Markovian, and how to calculate model parameters for a HMM based on the observation sequence. The modification for a Markov model is a simplification of the HMM method, and does not need to consider the emission matrix, resulting in a more straightforward calculation.

Markovian models are based on the underlying assumption that the future state of a system, Q_{t+1}, is independent of all past states, given the current state Q_t. HMMs and other DBNs with latent variables are useful for representing phenomena that care not directly observed, but for which a correlated variable can provide sufficient information to make inferences about the latent or *hidden* state's value.

Forward-backward Algorithm

HMMs are defined by a triple, $\{A, B, \pi\}$ over a set of discrete states and distinct observations. The matrix A represents the state transitions, or the probability of

moving from the current state, q_i, to the next state, q_{i+1}. The model parameter B indicates the probability of an observation value for each $q \in Q$.

To learn the parameters we can use the forward-backward algorithm, an instance of the Expectation-Maximization algorithm [16]. Given an observation sequence, $O = o_1, ..., o_T$, where T is the length, and $Q = q_1, ..., q_T$ is a corresponding state sequence composes of N possible random variables, we adjust the model parameters, $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$.

For discrete models, it is common to estimate the most likely model that generated the observed data for a Markov chain. The approach makes three key assumptions:

1. the states and observations are fixed, and the emission and initial distribution probabilities are unknown.

2. for each patient n, data consists of observed state sequences, $X_n = \{x_0, ...x_d\}$

3. we can incorporate priors (uniform, random, or informed)

To demonstrate how it works, I show the brute force approach that involves estimating the probabilities for all of the sequences, the forward and backwards equations, and how they are integrated into an EM framework, commonly referred to as the Baum-Welch algorithm or reestimation [52].

Brute-force Approach: A brute force approach enumerates every possible state sequence. For sequence Q equation 3.3 shows the probability of the observation sequences, given the state sequence and the HMM model, λ, that can be calculated from the emission matrix, B. The probability of the sequence is

given by equation 3.4, and both equations are used to define the joint probability, equation 3.5

$$P(O|Q,\lambda) = \prod_{t+1}^{T} P(O_t|q_t,\lambda) = \prod_{t=1}^{T} b_{q_t} o_t \tag{3.3}$$

$$P(Q|\lambda) = P(q_1) \prod_{t=2}^{T} P(q_t|q_{t-1}) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1}} q_t \tag{3.4}$$

$$P(O,Q|\lambda) = P(Q|\lambda)P(O|Q,\lambda) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1}} q_t \prod_{t=1}^{T} b_{q_t} o_t \tag{3.5}$$

Since each q_i can be represented by one of n random state variables, in order to maximize $P(O|\lambda)$ we need to calculate N^T possible state sequence representations, performing about $2T$ calculations for each. Therefore, the brute-force approach is a problem in the order of $2TN^T$.

Forward-Backward Procedure: The forward-backward algorithm is a dynamic programming algorithm that is closely related to the Viterbi algorithm for decoding, another learning task common to HMMs that involves selecting the most likely state sequence at time t based on an observation at time $t+1$.

The procedure decomposes parameter estimation into two parts:

1. for state i and at time i, the forward algorithm, $\alpha_t(i)$, moves forward in time, and estimates the probabilities that λ will output an observation sequence $O_{1,t}$ and land in state s_i.

2. the backward algorithm, $\beta_t(i)$, estimates the probability that starting in state s_i and at time t the rest of the observation sequence $O_{t+1,T}$ will be generated

These are combined to estimate the probability of observing $O_{1,T}$ and being in

state s_i at time t as follows:

$$P(O_{1,T}, q_t = s_i) = \alpha_t(i) * \beta_t(i)$$

Forward Procedure: The forward probability for a state s_j at time t, $\alpha_j(t)$, is the probability of the partial observation sequence $O_{0,t}$, with state s_j at time t

$$a_j(t) = P(o_1 o_2 o_t, q_t = S_j|\lambda) \tag{3.6}$$

We can compute this inductively beginning with the initial state probability matrix π where $1 \leq j \leq N$ and $1 \leq t \leq T-1$

$$\begin{aligned} \alpha_j(1) &= \pi_j b_j o_1 \\ \alpha_j(t+1) &= \left(\sum_{i=1}^{N} \alpha_i(t) a_{ij} \right) b_{j_{o_{t+1}}} \end{aligned} \tag{3.7}$$

then calculating the likelihoods takes N^2T operations

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, q_T = s_i|\lambda) = \sum_{i=1}^{N} \alpha_i(T) \tag{3.8}$$

Backward Procedure: The backward probability is the probability of the partial observation sequence after time t given s_i at time t, and represented by $\beta_i(t)$.

$$\beta_i(t) = P(o_{t+1}o_{t+2}o_T, q_t = S_i|\lambda) \tag{3.9}$$

We can compute this inductively, this time starting at the end of the sequence, where $1 \leq i \leq N$ and $2 \leq t \leq T$

$$\begin{aligned} \beta_i(1) &= 1 \\ \beta_i(t-1) &= \sum_{j=1}^{N} a_{ij} b_{jo_t} \beta_j(t) \end{aligned} \tag{3.10}$$

Baum-Welch Reestimation: I have shown that the problem of maximizing $P(O|\lambda)$ is in the order of $2TN^T$. There is no known analytic solution, but we can use EM to find a local maxima using the Baum-Welch algorithm.

First we define $\epsilon_{ij}(t)$ the probability of being in state s_i at time t and state s_j at $t+1$. Using the forward procedure (Eqn. 3.6) and the backward procedures (Eqn. 3.6), which can both be computed inductively as shown in Eqns. 3.7 and 3.10, the Baum Welch algorithm is

$$\epsilon_{ij}(t) = P(q_t = s_i, q_{t+1} = s_j|O, \lambda) \tag{3.11}$$

$$= \frac{\alpha_i(t) a_{ij} b_{j_{o_{t+1}}} \beta_j(t+1)}{P(O|\lambda)} \tag{3.12}$$

$$= \frac{\alpha_i(t) a_{ij} b_{j_{o_{t+1}}} \beta_j(t+1)}{\Sigma_{i=1}^{N} \Sigma_{j=1}^{N} \alpha_i(t) a_{ij} b_{j_{o_{t+1}}} \beta_j(t+1)} \tag{3.13}$$

Chapter 4

Temporal Clustering

Clustering is both a pervasive activity and a natural human activity that affects knowledge representation and discovery [26]. Typically, we use it to group similar objects together, and establish criteria that are useful for their definition. Computational clustering algorithms can help reveal inherent group structures, or clusters, among observations without requiring the use of class labels for learning.

Clustering has been successfully applied to identify important regions in animal genomes that correspond with biomarkers for diseases such as cancer [19, 64], to automatically identify themes in text [8, 9, 73], and to many other machine learning tasks where labeled data may be not be readily available [34, 50, 75]. In terms of clustering the entities in our datasets, we make the assumption that a clustering assignment is valid if and only if the indicators for the phenomena within a group are correlated, but indicators for patients in different groups are uncorrelated or not as strongly correlated.

For large data sets that are high dimensional, and where the distinguishing characteristics for establishing class boundaries are unknown, clustering is particularly attractive for exploratory analysis, and can enable a systematic examination of a data set to inform the generation of new hypotheses. In addition, clustering can be used as a preprocessing step with the aim of enriching the quality of a data sample by filtering out noisy or less informative examples.

4.1 Comparing Methods

To design a computational clustering algorithm, the task can be simplified to that of identifying a partition of n observations $x_i, ..., x_n$, so that each observation x is grouped into one of k disjoint clusters, such that each x_i is assigned membership in one group, where points in the same group are similar and points in different groups are dissimilar to each other.

Different clustering algorithms have been developed to suit a variety of tasks and types of data, and one way to broadly describe them is in terms of *parametric assumptions*. In statistics, the term 'nonparametric' refers to an assumption that the data lacks predefined characteristics. In practice, the parametric versus nonparametric distinction for a clustering algorithm is more of a propensity than a strict conformity. Although an algorithm can have both parametric and nonparametric qualities, the qualification is useful to describe and compare alternative methods. Nonparametric clustering methods aim to be more agnostic about clustering parameters, e.g. the number or shape of component clusters, or the number

of clusters.

When choosing a clustering algorithm for a new task, it is important to consider characteristics of group that the data modeler aims to discover. If a common pattern relevant to the distribution of observations is known, or the number of classes has been already established, then parametric information should be exploited by the algorithm. However, clustering is most desired when the data is poorly defined and often parametric information about group structure is not available for complex temporal data. If incorrect parametric assumptions are made, it can generate results that could be less meaningful in terms of revealing unexpected relationships than if they were not biased. Therefore, in the case when there is limited evidence to assume parameters for the clustering model, nonparametric techniques are typically preferred.

Another important characteristic for comparing clustering algorithms, and their appropriateness for temporal data is the choice of *similarity metric*. A common approach to evaluating similarities among a collection of observations is to measure the distance between them and there are many techniques that can be used, including: Euclidian distance, Manhattan distance, and edit based distance. Distance based similarity approaches are most effective when it can be reasonably assumed that the variables in each observation are independent and identically distributed (iid). However, in the case of time-series data, where adjacent observations are more likely to be correlated with those in close proximity, and the dimension is unidirectional, the development of accurate distance metrics can be more challenging. For these reasons, techniques that can abstract a time series

into less correlated features, or more accurately assess similarity directly from the time series, have been an active area of research for a long time.

I provide a summary of the benefits and limitations of some methods that have been applied for temporal clustering in the Appendix, Table A.2. The three clustering methods that are described in more detail in this section are implemented in my work including k-means, spectral clustering and nonparametric Bayesian clustering.

To provide a baseline clustering approach that is simple to implement, I use a feature-based clustering method with *k-means*. The two nonparametric methods I apply for clustering parametric temporal models include *spectral clustering* and *nonparametric Bayesian clustering*. Spectral clustering, has been shown to be well suited for the semiparametric clustering framework and a popular choice. However, it is limited by the need to express the number of clusters $a\ priori$. In this thesis, I propose to pair nonparametric Bayesian clustering. which does not have this limitation, as an alternative to spectral clustering.

4.1.1 *k*-Means

The most popular clustering technique is k-means. It is simple, efficient and performs well on many tasks. Using unlabeled training set examples, the algorithm uses an iterative method to partition a data set of n values into k clusters without being told their categories. It begins by randomly assigning k points as the initial 'seed' representatives or centroids. After the initial assignment of k points, an iterative process of reassignment based on the 'closest' centroid, is repeated until a

convergence criterion is met (e.g., the squared error ceases to decrease or there is no reassignment of centroid location) or until a pre-specified number of iterations is reached.

The objective function of k-means is to minimize the total intra-cluster variance, and the common measure is the sum of the squared error:

$$V = \sum_{k=1}^{K} \sum_{X \in C_k} ||X - \mu_k||^2$$

where μ_k is the center, or mean of cluster C_k, and where $||X - \mu_k||$ is the distance between a point in cluster C_k and the cluster's centroid. The default measure of distance is Euclidian distance, with ties broken arbitrarily.

Limitations of k-means include sensitivity to initialization, faltering when the data set is not naturally represented as spherically shaped clusters, and the presence of outliers, which can substantially influence the location of centroid centers.

4.1.2 Spectral Clustering

State of the art semiparametric clustering methods have typically used embedded Markov or hidden Markov models as input to a nonparametric spectral clustering method. *Spectral graph theory* has been applied in many fields to address the limitations posed by centroid-based definitions of a cluster. Spectral techniques are based on performing the eigenanalysis of graphs. *Spectral clustering* recasts graph partitioning as a eigenvector problem.

Eigen- is a German word for 'self' or 'characteristic'. Eigenvalues are derived

from a $n \times n$ (square) matrix and typically represented as λ. If $Ax = \lambda x$ and x, where A is a $n \times n$ matrix and x is a non-zero vector, then x is the *eigenvector*.

Eigenvectors are also known as 'steady-state' vectors. For example, if a Markov chain converges after many steps, any row of that matrix is an eigenvector for A^T. Eigenvectors are useful for characterizing the motion structure, and the eigenvector of highest magnitude, the 'dominant eigenvector', can be described as a natural frequency of the motion.

Spectral clustering provides a nonparametric approach by using eigenvector segmentation, or graph partitioning based on graph cuts. Matrix theory allows for the rewriting of a matrix in terms of smaller matrices, or blocks. In the context of clustering, the blocks correspond to bunches, or groups, of segmented points where the similarity among points in the same bunch is high, but low relative to other points in alternative bunches.

Similarity Graphs

Using a graph, $G = (V, E)$ where V is a set of observations represented as vertices $\{x_i, ..., x_n\}$ and E a set of edges representing the similarity between observations, spectral clustering algorithms formalize the partitioning problem with a variety of different approaches [44, 60]. There is no clear best method, but what is common to all of them is the use of an $n \times n$ matrix to store values to indicate the strength of the relation, $x_{i,j}$ where i and $j \in \{1, ..., n\}$, weighted by similarity.

The similarity matrix, S, which can be constructed using a variety of ways, a weighted adjacency matrix of G is the matrix $W = (w_{ij})i, j = 1, ..., n$ repre-

senting the weights between all connected points, or 'affinity'. The more similar x_i and x_j, the higher the value. In order to make computation more efficient, the pairwise similarities for only the local neighborhood to a point i may appear in the graphs weighted adjacency matrix W. Another approach to computing the similarity between any two observations is the Gaussian similarity function:

$$s_{ij} = d(x_i, x_j) = \exp\left\{\frac{||x_i - x_j||}{\sigma^2}\right\}$$

where the parameter σ controls the width of local neighborhoods in the data. In W, if $w_{ij} = 0$, then x_i and x_j are not connected by an edge.

The minimum cut of the weighted adjacency matrix, W, determines the optimal partitioning of the dataset. A cut between any two subgraphs is calculated as follows:

$$Cut(C_1, C_2) = \sum_{i \in C_1} \sum_{i \in C_2} w_{ij}$$

Graph Laplacians

Spectral clustering algorithms recursively partition a data set by identifying the minimum cut and removing the edges until k clusters are identified. The problem of identifying the minimum cut is NP-hard; however there are more efficient approximations that are based on linear algebra using graph Laplacians and their basic properties. Graph Laplacians are the main tools for spectral methods.

From W we can derive the degree matrix, D, defined as the diagonal matrix

of the degrees $d_1, ..., d_n$, where each degree of a vertex $x_i \in V$ is determined by:

$$d_i = \sum_{j=1} nw_{ij}$$

.

The unnormalized graph Laplacian, L, is defined using the degree matrix, D, and W, the weighted adjacency matrix that is generated from a similarity graph:

$$L = D - W$$

and used to identify the first k eigenvectors, $v_1, ..., v_k$, that are used to create an $n \times K$ eigenvector transformation of the data V, by stacking the eigenvectors in columns. For $i, ..., n$, let y_i be the vector corresponding with the i-th row of V. Operating on V, k-means is then used for clustering the points $y_i, ..., y_n$ into clusters $C_1, ..., C_k$. In the final step of the spectral clustering algorithm, the n points are projected back to the initial data representation by assigning each $x_i \in X$; if row i of the matrix V was assigned to the cluster C_j, where $j \in \{1, ..., k\}$, then x_i is a member of C_j

4.1.3 Nonparametric Bayesian Clustering

A nonparametric alternative to spectral methods that does not require that k is indicated in advance are Bayesian clustering methods. Bayesian nonparametric models have been applied to both supervised and unsupervised learning tasks where

it's desirable for the number of modeling parameters to adapt with the complexity of the data. For this reason, they are labeled 'nonparametric'. Often named by the processes they are used to model, they can be used for clustering, as a density estimator, as features for regression, and more.

Although k is not required, a prior on the distribution of the sample is assumed. One flexible prior often used in Bayesian statistics is the *Dirichlet*, which is a conjugate prior that describes the multivariate extension of the Beta distribution. For nonparametric Bayesian clustering, this is nicely suited for a two-level Bayesian hierarchy, where the base measure for a data set is assumed to be a Dirichlet distribution.

It is important to note that there are several clustering approaches based on the nonparametric empirical Bayes estimation of Dirichlet priors. To develop a probabilistic generative model of a data sample, one can model each component as a Dirichlet Process (DP), and pose the clustering problem as that of estimating the parameters of the Dirichlet Process Mixture Model (DPMM).

The most popular nonparametric Bayesian clustering method is latent Dirichlet allocation (LDA), which has been widely applied for discovering hidden thematic structure from a document corpus, and was developed by Blei et al. [10]. To model a corpus with a distribution of hidden themes, only a two-level hierarchical Bayesian model is required. To model a collection of documents that can be associated with multiple topics, LDA uses a more sophisticated three-level hierarchy. Also, LDA has extensions for dynamic topic modeling [8].

Another method for nonparametric Bayesian clustering was developed by Heller

et al. [28] and is coined 'Bayesian Hierarchical Clustering'. The clustering approach uses DPMMs for agglomerative hierarchical clustering, which outputs a binary tree with internal nodes that correspond to clusters, and leaves that represent data points. Cooke et al. [13] extend Bayesian Hierarchical Clustering to the temporal setting by developing a technique for clustering gene expression data.

Although nonparametric Bayesian clustering can vary in the number of hierarchies they use, the choice of model priors, and other features to reflect problem semantics, the approaches share and underlying foundation. In this section, I describe the details of the two-level hierarchical Bayesian model.

Hierarchical Bayesian Model

A Bayesian model provides an inherent hierarchical structure that can be applied to clustering. Any Bayesian approach begins with using prior probabilities to describe what is known at the current state. By combining the prior probabilities with new observations, new information is incorporated into the model, resulting in new posterior probabilities.

Figure 4.1 shows a two level prior distribution for a hierarchical Bayesian model. The upper level represents the probability distribution $f(p|h)$ conditional on the set of hyperparameters h, and the first level probability distribution $f(\theta|p)$ that is conditional on hyperprior p.

For clustering real-world time series datasets, a two-level hierarchical Bayesian model offers several advantages:

1. a large number of real-world datasets provide information about a population-

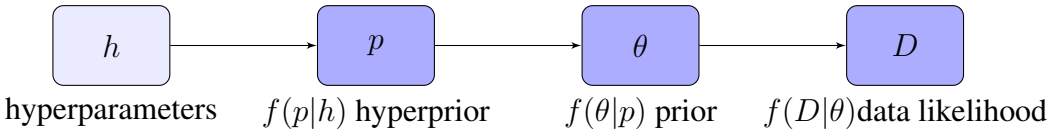

Figure 4.1: Two level hierarchical Bayesian model

based phenomena for which some prior information about group variability is known,

2. certainty about prior information is explicitly provided to the estimation procedure,

3. it allows us to interpret results in terms of population based and group based heterogeneity, and

4. when the size of the dataset is small, it is less prone to overfitting than other approaches.

Dirichlet Process Mixture Model (DPMM)

By defining the clustering problem as identifying the components of an infinite mixture where k is a random variable in the model, nonparametric Bayesian approaches allow k to be determined by the size and complexity of the data. One method is performed by taking the limit of the number of mixture components, k, as a hierarchical Gaussian Mixture Model (GMM) approaches infinity.

In this section, I first discuss how the method is used for finite mixture models, and then expand the finite case to that of one with infinite components. Then

I describe characteristics of DPGMMs, and how they are used for unsupervised learning.

Finite and Infinite Mixture Model

Nonparametric Bayesian methods can be used to identify the number of component, and their densities, in a finite mixture model. The density function of a finite mixture model is defined as

$$p(x) = \sum_{k=1}^{K} \pi_k p(x|\theta_k)$$

where x is the data set, π is the mixing proportion, and θ_k are the model parameters for the cluster k.

We can define the discrete case in the form of the integral as [47]

$$p(x) = \int p(x|\theta)G(\theta)d\theta, \text{ where } G = \sum_{k=1}^{K} \pi_k \delta_{\theta_k} \tag{4.1}$$

In the case of infinite components the extension is expressed as the following for a Bayesian nonparametric models with a potentially infinite value of k

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \tag{4.2}$$

Dirichlet Distribution

In Bayesian statistics, a *Dirichlet distribution*, Dir(α), is the conjugate prior of the categorical distribution and multinomial distribution. In terms of a finite mix-

ture model with k components, it is a k-dimensional generalization of the beta distribution.

Since we can view model components as groups, or mixtures in the data, with a two-level hierarchy it has natural extensions for clustering. Specifically, if a population can be described by the probability distribution Θ with components $\theta_1, ..., \theta_k$ that sum to 1, we can reasonably infer that

$$\Theta \sim \text{Dirichlet}\{\alpha_1, ..., \alpha_k\}$$

The probability density function for a Dirichlet distribution uses a normalization factor that is defined in terms of the multinomial beta function, $B(\alpha)$, that is expressed in terms of the gamma function

$$B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i)}$$

and the probabilities p and parameters α of each of the k components, or clusters

$$\text{Dirichlet}(p; \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} p_i^{\alpha_i - 1}$$

Dirichlet Process Gaussian Mixture Model (DPGMM)

One approach to nonparametric Bayesian clustering is *Dirichlet process Gaussian mixture modeling*. Although this technique was initially applied for density estimation, it is increasingly being used for clustering applications [47].

A Dirichlet Process (DP) is the prior over the mixing distribution, G, and

defines a measure on measures. It is characterized by two parameters: a base distribution G_0, from which samples are drawn, and is the parameter on which the nonparametric distribution is centered. The second is a positive scaling parameter α. More intuitively described as a 'splitting' criteria, α is a scaling factor that is associated with the probability of forming a new cluster. The base distribution is defined by G_0.

$$G \sim DP(G_0, \alpha)$$

For a sample, G, drawn from the base distribution G_0, if $G \sim DP(G_0, \alpha)$, then for any set of partitions $A_1 \cup A_2 \cup ...A_k$ of A:

$$(G(A_1), ..., G(A_k)) \sim Dir(\alpha G_0(A_1), ..., \alpha G_0(A_k))$$

To automatically determine k, Dirichlet Process Gaussian Mixture Modeling defines a DPMM as that of one with infinite components as shown in Eq. 4.2, taking the limit, k, as the number of unseen, Gaussian components approaches infinity.

4.2 Model Validation

Despite successful applications, an ongoing problem discussed in the literature is the evaluation of clustering results. Although there has been exciting theoretical work in clustering, and many applications demonstrate meaningful results, there is an need to establish a better and 'deeper science' than is currently offered to

address the issues that are independent of specific clustering methods [11, 26, 41].

For many data sets, the search for a true or gold standard maybe futile. When working with multi-faceted processes such as health, which can be assessed on the phenotypic and genotypic level, contradictions can appear when considering only one. Similarly, classification of organisms also poses diverging opinions. Despite the hundreds of years of scientific examination by the most notable biologists in the world, and more recent innovations that allow for the sequencing of entire genomes, there are still arguments about the system of biological nomenclature [68].

Although there has been a variety of metrics for measuring the extrinsic and intrinsic quality of groups, some still consider clustering a craft rather than a science. However, we continue to perform clustering, and use these metrics to compare results with good reason. We're accumulating digital data at an exponential rate and methods that can be used to preprocess data, making consequent analytic steps or human perusal easier, or to reveal unseen, meaningful pattern that were not predicted by researchers, are highly desired. Although, validation metrics may not guarantee meaningful clusters, they are very useful for the development of new clustering methods.

4.2.1 Cluster Quality

For the purpose of evaluating clustering algorithms, one can assess *intrinsic* and *extrinsic* cluster quality. Metrics used in practice enable these judgements to be made in terms of abstract properties that exist independently of the data set. How-

ever, these metrics make assumptions about conceptual questions such as "what is an *optimal* clustering", which may not be possible to define. Research shows that humans employ multiple strategies for finding the number of clusters, k, and even on simple data sets the number of possible interpretations can be high [38].

Despite their short-comings, validation metrics are useful for developing systems and evaluating clustering results, and researchers continue use and improve them. A cluster can be informally described as a maximally coherent subset, C, that satisfies both inter and intra-cluster criterion:

- items in C should be homogeneous in type
- no larger cluster should contain C as a proper subset

These generic qualities of an optimal clustering are generally agreed upon, and serve as the basis for developing metrics, such as purity, B-cubed, V-measure and others.

For simple data sets, evaluation can be straight-forward. However, for more complex tasks some challenges that are in direct conflict with evaluating clustering results based on established metrics alone are:

- for the same set of observations, an optimal clustering cannot be established
- the relative importance of clusters may be unequal and dependent on problem context; i.e. maximal coherency may be more important for certain clusters, and
- categorical similarity can be non-metric.

Intrinsic Validation

When a gold standard is unavailable to evaluate clustering performance, heuristics can be used to assess the *intrinsic quality* of clusters.

Silhouettes

One common heuristic that helps to quantify the intrinsic goodness and visualize cluster differences is the silhouette method [54]. For a clustering assignment, the data set's silhouette is defined by the difference of the average dissimilarity of a point to members or its own cluster with that of the 'neighboring' cluster over the max of these two dissimilarity measures.

For example, the silhouette validation technique can be used to validate patient clusters when human judgement is not available. For each patient, let $a(i)$ be the average dissimilarity of patient i to all patients in its respective cluster. We then calculate the average dissimilarity of patient i with patients of another cluster, repeating for all clusters that patient i is not a member of. The cluster with the lowest average dissimilarity is the 'neighboring cluster' of patient i and indicated by $b(i)$. The resulting silhouette value is defined as:

$$s(i) = \frac{(b(i) - a(i))}{\max\,(a(i), b(i))}$$

and the average $s(i)$ of a clustering assignment is a measure of how tightly patients are grouped into their respective clusters and how distinct clusters are with respect to each other.

When $s(i) \geq 0.6$, patient i can be considered to be appropriately clustered. A value close to -1 indicates that a patient would have been more appropriately assigned to the neighboring cluster, $b(i)$, and a value close to 1 indicates that individuals in the patient's respective cluster are very similar and that the cluster is distinct from other clusters.

Extrinsic Validation

A clustering assignment that demonstrates high intrinsic quality may not always translate to high *extrinsic quality*. When available, comparison with a gold-standard is preferable. However, the choice of clustering metric is important and many that are commonly used fail to meet basic "goodness" criteria.

Criteria for external clustering metrics are discussed in Amigo et al. [1], Rosenberg et al. [53] and Pelillo [41]. Although additional criteria are established by some of these researchers for high quality clusters, two central themes are apparent – their emphasis on the importance of cluster homogeneity and completeness. *Homogeneity* measures the ability of clusters to group similar members together and the maximum is reached when all groups contain members of only one type. *Completeness* of a cluster is achieved when the cluster is maximal for the members of a specific type.

Figures 4.2 and 4.3 show two examples to help illustrate homogeneity and completeness. Each class label is represented by one of four colors, purple, pink, blue or gray, and the boundaries of discovered clusters are indicated by the large circular envelopes.

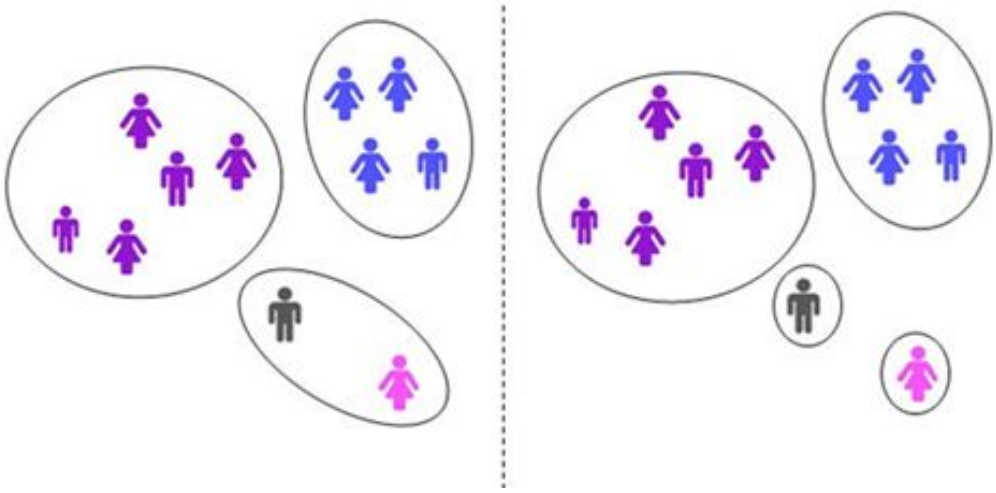

Figure 4.2: As points are reassigned, completeness is equivalent and homogeneity increases

On the right-hand side of Figure 4.2 is a perfect clustering. Each cluster contains only elements of the same type, and each cluster is maximal for members of a specific type. Although some metrics, such as purity, measure only one of these cluster qualities, the examples in Figure 4.2 and Figure 4.3 show why a measure with the ability to assess homogeneity and completeness is preferable for evaluating cluster quality.

Figure 4.2 shows a reassignment from left to right where the completeness of the assignments is equivalent, but the homogeneity increases. In the initial cluster composition on the left, gray and pink members appear together in the same cluster. In the reassignment on the right, pink and gray entities appear in their own cluster, and all clusters reach maximum homogeneity.

Figure 4.3 shows a reassignment from right to left where the homogeneity of the clusters are equivalent, but completeness increases. The initial assignment on

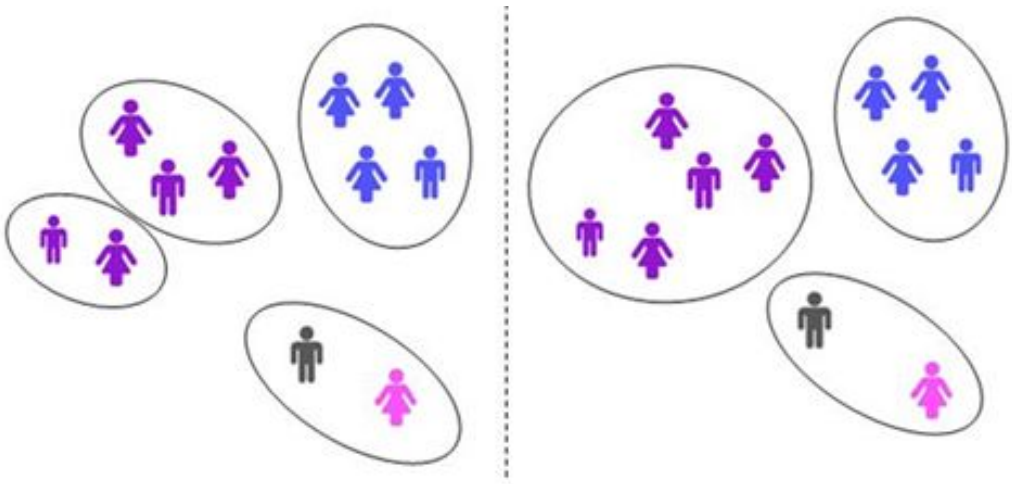

Figure 4.3: As points are reassigned, homogeneity is equivalent and completeness increases

the left shows the two clusters that are only composed of purple entities. In the second assignment, these two clusters appear in the same class, and the cluster is now maximal for purple entities.

B-cubed

To evaluate information retrieval systems, the *F1 score* is the established performance metric. It is the harmonic mean of the *precision*, or the fraction of instances that are relevant, and the *recall*, or the fraction of instances that are retrieved.

B-cubed [3] extends the F-measure to clustering. It decomposes precision and recall to each datum in order to evaluate a pointwise average for the data set. This metric satisfies key criteria of homogeneity and completeness that is associated with better clustering metrics.

The B-cubed score is calculated as follows. For each entity e, where $e \in$

$\{1, ..., N\}$, the B-cubed precision and B-cubed recall score for the assignment is

$$\text{B-cubed Precision} = \frac{\sum_{e=1}^{N} P(e)}{N} \tag{4.3}$$

$$\text{B-cubed Recall} = \frac{\sum_{e=1}^{N} R(e)}{N} \tag{4.4}$$

Similar to F1, the B-cubed recall and precision scores can be combined. Here, P refers to the B-cubed precision (Eq. 4.3) and R refers to the B-cubed recall (Eq. 4.4)

$$\text{B-cubed score} = 2\left(\frac{P * R}{P + R}\right) \tag{4.5}$$

An example for calculating pointwise precision and recall for a clustering assignment is shown in Figure 4.4. Each class label is represented by one of three colors, purple, blue or gray, and the boundaries of discovered clusters are indicated by the large circular envelopes. Entity e, who has a class of purple, is a member of a cluster where four of the five entities share e's class, resulting in pointwise precision of 0.80. However, not all purple entities appear in e's cluster. Two have been assigned to another cluster. Only four of the six purple entities appear in e's class and the pointwise recall of e is 0.67.

To evaluate a clustering, the pointwise precision (Eq. 4.3) and recall (Eq. 4.4) are averaged over the population and used to calculate the combined B-cubed score (Eq. 4.5).

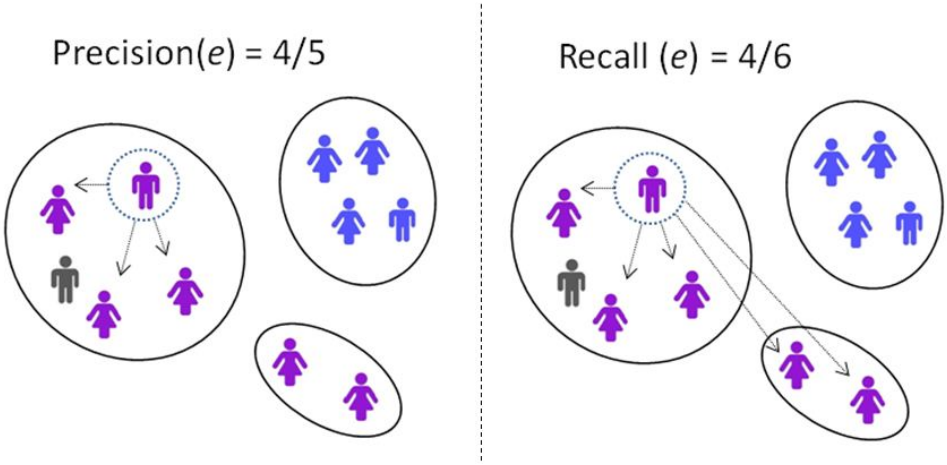

Figure 4.4: Calculation of pointwise precision and recall for an entity e.

Motivation for B-cubed

Although entropy-based metrics such as the V-measure [53] also satisfy important formal constraints associated with a good clustering metrics, there is a domain correspondence that makes the B-cubed metric better suited for evaluating patient clusters. B-cubed shares features with statistics that are commonly used by medical researchers to assess the performance of classification tasks such as diagnosis. Specifically, sensitivity (Eqn. 4.6), specificity (Eqn. 4.6), and positive predictive value (Eqn. 4.7). Entropy-based metrics are not nearly as common in the medical literature, and for that reason, these scores maybe less intuitive to end users of a clustering application for chronic disease patients.

Table 4.1: Relationship between B-cubed, sensitivity and specificity.

	Gold Standard	
Outcome	**Condition Positive**	**Condition Negative**
Test Positive	TP	FP
Test Negative	FN	TN

$$\text{Sensitivity} = \text{Recall} = \frac{TP}{Condition\ Positive} = \frac{TP}{TP + FN} \tag{4.6}$$

$$\text{Precision} = \text{PPV} = \frac{TP}{Outcome\ Positive} = \frac{TP}{TP + FP} \tag{4.7}$$

$$\text{Specificity} = \frac{TN}{Condition\ Negative} = \frac{TN}{TN + FP} \tag{4.8}$$

$$\tag{4.9}$$

Sensitivity can be interpreted as the ability of a diagnostic test to detect a condition, and based on the relationships shown in Table 4.1 is equivalent to recall, the first component of the B-cubed score. The more intuitive way to think of sensitivity is the probability of a positive test, given that the patient is ill. In the same way precision and recall are paired in information retrieval, sensitivity is paired with *specificity*, the probability of misdiagnosing a diseased patient as unhealthy, or 'false alarms'. Precision, the second component of the B-cubed score, is also known as the *positive predictive value* (PPV) in biostatistics. Therefore, an alternative way to describe the B-cubed score is that it is the per patient average the PPV and sensitivity.

4.2.2 Importance of Problem Context

Clustering encompasses a wide range of problem types, many of which can be expressed in a taxonomy to help clarify design decisions and evaluation of the results [11, 26]. However, validation metrics are for the most part agnostic about the clustering procedure and deeper problem context.

In contrast, to derive meaning from a grouping, humans consider the nature of the relations between entities, and the clustering goal. For this reason, the treatment of clustering as an application-independent mathematical problem with associated metrics is limiting. Therefore, in addition to validation metrics, interpreting clustering results in the context of problem semantics is also an important part of determining if a clustering application has produced meaningful results.

Chapter 5

Semiparametric Bayesian Clustering

In this Chapter, I describe the details of a new semiparametric Bayesian clustering method that I developed to handle the type of temporal data that can be found in clinical records. The approach extends the range of scenarios for which semiparametric clustering has been applied. As noted earlier, both of the clinical data sets used in my experiments are secondary data sources that represent challenges for temporal modeling in that they do not fit the canonical time series description, and instead are subject to various level of incompleteness, contain variable length observations, and reflect arbitrary sampling schemes. More details on the model extensions that are data set specific (e.g. learning model states) are described in the methods section for the relevant experiment.

5.1 Modeling Chronic Disease Dynamics

Health services researchers generally agree that Electronic Health Record (EHR) data provides many opportunities for new knowledge discovery, and there is tremendous potential for value beyond the meaningful use standards that have been defined by the Centers for Medicare and Medicaid Services to incentivise the adoption of EHRs [43]. For example, one of the most significant issues facing the US healthcare system is the growth of chronic diseases. Temporal clinical data can provide important clinical context for the diagnosis and prognosis of chronic diseases, and can be used to provide important insights into chronic disease trajectories that are not well-understood.

Longitudinal studies have a long history in medical research. However, historically, following a cohort of individuals entailed the design of customized measurement tools to monitoring patients for a research study and explicit checkpoints for data collection. Now, data is pervasive but it of a is lower quality. Traditional longitudinal analysis methods are limited in their ability to operate on non-canonical time series, in the presence of system noise, and other problematic features that are associated with modeling from patient records.

Although clinically significant work applying exploratory techniques for patient and population level disease modeling have been developed, they are most appropriate for methods that learn from data that has been captured in a high frequency sampling setting. For example, Marlin et al. use probabilistic learning methods to cluster patients using physiological signals collected in the ICU [42].

Their data is based on a per patient 24 hour snapshot. Saria et al. [56] develop a technique to detect important signals for monitoring neo-natal ICU data using nonparametric Bayesian methods for learning model features, clinical 'topic models', that correspond to factors associated with newborn morbidity and mortality.

In contrast to the critical care environment, the type of data that is available for learning chronic disease dynamics from medical records is sampled in a low frequency setting. Chronic disease progression can take months or years, or decades to manifest, and progressive trends instead of significant features can have increased importance. Since patient observations are typically documented only during hospital or physician visits, the data is subject to arbitrary sampling schemes and interval censoring. Also, a patient's measurement sequence can be short or can span over many years.

To cluster variable length, noisy time series, the *semiparametric clustering framework* pairs a parametric abstraction method with a nonparametric clustering step. This framework is described in more detail in Section 2.3.2. Recent work has demonstrated success pairing discrete-time HMM abstraction with spectral clustering [24, 23, 34, 77].

With medical record data in mind, I aim to broaden the range of scenarios for which the framework can be successfully applied with two key extensions: (1) Continuous-Time (CT) models for abstracting temporal information, and (2) nonparametric Bayesian clustering.

5.2 Continuous-time Model Abstraction

Regardless of the temporal learning task, the approximation of temporal data, 'abstraction' to a more concise representation, is critical for large, irregularly sampled data sets. Probabilistic graphical models for modeling dynamic phenomena have demonstrated their use for a variety of abstraction tasks, but make a simplifying discrete-time assumption that can be problematic when data is sparse, and can be missing.

By default, Markov models and their variants make Discrete-Time (DT) assumptions. The DT models we have discussed in Chapter 3 provide useful simplifying assumptions for modeling many temporal data sets, but they are less good at modeling chronic disease data extracted from clinical records. There are key limitations that have been noted [33, 45, 46, 57] and are directly relevant to the type of data typically found in health provider databases:

- if the underlying health related phenomena that is being modeled progresses in individuals at different rates, the smallest granularity must be used to express time steps for the entire system,
- when data is unavailable, intervening time slices must still be represented, and
- chronic disease progression can occur in non-linear time.

To more naturally represent temporal data, and avoid forcing the representation of temporal measurements without support, I use Continuous-Time Bayesian

Network (CT-BN)s [45, 46]. CT-BNs are based on finite state *CT Markov processes*, which connect them with continuous-time models used in biostatistics, Multi-state Markov Model (MSM)s.

In my work, I make an important connection between MSMs and CT-BNs and use this to inform the design of CT-BN abstraction models and assist in interpreting the meaning of discovered clusters. MSMs are exclusively used to build population-wide models of disease dynamics. Notably, their development was also motivated by the limitations of discrete-time models and the need for more expressive temporal models.

5.2.1 Continuous-time Markov Processes

When there are no natural time slices, CT models can be used to more directly reflect sequential dependencies, and avoid discretizing the time intervals. Abstraction with CT models does not have the same limiting factors of traditional methods for longitudinal data analysis, or dynamic Bayesian network approaches, such as traditional HMMs.

Rather than use the sequences of values directly, and based on recent work outlining a framework for semiparametric clustering of time series data [34], we build probabilistic models, abstractions of these sequences, using CT-BNs. This section describes their characteristic structure, the ways in which they deviate from discrete-time representations and how learning and inference is performed.

5.2.2 Discrete-time versus Continuous-time

By default, all dynamic Bayesian networks, of which traditional Markov models can be viewed as a variant, make discrete-time assumptions. The main distinguishing characteristic with the continuous-time setting is that in the discrete-time case, the Markov process stays in a state i for a time distributed according to $F(t)$ and in the continuous-time case the holding time is *exponentially* distributed according to $F(t) = e^{Qt}$ where Q is the intensity of the transitions, or the tendency to change state.

Discrete-time Markov models are characterized by a tuple, $\lambda = (\pi, A)$, where π consists of an initial state distribution that describes the transition probabilities for a set of states, $X = \{x_i, ..., x_m\}$ is a set of m states, and A is a $m \times m$ transition probability matrix that in a fully connected model describes the probability of transitioning from each state to all others and often denoted as P. In the case of hidden Markov models, the emission matrix, B, is also required to indicate the probability distribution of the observations, given A, $f(B|A)$.

In the Continuous-Time (CT) setting, the transition probability matrix, P, with entries representing values for $p_{ij} = P(X_t = j|X_{t-1} = i)$, where $p_{ij} \geq 0$ and $\sum_i p_{ij} = 1$, becomes an transition intensity matrix, Q, with entries that now correspond with an instantaneous probability of transitioning from state q_i to q_j, or:

$$p_{ij} = \lim_{\delta t \to +0} \frac{p_{ij}(t, t + \delta t)}{\delta t}$$

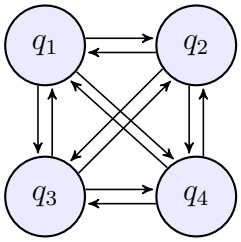

Figure 5.1: Four-state CT Markov model

Each intensity, or sojourn time in state i, has an exponential distribution (unlike a discrete-time model) with a rate given by $q_{i,i}$, the ith diagonal element of Q, where:

$$q_{i,i} = -\Sigma_{i \neq j} q_{i,j}(t)$$

Also, in a CT Markov model, the rows in the matrix Q sum to zero instead of one, with the sum of all transition intensities $q_{i,j}$ in the ith row, where $j \neq i$, equal to the absolute value of $q_{i,i}$ and the probability of observing j immediately after state i is $q_{i,j}/q_{i,i}$.

Example

A four-state CT Markov model with all allowable transitions is shown in Figure 5.1, where $i \in \{1, 2, 3, 4\}$, r is the state at time t, and s is the occupied state at the next observation. The intensity matrix, Q, represents the instantaneous behavior of the process X, and the associated set of model intensities.

For a continuous process variable X with a domain of of $x_1, x_2, ..., x_n$, where

$$Q_X = \begin{bmatrix} q_{1,1} & q_{1,2} & q_{1,3} & q_{1,4} \\ q_{2,1} & q_{2,2} & q_{2,3} & q_{2,4} \\ q_{3,1} & q_{3,2} & q_{3,3} & q_{3,4} \\ q_{4,1} & q_{4,2} & q_{4,3} & q_{4,4} \end{bmatrix}$$

Figure 5.2: Intensity matrix

n corresponds with the number of states, the intuition is that the intensity, q_i, no longer corresponds with the transition probability that is constant for the length of a time slice, but is rather the instantaneous probability of leaving state x_i and the intensity of $q_{i,j}$ gives the instantaneous probability of transitioning from x_i to x_j. In the intensity matrix shown in Figure 5.2, the expected time of transitioning is $1/q_i$ and movement to state j has the probability q_{ij}/q_i.

Embedding CT-BNs

For temporal abstraction, we describe Q in the form of two matrices, the holding matrix, H, and the transition probability matrix, P_E, for the embedded Markov model, which is used for embedded model clustering. This relationship is described by

$$Q = H(P_E - I)$$

and P_E is defined by writing out the $n \times n$ probability matrix by entering zeros along the diagonal and q_{ij}/q_i in the q_{ij} entries. The holding matrix, H. which defines the probability of staying in the same state, is created by putting q_i on the

diagonal and zeros in all other entries.

An example of how the embedded Markov chain is derived from Q is shown in Figure 5.3. It is important to note that P_E is not equivalent to the transition probability matrix in DT models. Rater, it corresponds with the probability of an instantaneous transition.

$$\begin{bmatrix} -.23 & .03 & .10 & .10 \\ .40 & -.67 & .20 & .07 \\ .01 & .01 & -.05 & .03 \\ .05 & .05 & .02 & -.12 \end{bmatrix} = \left(\begin{bmatrix} 0 & \frac{.03}{.23} & \frac{.10}{.23} & \frac{.10}{.23} \\ \frac{.40}{.67} & 0 & \frac{.20}{.67} & \frac{.07}{.67} \\ \frac{.01}{.05} & \frac{.01}{.05} & 0 & \frac{.03}{.05} \\ \frac{.05}{.12} & \frac{.05}{.12} & \frac{.02}{.12} & 0 \end{bmatrix} - I \right)$$

Figure 5.3: Q described in terms of P_E and H

Multi-state Markov Model

In this section I discuss an instance of CT-BNs used by biostatisticians and epidemiologists to build survival models for chronic and terminal diseases, the Multi-state Markov Model (MSM). To design CT models for patient level chronic disease monitoring, I incorporate aspects of MSMs, which share a similar mathematical foundation with CT-BNs. Although both are rooted in homogenous-time stochastic process theory, they have evolved separately from work in computer science.

MSMs have been widely applied for population-level disease modeling. For example, one common application is a to construct a survival model that can be used to examine the influence of covariates, and the variable risk it may present to

patients with specific values. Also, numerous variants of the basic Markov model have been developed. Including types that only allow transitions to adjacent states, and unidirectional models that progress to an absorbing, or censoring event such as death. In the HMM extension, the emission matrix is called a misclassification matrix, and there are techniques for translating a diagnostic test's sensitivity and specificity to correspond with the traditional HMM's emission matrix.

Also, variants of MSMs have been developed to capture auto-regressive time series features, integrate piecewise linear constant functions that more precisely examine risk based on the duration in a model state, and the integration of additional clinical knowledge to estimate risk in the presence of informative sampling times [33, 66]. MSM theory provides principles to govern their construction, application and interpretation. In terms of structure, nodes represent disease states that are ordered progressively to reflect stages in a disease trajectory. A patient with chronic diseases may traverse these nodes as their disease progresses, and typical MSM states correspond with 'healthy', 'diseased', and 'diseased with complications'. In the case of applications to survival analysis, transitions may be unidirectional and terminate with a final absorbing state that has no outbound transitions is used to indicate death. Also, for modeling latent states, it is common for an MSM emission matrix to reflect diagnostic misclassification error, and there are techniques to estimate initial values based on sensitivity and specificity.

5.2.3 Learning

When the parameters for a model are unknown, one way to compute the likelihood of a model is from the transition probability matrix, $P(t)$. To calculate $P(t)$ in terms of Q, I use the Kolmogorov differential equations [14]. These equations characterize the transition functions and are used to extend the forward-backward algorithm, described in detail in Section 3.4.3, to the continuous-time setting. The relationship is defined by the matrix exponential of Q scaled by the time interval

$$P(t) = exp(tQ)$$

Eigensystem decomposition is used to estimate the matrix exponential.

To determine the likelihood for the full model, $L(Q)$, the product of the transition probabilities between all observed states is calculated. The implementation I use for CT-BN parameter learning comes from the R package MSM [33]. Specifically, I use functions that can be used to calculate the matrix exponential, and provide a naive estimation of the initial model probabilities. However, to compare model likelihoods and determine the number of states that should be in the model, generate a collection of n singleton temporal models, derive the matrix P_E, and parallelize the parameter learning computations, I customized the package. Also, it is important to note that the process of parameter learning can be trivially parallelized.

For calculation, the MSM package calls R's base optimization function, `optim`. In the case of a simple model with only a few states, an analytic solution can be

calculated by setting the method parameter of the `optim` function for the Nelder-Mead method. For hidden Markov models, or models with five or more states, the BFGS algorithm [2] can be used to perform likelihood estimation. If model convergence is an issue for parameter learning, this function should be adjusted to modify the convergence criteria.

5.3 Nonparametric Bayesian Clustering

In addition to CT-BN abstraction, I extend the clustering step to the nonparametric Bayesian setting, which allows the number of clusters, k, to be expressed as a function of the size and complexity of the data. State-of-the art semiparametric temporal clustering algorithms that have been applied in recent work use spectral methods for the nonparametric clustering step. However, a major limitation of spectral methods is the need to prespecify k using a heuristic.

For clustering spectral clustering applications, how to determine the value of k for a data set is still unresolved. Typically, k is estimated using the spectral gap technique, or predictive estimates. However, both are heuristics, and do not guarantee the choice of k.

There is a more fundamental problem with the notion of a fixed k. Not only does this suggest that the categories exist independently of deeper task context, but it can force the membership of observations into meaningless clusters, and lead to spurious clusters. Also, research has shown that humans employ multiple strategies for finding k, and even on simple data sets the number of possible

interpretations can be high [38].

As a nonparametric alternative to spectral methods for clustering embedded models, nonparametric Bayesian methods are attractive in that they do not require the need to determine k in advance. Nonparametric Bayesian methods pose partitional clustering as a problem of identifying the two level prior distribution for a hierarchical Bayesian model. The nonparametric quality of the approach is achieved by defining the clustering problem as identifying the components of an infinite mixture where k is a random variable in the model. Although others can be used, the flexibility of the Dirichlet prior makes it a popular choice.

To cluster patient time series, I take the limit of the number of mixture components, k, as a hierarchical Gaussian Mixture Model (GMM) approaches infinity. More background on hierarchical Bayesian models and Drichlet process Gaussian mixture models appear in Chapter 4, Section 4.1.3.

5.3.1 Adaptation to New Problem Tasks

I have argued that the approach I have developed solves several problems with existing approaches to clustering temporal data. It has one other advantage. To keep pace with the proliferation of data, learning methods that can separate problem semantics and algorithmic components facilitate reuse. My approach not only more accurately represents CT data and provides an alternative to spectral methods, but similar to other modeling problems based on the expressive language of BNs, is flexible enough to adapt to new problem semantics with little algorithmic tuning.

Chapter 6

Application: Chronic Hepatitis

In this chapter I describe an application of my approach to the clustering of patients with chronic hepatitis. The goal is to perform temporal clustering using lab test data that are measured as part of a liver panel, which would be ordered by a clinician to check on liver function, and urinalysis results.

6.1 Grading and Staging Liver Disease

The liver is a vital organ with a wide range of functions including detoxification and amino acid synthesis. 'Hepatitis' is a Greek word with the root 'hepat' meaning liver and the suffix 'itis' indicating inflammation. It is used to describe a class of viruses that are associated with liver inflammation and is characterized by three main types, A, B and C.

Hepatitis A is associated with acute inflammation of hepatocytes ('cyte' is a

suffix meaning cell), and types B and C, chronic inflammation. Individuals with types B and C have the potential risk of developing more severe stages of the disease that may result in permanent scaring of liver tissue, a condition known as cirrhosis, hepatocarcinoma, a common type of liver cancer, and other end stage liver diseases.

An indictor of liver disease is the fibrosis of the hepatocytes. Fibrosis is the formation of excess connective tissue, known as scaring when in response to an injury. In the pathological state this results in the loss of liver function.

The liver biopsy is the gold standard for the prognosis and treatment of hepatitis B and C and provides information about the grade and stage of chronic hepatitis. It determines the health of the liver by measuring the degree of inflammation and architectural features of fibrosis.

Although it provides important clinical information to guide the care of chronic disease patients, biopsy is invasive, costly and subject to complications. Biopsy involves extracting a tissue sample of at least 2 to 3 cm in length with a 16-gauge needle that is inserted between two of the patient's ribs. The procedure has been association with complications that can be potentially life-threatening. Also, can cost hundreds of dollars, and is subject to diagnostic error [12]. For these reasons, alternatives for assessing the stage of liver fibrosis are in great demand and the lack of alternative assessment methods has been noted as a major limitation in both management and research in liver diseases.

6.2 Data Description

The hepatitis data set used for the first set of experiments consists of blood inspection and urinalysis laboratory data that was provided by the Chiba University Hospital in Japan, and was used for the ECML/PKDD-2004 and 2005 Discovery Challenges. It consists of test values for 771 patients with hepatitis type B or C, and spans the years 1982 through 2001.

The data is de-identified and the temporal granularity is one day. For each patient, available test results appear with a value and the data of the corresponding healthcare encounter, or "visit".

For extrinsic validation of clustering assignments, about 60% of the patients have corresponding fibrosis scores reported in the data set. Although these values are subject to error, they are considered the gold standard test for determining the degree of liver fibrosis.

6.2.1 Gold Standard: Liver Biopsy

Two common assessment tools for grading and staging fibrosis are the Metavir test are shown in Table 6.1 and the Histologic Activity Index (HAI) shown in Table 6.2. The Matavir score gives an indication of the amount of the degree of physical scarring, or fibrosis. HAI represents an activity level, that corresponds with hepatocyte inflammation.

Table 6.1: Metavir fibrosis scoring

Score	Assessment
0	no scarring
1	minimal scarring
2	scarring has occurred and extends outside the areas in the liver that contains blood vessels
3	bridging fibrosis is spreading and connecting to other areas that contain fibrosis
4	cirrhosis or advanced scarring of the liver

Table 6.2: Histologic Activity Index

Score	Assessment
0	no inflammation
1-4	minimal inflammation
5-8	mild inflammation
9-12	moderate inflammation
13-18	marked inflammation

Challenge Description and Task Background

There were several goals posed by the ECML/PKDD hepatitis task. The goal most relevant to my work is the development of techniques for modeling longitudinal laboratory exam data to assess liver fibrosis without biopsy information, and to better understand the temporal patterns that correspond with the results of biopsy grading and staging. Specifically:

> *Are there temporal patterns that can be detected from lab data to help distinguish patients that progress to end stage liver disease and those that do not?*

Clustering applications for this data set appear in the literature and have been published by other researchers as recently as 2012. I use previously published results to compare the results of semiparametric Bayesian clustering with state-of-the-art temporal learning systems.

This includes the work of Hirano et al. [29]. These authors describe their method that clusters patients using PLT lab results and they report results for three subsets of patients in the data set, HVB, HVC without interferon therapy and HVC patients without interferon therapy. Other work that describes multivariate methods for clustering HVC patients without interferon therapy that I use to benchmark performance includes Hirano et al. [30, 31] and Tsumoto et al. [70]. Using the PLT and ALB data and PLT, ALB and ChE data respectively, they demonstrate that medically-relevant time series features associated with the progression of liver fibrosis could be learned from clustering methods using their method, "trajectory mining".

6.2.2 Temporal Liver Disease Indicators

For each patient, the data set includes demographics, the pathological classification of the disease, and date associated measurements including biopsy, blood test and urinalysis results. For patients with hepatitis C, there is an additional indication for interferon therapy, which is used to treat the disease and can effect the values of indicators for hepatocyte inflammation.

The progression from chronic hepatitis to hepatocarcinoma via liver cirrhosis is mainly observed in HVC patients. However, the detailed mechanism of

this disease trajectory is not well-understood, and it can take decades to develop more advanced stages of the disease. Also, the majority of HVC patients are asymptomatic, and of those that show symptoms, only a fraction progress to more advanced stages of the disease.

At the time the data was collected, medical research reported platelet count values (PLT) were correlated with fibrosis score at the time of biopsy, but cross-sectional analysis provided limited predictive power. It was hypothesized that analysis of patient level temporal patterns could provide additional information for categorizing hepatitis patients by disease-related risk types. However, temporal analysis of the PLT data was rarely performed and limited by difficulties in time series comparison, irregular sampling intervals, and variable sequence lengths [29].

6.3 Methods

In this section I describe the implementation details for the experiments. I use feature-based clustering with k-means as a baseline clustering algorithm. Once features have been identified for the model, the approach is simple to implement and describe to a domain expert, and requires many less computational steps compared with model-based clustering techniques. In addition, I perform semiparametric clustering in the univariate and multivariate setting.

I assess performance by extrinsic validation, which determines the correspondence of the resulting clusters with an gold standard score, liver biopsy. In these

experiments, I evaluate the results of semiparametric clustering that pairs CT abstraction with spectral and nonparametric Bayesian clustering. Also, I compare the top results of semiparametric clustering with state-of-the-art temporal mining systems that report results for univariate and multivariate clustering for a subsets of the hepatitis population.

Although the temporal data contains the results of 983 types of examinations, I use the key indicators that are noted in the literature, and those indicators that showed diagnostic relevance for the machine learning challenges. These include Bilirubin (D-BIL), Cholinesterase Test (ChE), and Albumin Test (ALB) lab tests that have been used for temporal clustering, and Zinc Turbidity Test (ZTT) and Bilirubin (D-BIL), which were reported by challenge participants as informative for for building decision tree classifiers to predict fibrosis stage [18, 29, 71]. A detailed description of each of these five tests appear in Table A.1 of the Appendix. In the dataset was a file that provide low and high threshold values for each of the tests that are used in clinical practice.

6.3.1 Feature-based Clustering

The first type of feature, low, normal and high *transitions*, generated for clustering temporal sequences, were extracted using the observed value, and information that was provided to label each value as low, normal or high. For each patient, I calculated the number of sequential transitions of each value type. For example, the number of times a normal test registered as low, normal or high at the next time point. The counts for each transition type was divided by the total number of

transitions in the patient's record for the same test.

The second type of feature that was extracted also used the information provided by the low and high test value thresholds. For each test, I calculated the *fraction of days* measured, or the number of tests divided by the difference of the last and first testing times in the patient's record. Lastly, some *demographic* features such as age and gender were provided for clustering.

After the features were extracted, k-means was performed using the collection of feature vectors, one for each patient. This clustering algorithm is described in 4.1.1. To determine the best value of k, I used the elbow method. This is a heuristic approach that consists of plotting the mean squared error for all points from their cluster's centroid against the value of k and looking for where improvement level out, suggesting that there is little additional contribution from adding more cluster parameters.

6.3.2 Semiparametric Bayesian Clustering

The procedure for clustering multivariate temporal data is below. In contrast to a DT model, temporal sequences do not need to be in the form of uniformly spaced time series. However, raw data values may be transformed either to capture problem semantics, or to provide a continuous rather than a discrete set of values, and this process is indicated by the map function below.

6.3.3 Procedure

Definition: For a collection of patients, X, let x_i^j be one of n patients sequences for one of m variables where $i \in \{1, ..., n\}$ and $j \in \{1, ...m\}$; λ_i^j is the ith patient's model, abstraction, for the jth variable; $\Phi_i = \{\lambda_i^1, ..., \lambda_i^m\}$ is the model collection for patient X_i; and k be the number of mutually exclusive clusters for the set $C_1 \cup C_2 \cup ...C_k$ that partitions X.

Input: Patient data set X, state sequence mapping, map, to transform observations to state values

Output: cluster composition with each patient assigned into one of k groups

Algorithm 1 Multivariate Semiparametric Bayesian Clustering with Continuous-Time Abstraction

for $(1 \leq i \leq n)$ **do**
  for $(1 \leq j \leq m)$ **do**
    $x'_i \Leftarrow$ map(x_i) ▷ transform X into X'
    $\lambda_i^j \Leftarrow argmax_{\lambda_i^j} P(x'^j_i)|\lambda^j)$ ▷ **abstract** x_i's model for variable j
  end for
  $\Phi_i \Leftarrow \{\lambda_i^1, ..., \lambda_i^m\}$
end for
$\{C_1, ..., C_k\} \Leftarrow$ cluster$(\{\Phi_1, ..., \Phi_n\})$ ▷ **non-parametric clustering** step

Once a model is defined for the underlying disease process, the sequences are abstracted, then clustered, assigning every patient into one of k patient groups. Since tests do not necessarily appear on the same day, in the multivariate setting a model is learned for each variable and combined in the clustering step.

6.3.4 Continuous-time Abstraction

In this section I describe the details for clustering temporal patient-level hepatitis data. In contrast to DT methods, the approach that I have developed uses the embedded Markov chains of continuous-time Markovian models for embedded clustering. Each patient's model parameters, abstractions of their temporal data, serves as the input to a clustering method.

Model Structure

I use a 3-state continuous-time Markov model to abstract temporal information. This is shown in Figure 6.1. Ideally, expert knowledge is available to determine the number of disease states and the initial probabilities for the intensity matrix. For the hepatitis lab tests, threshold values for low and high test values were indicated. This information was directly represented as three discrete states in a probabilistic graphical model, 'low', 'normal' and 'high', and provides the mapping to transform the test values in the raw sequence to a state sequence for each patient.

Variable Selection

For multivariate clustering, the temporal data for each is abstracted separately. Although CT-BNs allow for covariances, they must be observed concurrently, and there are numerous days where the data does not appear for all three. For example, a patient may have been given both a blood and urine test during a visit, or only

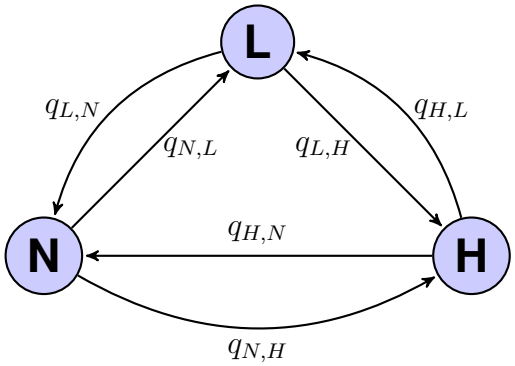

Figure 6.1: 3-state CT Markov model

one of these at a particular time. For each patient, models for each variable are learned independently, and the collection of features is used for clustering each patient.

To determine the best variable combination for multivariate clustering, *feature selection* was used to determine which of the five indicator variables could be excluded from the final model. This consisted of clustering each indicator individually and evaluating the mutual information scores between each pair of assignments. Of the five initial variables, two were eliminated, ChE and D-BIL and for the final multivariate clustering model PLT, ALB and ZZT data were used for each patient.

Model Learning

The parameters, or abstraction, used as input to clustering is derived from the intensity matrix, Q, as detailed in Section 5.2.2. The embedded representation describes the instantaneous behavior of the process X, as an $n \times n$ matrix. As

noted previously, for the multivariate setting, the parameter set would include an intensity matrix for each of the model variables.

Although, explicit information was indicated to indicate model states, the values for the initial transitions for the matrix, π, or the priors for the model at time zero are not available. To determine the initial values, I estimate an initial intensity matrix based on the population-wide frequency counts for transitions, and using a naive estimation function to calculate the matrix exponential.

Table 6.3 shows a probability of each transition type based on the total counts for all transitions for platelet test values. For example, based on population-wide transition counts, patients reporting a high platelet count show the following probabilities for their next next observation period: high 92%, normal 83% and low is less than 1%.

Table 6.3: Input for model abstraction step

$q_{i,j}$	H	N	L
H	0.915	0.083	0.001
N	0.035	0.961	0.004
L	0.025	0.224	0.751

Naive estimation using this approach was preferential to a patient-level frequency based method, and a random initialization method, both of which posed more convergence issues for learning the final patient-level models.

An example of the input used to learn the CT-BN parameters for each patient based on their state observation sequence is shown in Table 6.4. It consists of the date and state for each of the patient's lab values in chronological order, the

last column shown "PLT" indicated the patient's lab measurement, which has been mapped to a state of "2", or "normal", and is not used for the parameter estimation.

Table 6.4: Input for model abstraction step

data	state	PLT
19811111	2	177
19830720	2	182
19830818	2	167

To learn the parameters that govern each patient's model, and are used as input to the clustering algorithm, a continuous-time extension of the Baum-Welch algorithm is used to learn the intensity matrix, Q. The details of the forward-backward algorithm are given in Section 3.4.3. For calculating likelihoods, I use the BFGS quasi-Newton optimization algorithm [2] that is available in R.

6.4 Nonparametric Clustering

After raw time series sequences have been abstracted, transforming them to a succinct uniform length representation, they are clustered. The justification for using a nonparametric clustering method is that it allows the modeler to be more agnostic about specific attributes of clustering results.

To assess potential benefits if pairing nonparametric Bayesian clustering instead of spectral methods, I will compare two approaches to nonparametric clustering. The first approach, spectral clustering, uses the patient models and the number of clusters, k, as input. The second clustering technique, an implemen-

tation of nonparametric Bayesian clustering uses a two-level Bayesian hierarchy where the base measure for a data set is assumed to be a Dirichlet distribution. Unlike spectral methods, k does not need to be indicated $a\ priori$.

6.4.1 Spectral Clustering

Many varieties of spectral clustering algorithms exist and in this work I use a method first proposed by Ng et al. [44]. Variants of spectral clustering often differ in the representation of the graph Laplacian, and have been shown to enhance the clustering so that the cluster-properties in the data, so that groups in the data can be easily detected by k-means in high-dimensional space [44]. The best value for k is determined by the *eigengap*.

More background on spectral clustering can be found in Section 4.1.2. An outline of the procedure I use for the experiments is as follows. Given the n patients, each represented by a parameter set, or the set of patient models for clustering, $x_1, x_2, ..., x_n$, and k:

- Create the affinity matrix A, defined by $A_{ij} = exp(-||s_i - s_j||^2/2\sigma^2)$ where $i \neq j$ and $A_{ii} = 0$.
- From A define D as the diagonal matrix where D_{ii} is the sum of the ith row.
- Construct the Laplacian matrix $L = D^{-1/2}AD^{-1/2}$.
- Find the k largest eigenvectors of L, $\{s_1, ..., s_k\}$ and a matrix S by stacking them.

- Define a new matrix Y by renormalizing each row in X. On X, perform k-Means to obtain a clustering assignment $C = \{c_1, ..., c_k\}$.
- For each initial point x_i where $i \in \{1, ..., n\}$, if row i of the matrix Y was assigned to the cluster j where $j \in \{1, ..., k\}$.

6.4.2 Bayesian Clustering

Clustering is performed using a *Dirichlet process Guassian mixture modeling* method. For more details on this method, and the justification for its use, see Section 5.2.3.

A Dirichlet Process Gaussian Mixture Model (DPGMM) defines a DPMM by taking the limit of the number of mixture components, k, as a hierarchical Gaussian mixture model approaches *infinity*, expressing k as a function of the size and complexity of the clustering sample. Historically, DPGMMs have been used as a density estimation method, and more recently have been applied to clustering [47].

Two methods used to specify the priors for a DPGMM are Markov chain Monte-Carlo (MCMC) and variational inference, the later of which is described in the literature [7]. To compute the model likelihoods and posterior distribution of the clusters, I use an implementation of the algorithm that uses mean variational inference [49]. It is important to note that for variational inference, the upper bound for the number of clusters, and α must be prespecified. Only the later is true for MCMC.

To report alternate 'views' of clustering assignments for the sample, I use

different values of α, the scaling factor for the model, with the upper bound for the maximum number of clusters set to ten. I modified this package to generate multiple runs for a given α, and for runs with the same k value, to return only the assignment with the lowest score based on calculating the BIC.

Validation

Using the B-cubed metric described in Section 7.3, I validate the results of clustering. In contrast to some popular clustering metrics such as purity, the B-cubed metric satisfies important formal constraints for a good clustering metric, and its interpretation is more intuitive for clinicians and health services researchers due to the commonalities it shares with sensitivity and specificity, important pairwise counting statistics for determining the quality of a diagnostic test and discussed in Section 4.2.1.

6.5 Results

In this section I present the results of clustering temporal lab data for chronic hepatitis patients. I conducted experiments on a population of patients with Hepatitis Virus B (HVB) and Hepatitis Virus C (HVC) and compare them with the results of three temporal clustering systems that have been described in the literature. Cluster quality is assessed with extrinsic validation to determine how well the assignments produced by different methods correspond with clinically significant groups.

It is important to note that most previous work on clustering hepatitis patients focuses on only those patients with HVC and no interferon therapy. Until the recent introduction of interferon therapy, HVC was considered an nontreatable disease. In terms of fibrosis indicators, it can cause fluctuations in physiological signals that would not be observed otherwise. Therefore, is can be helpful to distinguish the population of patients without interferon treatment.

In this section, I first report the extrinsic validation results of semiparametric clustering for the sample using a three-state continuous-time Markov model. The second set of results compares two alternative nonparametric clustering methods, spectral and Bayesian, that can be paired with model-based temporal abstraction techniques. Lastly, to access the relative performance of semiparametric clustering with state-of the-art systems, I compare results with published work on univariate and multivariate temporal clustering on the hepatitis data set.

6.5.1 Feature-based Clustering

Three lab tests, PLT, ALP, and ZTT, were selected for multivariate clustering. Using various combinations of variables I assessed clustering assignments. Based on the set of features that were extracted, and that I describe in full detail in Section 6.3.4, only the transition feature generated for PLT lab data, and the fraction of days measurements for ZTT and ALB data were useful for clustering.

The B-cubed scores for the baseline clustering appear in Table 6.5 and are compared with the top results of semiparametric Bayesian clustering, which is discussed in the next section. My results show a substantial benefit over the baseline

k-means algorithm, suggesting that a more complex temporal clustering approach is warranted.

Table 6.5: Precision, recall, and B-cubed scores for baseline and top scores for semiparametric Bayesian clustering.

method	sample	k	P	R	B-Cubed
Bayesian clustering	HVC no Tx	5	0.,48	0.58	0.52
k-Means	HVC no Tx	4	0.45	0.47	0.46
Bayesian clustering	all	4	0.35	0.62	0.45
k-Means	all	5	0.44	0.33	0.38

6.5.2 Semiparametric Bayesian Clustering

To assess the performance of semiparametric clustering using gold standard results, Figure 6.2 shows results for all patients, using temporal PLT, ALB and ZZT data for all patients and the subset of HVC patients with no interferon therapy indication. The B-cubed precision is on the x-axis and the B-cubed recall is on the y-axis. The B-cubed scores range from dark to light shades of blue, with lighter values indicating higher scores. Different marker shapes indicate the value of k, which is noted in the legend.

Table 6.6 and Table 6.7 report the B-cubed scores shown in 6.2. For clustering all patients, the top results show a 0.45 B-cubed value for $k = 4$, and for the HVC patients with no interferon therapy, a 0.52 B-cubed value for $k = 5$.

For the HVC patients with no indication of interferon therapy, the B-cubed scores that correspond with k=1 through 9, appears in Table 6.7. Cluster values

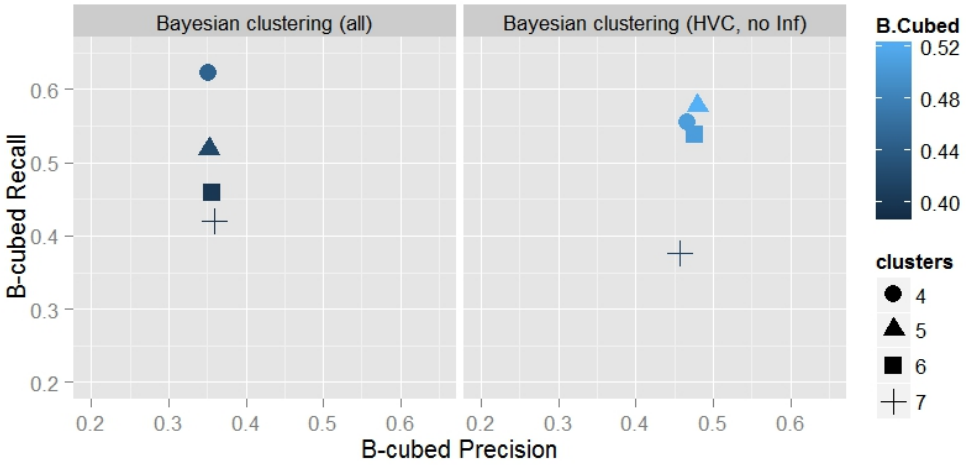

Figure 6.2: Comparison of semiparametric Bayesian clustering by patient population

Table 6.6: B-cubed scores for semiparametric Bayesian clustering (all patients)

k value	P	R	B-Cubed
4	0.35	0.62	0.45
5	0.35	0.52	0.42
6	0.36	0.46	0.40
7	0.36	0.42	0.39

less than three were generated by setting the upper-bound to the numbers of clusters I was looking to achieve, and did not result as 'naturally' as higher values, all of which were generated using an upper-bound of ten and decreasing α, the scaling factor.

Table 6.7: Semiparametric Bayesian clustering: HVC patients without an indication of interferon therapy: precision, recall, and B-cubed scores for k values 1 through 9.

k value	P	R	B-Cubed
1	0.36	1	0.53
2	0.62	0.42	0.50
3	0.48	0.49	0.481
4	0.47	0.55	0.51
5	**0.48**	**0.58**	**0.52**
6	0.48	0.54	0.51
7	0.46	0.38	0.41
8	0.49	0.43	0.46
9	0.50	0.29	0.37

Also, it is important to note that the high value at k=1 does not suggest a strong clustering, rather it shows a limitation of the B-cubed metric. The purpose of reporting it here is to indicate the B-cubed bias as the value of k gets very small. When k=1 all classes are maximal and the B-cubed recall is always equal to the highest achievable value, one. The limitation of the B-cubed score is more problematic when the number of classes is very small, and the probability of a random guess being a correct class assignment increases.

6.5.3 Bayesian and Spectral Methods

In this section I discuss the results of the alternative nonparametric spectral and Bayesian methods reported in the previous section. For both clustering methods, I evaluate the results of multivariate clustering using PLT, ALB and ZZT data against the gold-standard, liver biopsy.

The B-cubed values for different experimental settings are reported in Table 6.8, which shows that the best results are achieved with Dirichlet process Gaussian mixture modeling, the nonparametric Bayesian clustering method. As noted in the previous section, the highest score is reported for a run that assigned patients into one of five clusters.

Surprisingly, spectral clustering shows a lower relative performance than both feature-based clustering and semiparametric Bayesian clustering. The spectral gap heuristic indicated the best value of k=6, and an additional candidate value of k=7 followed in rank. To determine if the poor relative performance was a result of higher k values, I applied spectral clustering with the same value of k as the baseline clustering method, k=5. Although performance gains are observed, spectral clustering still shows the lowest overall score.

Figure 6.3 shows the results of CT-BN abstraction paired with each of the clustering methods that appear in Table 6.8. B-cubed precision and recall appear on the x and y-axis respectively. The B-cubed composite value is indicated in blue, with the higher scores on the lighter end of the gradient, and marked with a symbol that corresponds with the number of clusters indicated by the legend.

Using the same continuous-time temporal abstraction step, I show that semi-

Table 6.8: B-cubed value for baseline k-means, spectral and nonparametric Bayesian clustering, all hepatitis patients

method	k	P	R	B-Cubed
k-means	5	0.44	0.33	0.38
spectral methods	4	0.37	0.30	0.33
spectral methods	6	0.38	0.22	0.28
spectral methods	7	0.38	0.21	0.27
Bayesian clustering	4	0.47	0.55	0.51
Bayesian clustering	**5**	**0.48**	**0.58**	**0.52**
Bayesian clustering	6	0.48	0.54	0.51

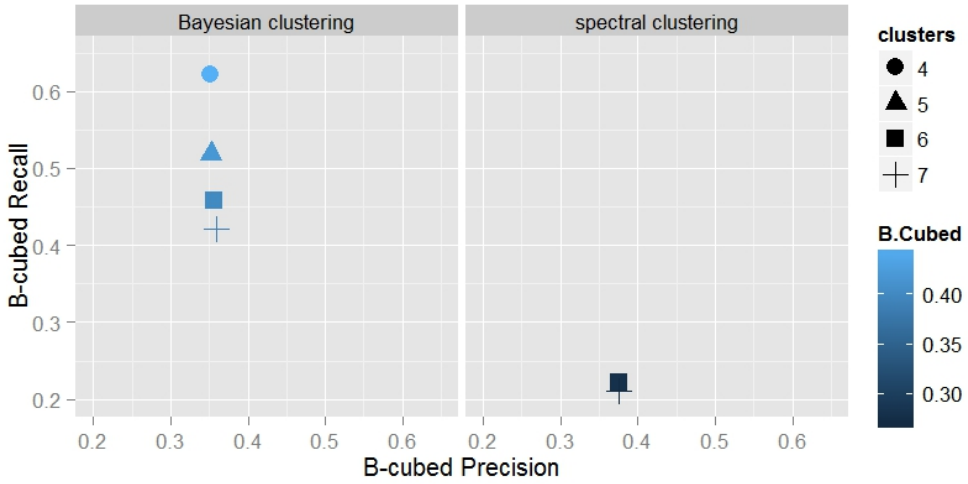

Figure 6.3: B-cubed value for different nonparametric clustering methods and k values for the hepatitis data set displayed by gradient.

parametric Bayesian clustering consistently performs better, overall, and when comparing performance for the same values of k. The absence of a heuristic step to determine k, and a relaxation of the assumption that a data set has a fixed k, are clear benefits. Also, in contrast to spectral methods and k-means, which make the assumption that there is one true k for a data set, it allows for multiple clustering views. However, nonparametric clustering is not fully automatic and still requires the tuning of the scaling factor.

Spectral methods are currently the state-of-the-art for nonparametric clustering techniques. These results indicate not only the appropriateness of nonparametric clustering, but also potential advantages over the state-of-the-art. Bayesian clustering methods offer more flexibility by determining k as a function of the size and the complexity of the data, and are not theoretically opposed to multiple views of the same data set.

One advantage that may be attributed to the clustering approach is the Gaussian assumption that DPGMMs makes in terms of component densities. Spectral methods attempt to balance the size of the clusters while minimizing the interaction between dissimilar points, and can bias results towards clusters of equal size [74]. Many phenotypes are normally distributed among the population, and our results suggest that spectral clustering may not be the best choice for modeling patient populations.

6.5.4 Univariate Systems

Hirano et al. [29] apply a subsequence abstraction method to platelet (PLT) time series for clustering raw time series sequences and I use this as a benchmark for which to compare my results. Using the 3-state Markov model trained *only* on a patient's platelet test data, Table 6.9 shows the results of semiparametric Bayesian clustering for the subset of patients with HVC and no indication of interferon therapy (no Tx), compared with the benchmark results for various population subsets including HVC no Tx, HVC with interferon treatment (Tx), and HVB patients. For the benchmark results, the B-cubed values were calculated using the cluster constitutions that were published in Hirano et al., with the authors indicating that small clusters of $n < 3$ omitted.

Table 6.9: Precision, recall, and B-cubed scores for alternative systems using only temporal PLT data

method	sample	k	P	R	B-Cubed
Bayesian clustering	HVC no Tx	**5**	**0.49**	**0.41**	**0.45**
Bayesian clustering	HVC no Tx	**6**	**0.48**	**0.42**	**0.45**
Bayesian clustering	all	4	0.36	0.39	0.37
Bayesian clustering	all	5	0.33	0.39	0.36
Bayesian clustering	all	8	0.37	0.35	0.36
Hirano 2005	HVC no Tx	6	0.46	0.39	0.42
Hirano 2005	HVC Tx	11	0.39	0.23	0.29
Hirano 2005	HVB	8	0.26	0.28	0.31

Despite the omission of smaller clusters that biased the score in the benchmark's favor, the results of semiparametric Bayesian still out perform the alter-

native(0.45 to 0.42) for the key comparison group, patients with HVC and no interferon treatment. Figure 6.4 shows a visual comparison of the top results for semiparametric clustering and the results for the three patient subsets reported by the benchmark system for univariate temporal clustering (0.42 HVC no Tx, 0.29 HVC Tx, 0.31 HVB).

Since it is impossible to estimate a precise B-cubed score precisely with an unknown number of clusters omitted, I do not make additional comparisons for subsets of patients with HVB or HVC and interferon therapy to avoid more approximations. However, the top score for the population wide performance using PLT tests only is reported at 0.37 for semiparametric clustering, and for multivariate clustering, Table 6.8 shows a population wide performance of 0.45, which is higher than the top results of the benchmark system.

6.5.5 Multivariate Systems

To compare semiparametric Bayesian clustering with state-of-the-art multivariate temporal clustering systems, we calculate the B-cubed scores for the subset of patients with HVC and no indication in interferon therapy in Table 6.10. Figure 6.5 visualizes the scores, and compares the two multivariate systems by Hirano et al. (PLT-ALB) [30, 31] and Tsumoto et al. (PLT-ALB-ChE) [70] and the one univariate benchmark Hirano et al. (PLT) [29] with semiparametric Bayesian clustering (PLT-ALB-ZZT). A description of these tests appear in Table A.1 of the Appendix.

It is important to note that some of the validation scores are generous. In the work of Hirano et al. (2007) [30, 31] clusters where $N < 2$ for the temporal

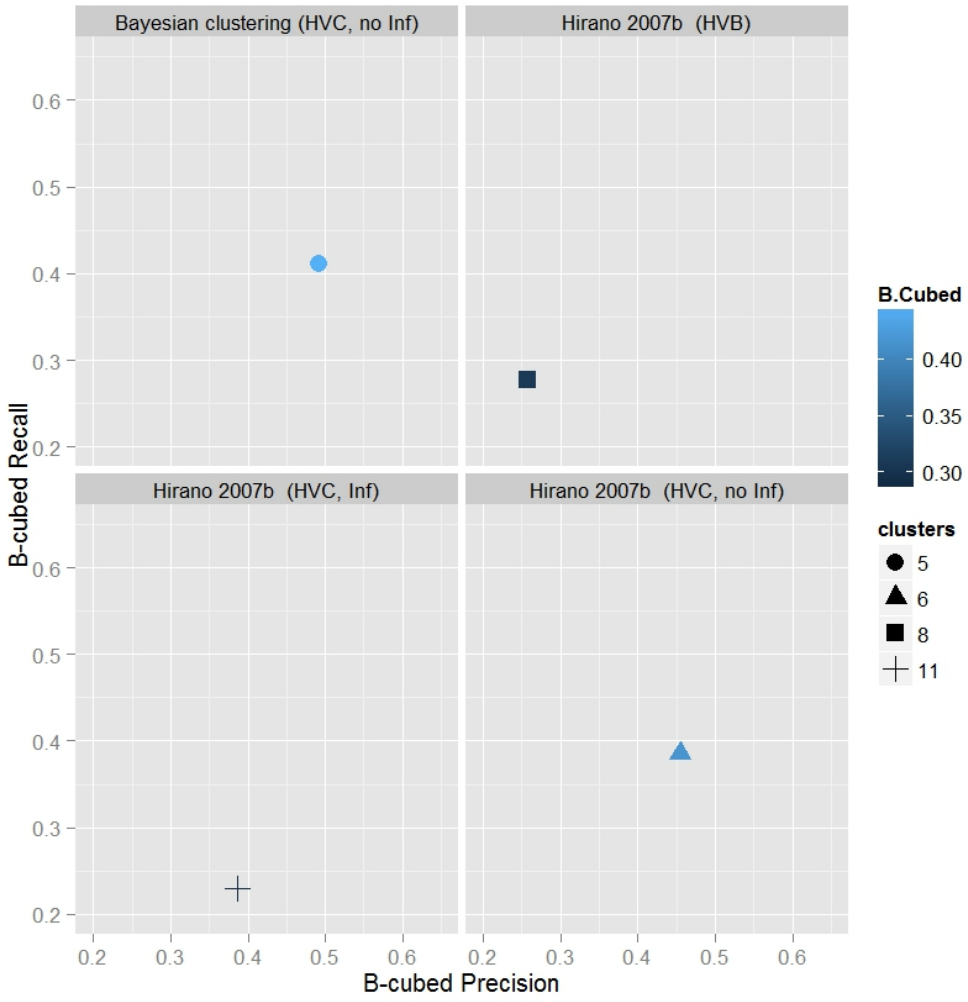

Figure 6.4: Comparison of semiparametric Bayesian clustering with previously published results using temporal PLT data

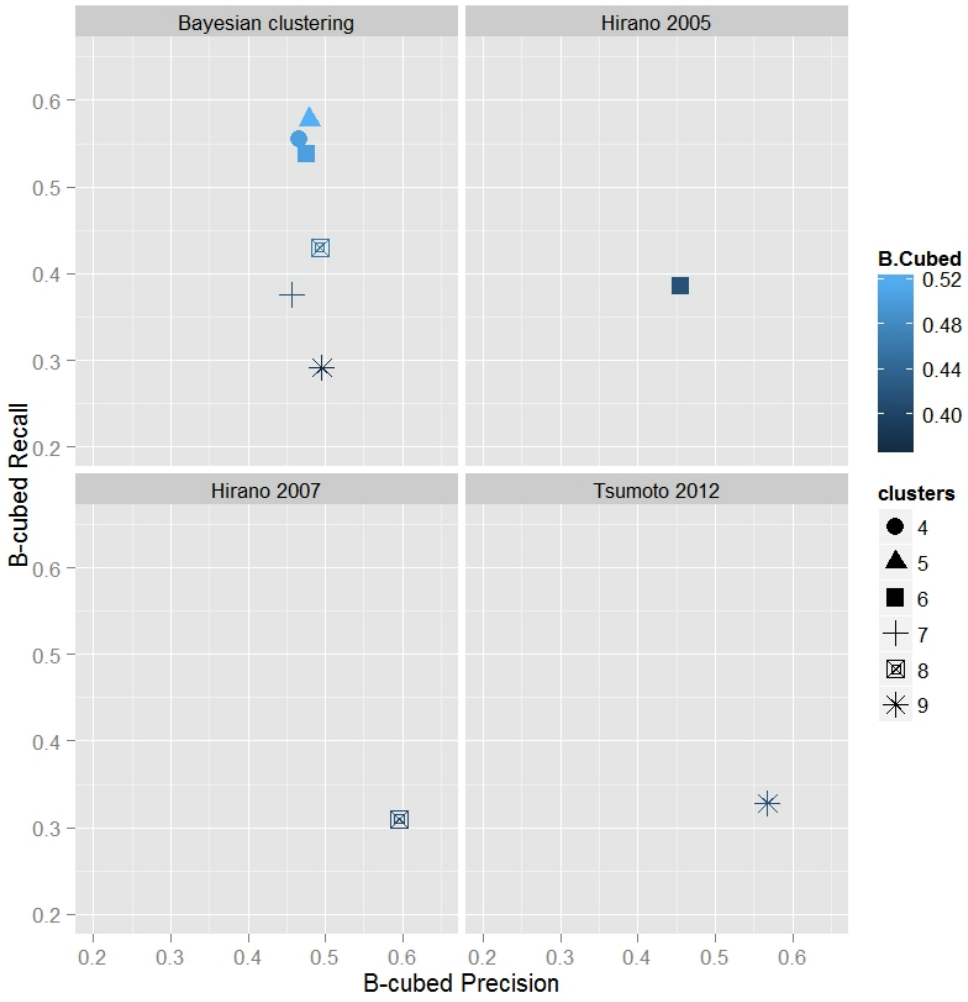

Figure 6.5: Comparison of semiparametric clustering with various benchmarks for HVC patients with no indication of interferon therapy

Table 6.10: Validation scores for semiparametric Bayesian clustering and previously published results for HVC patients with no indication of interferon therapy

method	k	P	R	b-cubed
Hirano 2007	8	0.60	0.31	0.41
Tsumoto 2012	**9**	**0.57**	**0.33**	**0.42**
Bayesian clustering	**5**	**0.48**	**0.58**	**0.52**

ALB-PLT data results were not provided in the membership table, or could be determined by another reported statistic. Inclusion of even one of the missing clusters would have reduced the completeness constraint, and lowered the value of the validation metric. In the work of Tsumoto [70] there are no indications of selective reporting, and these estimates are more precise.

Overall, the results show that semiparametric Bayesian clustering improves on existing systems. For multivariate temporal data, clustering performance increases over a 20% relative improvement using combined ALB-PLT-ZTT tab results, reporting a 0.51 b-cubed score. Cluster constitutions reported in the literature based on trajectory mining, reported the highest score at 0.42 and were based on the temporal modeling of features from combined ALB-PLT-ChE lab data. Although the ALB-PLT model reported a B-cubed score of 0.42, they omitted some clusters, and the actual score is lower.

6.5.6 Clinical Relevance

The goal of the temporal mining task was to discover patient groups that are useful or meaningful to:

- discriminate between those that progress to more advances stages such as cirrhosis or hepatocarcinoma, and
- determine if less invasive diagnostic procedures can replace liver biopsy.

For each patient, the models that serve as input to clustering characterize their disease-state dynamics, or the rate of change from of the disease states to all others. Cluster level models can be useful for describing the dynamics of clinically significant groups, and we show a visualization method that helps to describe group level differences, and can help to interpret temporal characteristics of each group to clinicians. We demonstrate this for the models learned for platelet data. Medical research has reported platelet count values (PLT) were correlated with fibrosis score at the time of biopsy, but cross-sectional studies have provided limited clinical insights.

Risk of Advanced Liver Disease

The cluster compositions appears in Table 6.11. The columns represent the fibrosis stage, the gold standard class used for external validation, and is labeled F0 through F4, with F4 indicating end-stage liver disease and corresponding with the Matavir scores described in Section 6.2.1.

Table 6.11: Cluster composition by fibrosis stage

Cluster	$F0, F1$	$F2$	$F3$	$F4$
c_1	5	2	-	-
c_2	17	11	6	10
c_3	6	1	3	6
c_4	25	1	-	1

To *stratify discovered groups by risk types*, I examine the proportion of class labels within each group. Figure 6.6 shows the cluster composition as a percent of total records for each class. The size of each circle is proportional to the total count for each Matavir score, or class label. Also, it shows the biopsy activity, Hepatitis Activity Index (HAI), with the darkest color indicating the highest activity, and which some consider a better indicator for liver fibrosis than the Matavir score.

Aided by Figure 6.6, shows each cluster as a column, we can broadly rank clusters by increased risk of end-stage liver disease based on the proportion of high and low risk members as $c_4 < c_1 < c_2 < c_3$, with c_3 at the highest risk for advanced stages of fibrosis, and c_1 at the lowest risk. In c_1, which shows the highest proportion of patients in the class F0, and a lower proportion of the total patients in each class as the as biopsy grades increase, shows a 93% positive predictive value for patient with little or no fibrosis. Consistent with this conclusion, relative to other clusters, c_1 patients also have lower activity scores.

Similar to c_1, c_4 also shows the presence of patients with lower biopsy grades, and low activity. In this cluster there are no patients with F3 or F4 scores. However, in contrast to c_1, it does not show the a larger proportion of these patients at

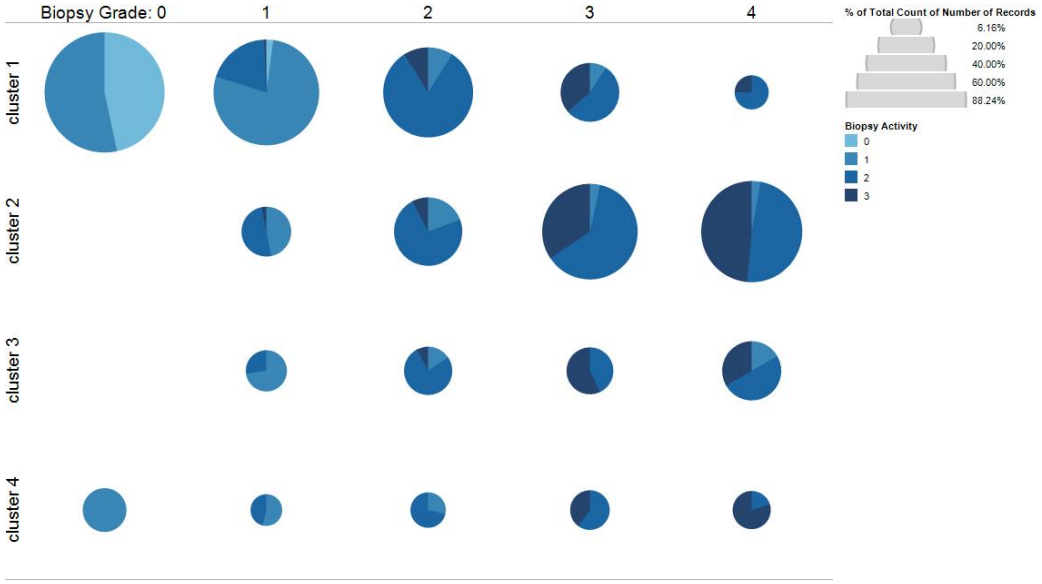

Figure 6.6: Metavir biopsy grade and HAI by cluster as a percent of total records for each class

lower biopsy grades, and reports a 73% accuracy for detection of little or no fibrosis. This suggests that members of c_4 progress to end-stage liver disease more often that those patients in c_1, but based on cluster composition, less often that patients in c_2, or c_3.

All but one patient with F3 or F4 biopsy scores appear in c_2 or c_3. Patients in c_2 are the most likely to progress to higher fibrosis stages, showing the highest proportion of patients with F3 and F4 scores, suggesting c_2 patients are a higher risk for progressing to advanced stages of fibrosis that patients in c_3.

$$Q_1 = \begin{pmatrix} 0 & 4.65 & -9.87 \\ -41.08 & 0 & -18.21 \\ -9.06 & 11.27 & 0 \end{pmatrix} Q_2 = \begin{pmatrix} 0 & -4.83 & -19.86 \\ -6.28 & 0 & -16.77 \\ -8.74 & -1.76 & 0 \end{pmatrix}$$

$$Q_3 = \begin{pmatrix} 0 & -34.75 & -23.15 \\ 5.14 & 0 & -14.16 \\ 1.60 & -5.36 & 0 \end{pmatrix} Q_4 = \begin{pmatrix} 0 & 15.12 & 8.95 \\ -59.67 & 0 & -27.21 \\ -8.36 & 17.72 & 0 \end{pmatrix}$$

Figure 6.7: Intensity Matrices for 3-state, 4 cluster hepatitis model, B-cubed=.51

Platelet Test Q Matrix Plots

As noted earlier, the entries of the intensity matrix, Q, in a continuous-time Markov model describe the disease state dynamics. The baseline intensity matrix is a transformation of Q where for each state r, q_{rr} is equal to zero. For each of the four clusters discovered, Figure 6.7 shows the mean baseline Q matrix entries for the three state (low, normal, high) chronic disease model learned from the PLT lab test values. Similar to the the untransformed Q matrix, the higher the value of q_{rs} the higher the risk of transitioning from state r to state s as $\Delta t \to 0$.

To make comparisons among clusters and for each matrix entry in Q, I calculated a standard score that no longer indicates the transition intensity, but how many standard deviations the cluster average is above or below the population mean, or the z score. Using the z normalized scores, and to facilitate communicating temporal clustering results to users that may not be familiar with the the technical details of the learning methods, Figure 6.8 provides a visual representation of the samples's intensity matrix matrices by cluster. Each of the nine $q(i, j)$ entries, now plots in the 3x3 grid that correspond with the entries of the Q matrix,

represent the possible state transitions from low, normal and high disease states. Individual clusters are indicated by color and number along the y-axis, and the x-axis represents the z-score.

Based on our assessment of risk types in Section 6.5.6, which broadly qualified the clusters by increased risk of end-stage liver disease as $c_4 < c_1 < c_2 < c_3$, and Figure 6.8, we can describe properties of patient subpopulations in terms of instantaneous risk, relative to other patient groups. One important state is $q_{norm,low}$, or the transition from a normal to low disease state based given that a patient is currently showing a normal PLT value. This plot can help to answer the question, *if a patient is currently in a normal level, how likely are they to progress to a lower (unhealthy) disease state?*

Figure 6.9 is an enlarged version of the $q_{norm,low}$ that appears in Figure 6.8. It indicates that patients in c_1 are the least likely to transition from a normal state to a more serious low state, with c_4 also showing that relative to the mean, they are both low risk and more likely to remain in the normal state. However, c_2 and c_3, the clusters associated highest fibrosis risk, are more likely to transition to poorer health based on the population average, with c_3 the most likely to transition. Notably, this interpretation is consistent with our designation for risk strata, and their ranking from lowest to highest risk.

Actionable Findings

One important finding that is not adequately captured by standard evaluation metrics, and is relevant to our problem is related to the increased importance of the

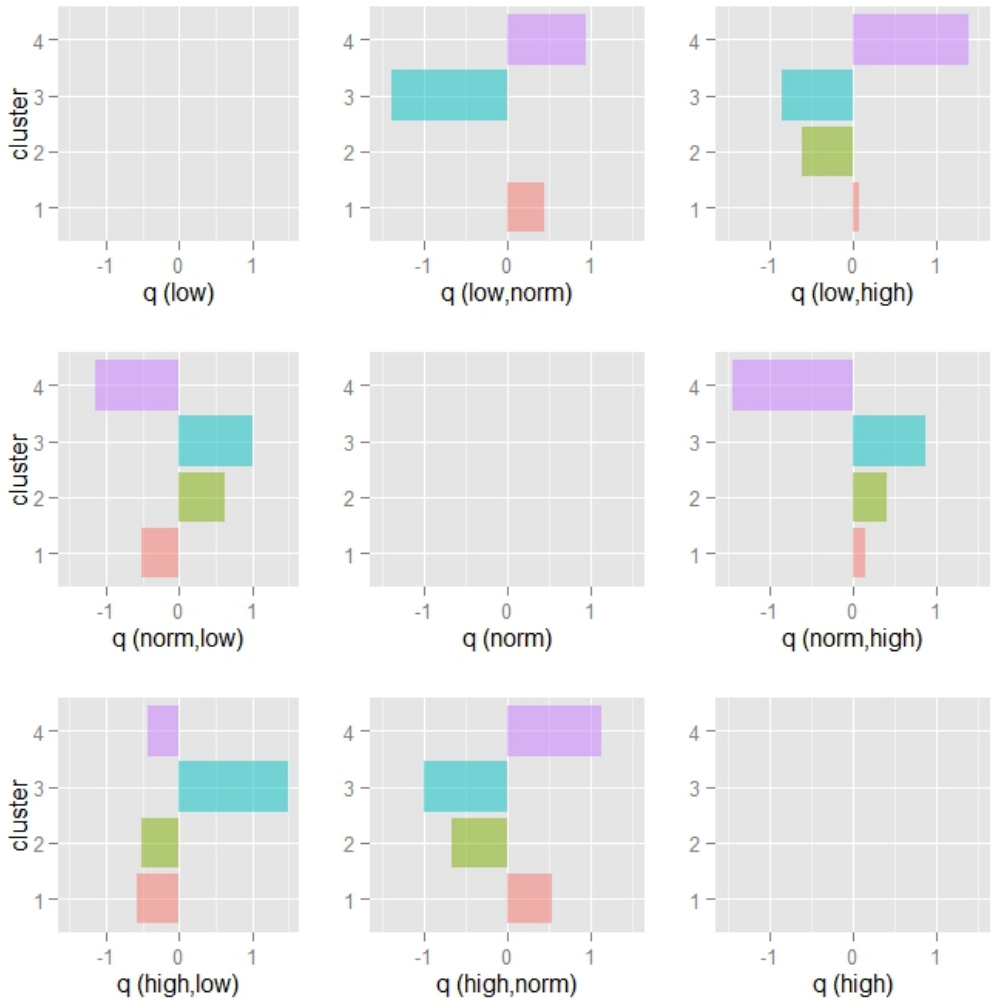

Figure 6.8: Q matrix plot for PLT clusters

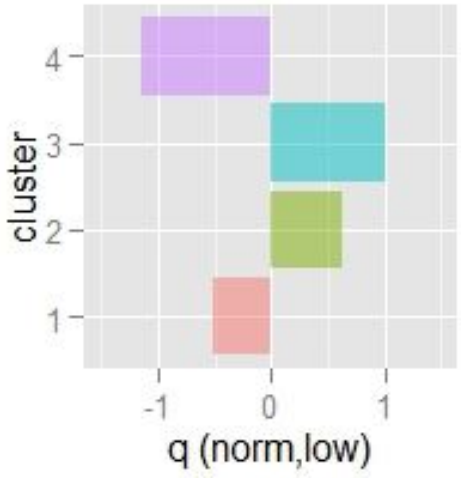

Figure 6.9: Instantaneous risk for transitioning a more unhealthy disease state

lowest fibrosis class, composed of patients labeled F0 or F1 in the gold standard. Specifically, liver biopsy is an invasive procedure that can put patients at risk of procedural complications, and costs hundreds of dollars. The ability to reduce the number of unnecessary biopsies is currently an active area of research.

The lowest risk cluster generated by our procedure consistently produced one cluster of high precision, or positive predictive value for F0 and F1 patients. Although this cluster was not maximal for this population, it can be used to identify patients for which biopsy is likely unnecessary. For example, 93% of patients in k_4 (see Table 6.11) had unnecessary biopsies, representing approximately one quarter of the patient sample. Also, combining c_2 and c_3, the clusters associated highest fibrosis risk, would provide a 96% sensitivity (recall) for patients at high

risk of end stage liver diseases. However, specificity is low.

Chapter 7

Application: Diabetes

Using a noisy, high-level signal, which consists of physician-ordered glucose tests for hospitalized patients, the goal of this clustering experiment is to determine if documentation patterns about testing can be used to stratify patients in terms of their risk of glycemic complications.

I evaluate the results of clustering for several different settings and report intrinsic validation scores. I compare semiparametric clustering model-based methods with with a baseline feature-based clustering method, and DT and CT HMM approaches for temporal abstraction. Also, I compare the results of spectral and nonparametric Bayesian clustering of the embedded CT models, and further examine cluster constitutions to describe the clinical significance of results.

7.1 Blood Glucose Management

National estimates report a 8.3% prevalence of diabetes in the Unites States, or about 25 million people, with over seven million undiagnosed [21]. Diabetes, which is diagnosed by an indication of hyperglycemia on two or more consecutive days, is a group of diseases that are characterized by high blood sugar, or hyperglycemic disease states.

In the 2010 the CDC reported that 11.5% of hospitalizations in the US indicated diabetes as the main diagnosis[1]. Glycemic events can result in various health complications such as kidney failure and blindness. However, medical research shows that behavioral changes and other interventions can prevent or delay diabetes onset showing the importance of early diagnosis and treatment.

Data mining and machine learning methods applied to clinical data have been the focus of a national challenge task, and various research practice partnerships[2], suggesting an important role for probabilistic learning algorithms in reducing the time to diabetes diagnosis, and the improvement of other outcomes.

7.1.1 High-level Signals for Glycemic Complications

There are two motivations for the use of test indication instead of the lab test results. First, some types of providers that haven't traditionally been involved in diabetes prevention initiatives, such as health management organizations, may not have access to lab test results but can easily access claims data. The second is

[1] http://www.cdc.gov/diabetes/statistics/hosp/adulttable1.htm
[2] http://www.kaggle.com/c/pf2012-diabetes

related to the sensitivity of glucose test results, which can vary by time of day, and other environmental factors. For this reason, a diagnosis of diabetes is typically made by two consecutive blood fasting tests, and not based on the results of one test.

A inpatient blood glucose test is performed when a patient shows signs of diabetes, or to monitor patients with diabetes. However, it can also be ordered as part of comprehensive metabolic panel that not only measures blood sugar, but kidney, liver and other functions, and can be part of a regular check-up.

Due to the prevalence of diabetes, its association with glycemic events that result in hospital admission, and its presence in a patient's record, glucose testing indications are common for hospitalized patients. Also, blood sugar testing is part of the metabolic panel that may be ordered to assess kidney and other organ health. Physician-ordered tests do not indicate diabetes, or even prediabetes. Glucose tests can also be ordered for patients that fit an at risk profile based on age, weight or other factors. Therefore, an isolated glucose test with out any follow up on the succeeding day may correspond with a short admission where a patient is discharged, or with a longer stay that does not require ongoing glucose monitoring.

For these reasons, it is not the indication of glucose testing with singular events that is informative for my specific clustering task, but rather a series of contiguous tests that indicate an increased risk of glycemic complications. For a hospitalized patient, this is a indication that that blood-glucose levels are being actively monitored by the attending physician, and are potentially diabetic. The longer the

signal persists, and the frequency at which episodes, indicated by a series of contiguous measurements, occur in a patient's data, the more likely it is that a patient has a serious blood glucose management condition related to diabetic complications.

7.2 Data Description

The anonymized glucose test data is for a population of patients admitted to New York Presbyterian Hospital with more than one physician ordered glucose test indicated in their EHR and a length of record (lor) greater than seven days. Similar to the hepatitis patient data, the glucose data set presents methodological challenges in that it is irregular in length, and subject to arbitrary sampling schemes.

An additional complicating factor is presented by missing data. Unlike other developed countries, the US health care system lacks centralization, and patient health data is more likely fragmented among providers, and hence more likely to be missing service utilization information. Also, observations, the correlated indicator variable here, have a weaker relationship to the underlying disease state. For this reason, and the availability of data, I use a latent model representation, or a hidden assumption for CT-BN abstraction.

7.2.1 Patient Time Series

Patient records were used to extract the temporal observations, which were then transformed into a time series. For each patient in the collection, the sequence be-

gins with the first physician ordered test on record, indicated by the presence '1' of a physician's order at t_1. Each successive day, $t_2, t_3, ..., t_T$, for a total T days, a daily value indicating the presence '1' or absence '0' composes the patent's temporal measurement sequence. At time T, the measurement sequence is terminated by a censoring state (e.g. death) or on the day the record was extracted from the hospital information system.

7.2.2 Aggregate Time Series Statistics

Figure 7.1 shows aggregate time series statistics for the entire patient population (N=242,335) in the glucose data set with the red line indicating population means. Although less informative than model-based abstraction, I have demonstrated the ability of these aggregate time series statistics to measure informative characteristics that are useful for feature-based k-means clustering in preliminary work for this thesis [67]. Here, I use them to describe the data for the purpose of selecting the study sample, provide features for a baseline clustering method, and assess clustering results in conjunction with intrinsic validation metrics such as cluster silhouettes.

Their description and calculation for a hypothetical patient,

$$x_i = \{1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1\}$$

for a total of T days, is as follows: record length (Eq. 7.1), the entropy of the observation sequence (Eq. 7.2), total "visits" or testing days (Eq. 7.3), the fraction

of total days measured (Eq. 7.4), and the length of the longest gap in each patient's observation sequence (Eq. 7.5). Also, in Figure 7.1 those patients with a sequence of all ones (e.g. $x_i = \{1, 1, 1, 1, 1, 1, 1\}$ having the lowest entropy are assigned a -0.5 value to better distinguish them from all other patients.

$$\text{lor}(x_i) = \text{len}(x_i) = n \tag{7.1}$$

$$\text{hmeas}(x_i) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{7.2}$$

$$\text{visit}(x_i) = \sum_{i=1}^{n} \tag{7.3}$$

$$\text{fdays}(visits, lor) = \frac{visits}{lor} \tag{7.4}$$

$$\text{lgap}(x_i') = \text{max}_{x_i'} \tag{7.5}$$

In terms of the population, these charts illustrate the range and distribution of glucose testing patterns, and can be used to roughly assess the degree of record incompleteness. To adjust for the length of record, we plot the fraction of days in Figure 7.1, which still shows that on average, patients have relatively few visits over the duration of their record. The entropy measure helps to provide more insight into the data generating process. The distribution for observation entropy, suggests that the underlying mechanism cannot be described by chance, and that

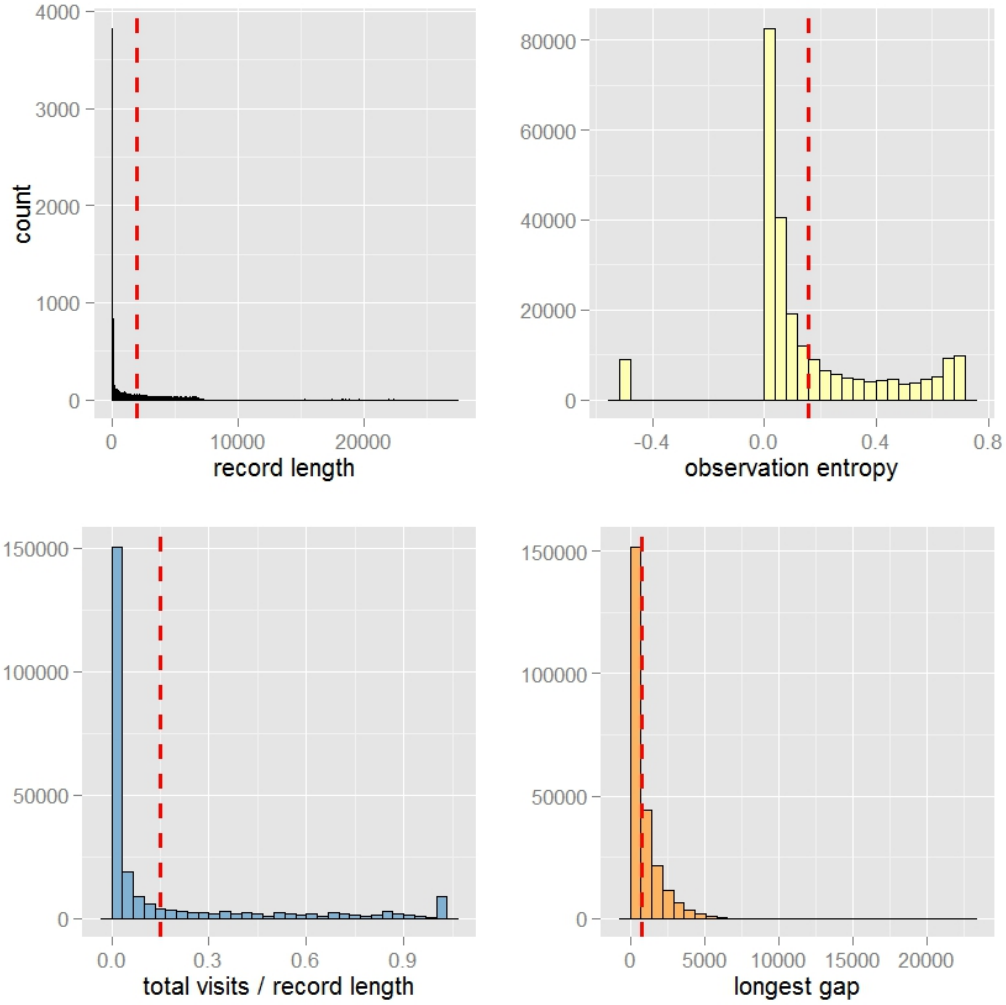

Figure 7.1: Descriptive statistics for all patients in the glucose data set

the correlations between contiguous values of the indicator variable, at least for some patients, are informative.

7.3 Methods

In this section I describe the implementation details for my experiments using the blood glucose data. A feature-based clustering that uses aggregate time series statistics as input to k-means is preformed to serve as a baseline clustering algorithm. For the semiparametric, embedded model clustering experiments I pair DT-HMMs and CT-BN abstraction methods with two nonparametric clustering alternatives that include spectral and nonparamteric Bayesian methods. Clustering assignments are evaluated using the b-cubed metric (see Section) and interpreted for any clinical relevance.

7.3.1 Selection Criteria

Although the methods can be applied to larger samples, in order to compare results visually for development and qualitative cluster assessment, I chose to select a subset of patients with a variable, but comparable record length. The goal was to identify a date range that was long enough to detect any long-term trends associated with patients that have glycemic complications but short enough to process in a reasonable amount of time.

For this work, the inclusion criteria was set to patients with a time series length in the range of 1000 to 1025 days, which I refer to as the study sample, and

provides just over 3 years of daily measurement data. To assess the ability of my method to generalize to other subsets of the glucose data, I also show results on slightly shorter, and longer time series sequences that are approximately the same sample size and consist of patients with record lengths in the 975-999 and 1026-1050 range.

Figure 7.2 shows the whole population with a study subset approximation that is indicated by a red line at the 1000 day mark on the x-axis.

The visit count distribution for the subset is comparable to the population at large, and observation entropy, and fraction of days values for the subset are outside of what appears to be a short stay bias. In addition, it excludes what appear to be outliers with very long stays. This suggests that a set of patients with the 1000 range length or record (lor) criteria is appropriate to capture longer term chronic disease trends, and avoid some of the computational burden associated with using longer time series for the development of new temporal clustering methods.

Description of Study Sample

Application of the selection criteria resulted in a total of 1024 patient measurement sequences in the range of 1000-1025 days. Their aggregate time series statistics are shown in Figure 7.3. The figure shows a similar, but smoother distribution in terms of observation and sequence entropy, a more uniform, but still variable record length distribution, and suggests a bimodal distribution in terms of longest gap length, and no longer a power law distribution. Similar to the larger population, of which the study sample is a subset, record sparsity is evident and shown

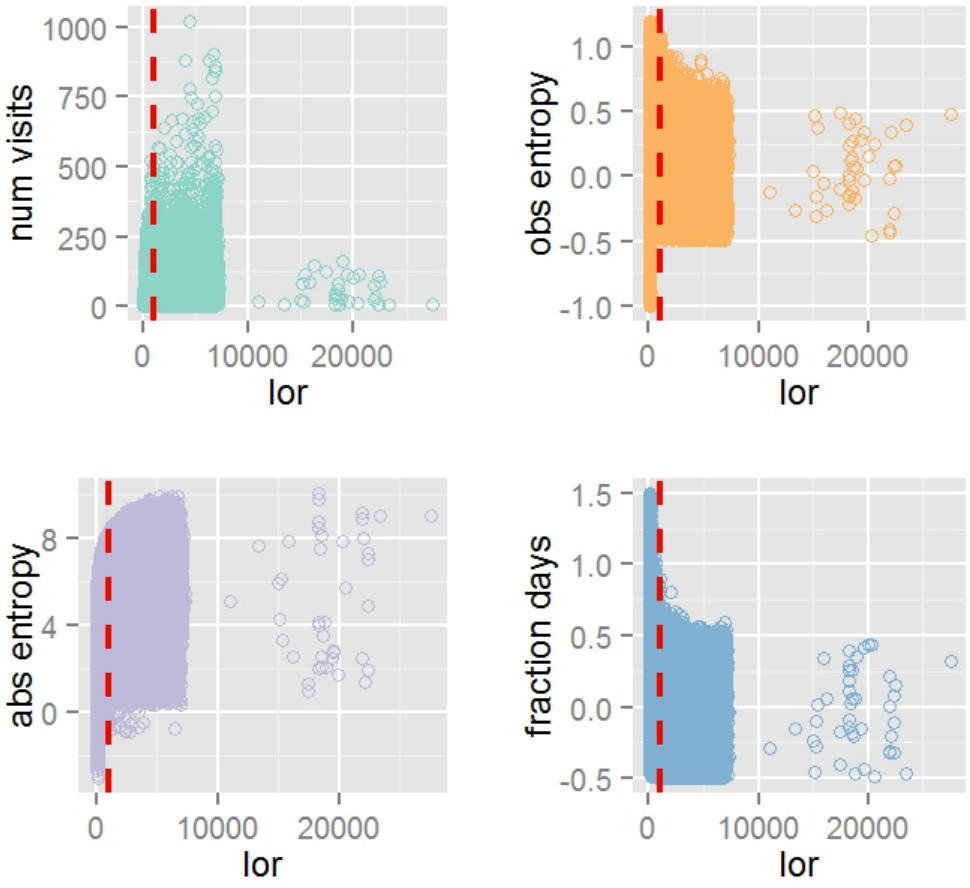

Figure 7.2: Comparison of study sample with larger glucose data set

by the fraction of visits (total visits/record length) over the duration of the record (approximately 0.02) and with most patients reporting less than 10 in total.

7.3.2 Feature-based Clustering

To provide a baseline clustering algorithm, I use a feature-based approach that is simple to develop and efficient to run. The features consist of the time series

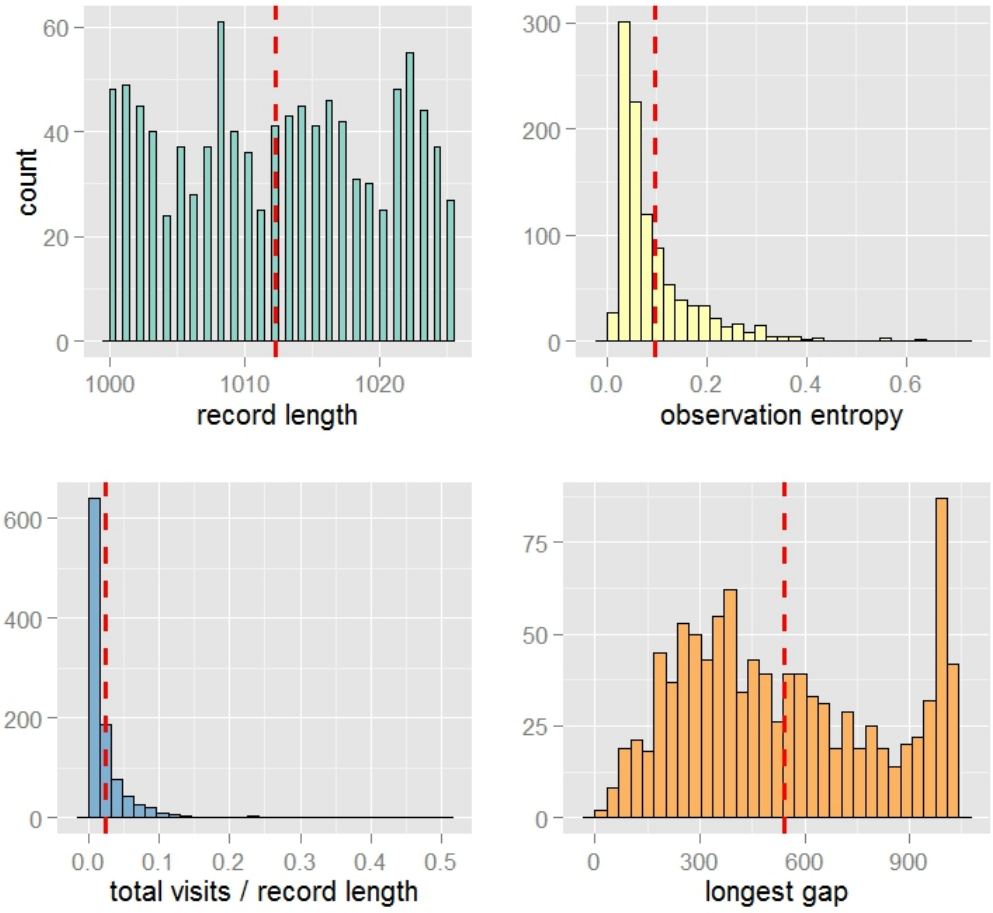

Figure 7.3: Descriptive statistics for the glucose study sample

statistics described in 7.2.1. These are used as input to the k-means algorithm, which is described in more detail in Section 4.1.1. Similar to the feature-based clustering method implemented for the hepatitis data described in Section 6.3.1, to determine the best value of k, I use the elbow method.

7.3.3 Discrete-time Abstraction

A general algorithm for semiparametric clustering using DT models is below. For discrete-time modeling, temporal sequences must be in the form of a time series, which requires that they are discretized into T units of interval Δt. When information is not observed, either the value must be imputed, or a default value can be used.

Procedure

Definition: For a collection of patients, X, let x_i be one of n patient sequences where $i \in \{1, ..., n\}$; λ_i is the ith patient's model; and k is the number of mutually exclusive clusters for the set $C_1 \cup C_2 \cup ...C_k$ that partitions X.

Input: Patient data set X, if spectral clustering is used, k

Output: cluster composition with each patient assigned into one of k groups

Algorithm 2 Semiparametric clustering with DT abstraction

for $(1 \leq i \leq n)$ **do**
  $x'_i \Leftarrow$ discretize(x_i) ▷ discretize temporal sequence x
  $\lambda_i \Leftarrow argmax_{\lambda_i} P(x'_i|\lambda)$ ▷ **abstract** x_i's model
end for
$\{C_1, ..., C_k\} \Leftarrow$ cluster$(\{\lambda_1, ..., \lambda_n\})$ ▷ **non-parametric clustering** step

The model structure was determined by learning the model parameters for every patient in the study sample. Since the number of patient models that converged was different for different model structures, I used the average likelihood to compare 3, 4 and 5 state models, and to assign a set of values to the initial probability

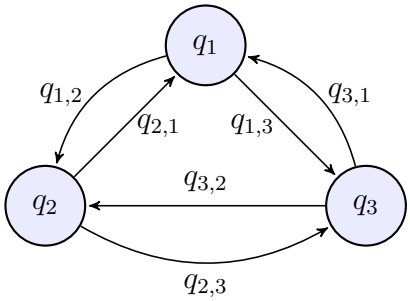

Figure 7.4: 3-state (discrete-time) HMM where q_1 through q_3 indicate the increasing risk of a hypoglycemic event

matrix of the final model.

The discrete time HMM that was used consists of a three-state model that shown in Figure 7.4 to represent low, normal and high risk states. Once a model was defined for the underlying disease process, the sequences were abstracted, then clustered, assigning every patient into one of k patient groups as described in the algorithm above.

7.3.4 Continuous-time Abstraction

An overview of the algorithm for the CT clustering procedure was described earlier in Section 6.3.3. Since the CT-model does not require a strict time series format, we use an alternative representation to model the sequence not by the daily value, but rather one that captures the total duration of daily glucose testing for the admission, or number of contiguous glucose tests per admission for the length of the patient's record. This representation more explicitly embeds problem specific context (*the importance of the number of contiguous tests*) and be used as the

training sequence for parameter learning.

Measurement Sequence Transformation

Sequence transformation is used to represent the temporal information in a new structure for parameter learning. For the CT model, this consisted of the start date of a testing period and the number of contiguous daily glucose test orders. In contrast to the DT time series representation that discretizes the entire measurement series by one-day time slices, it does not require the representation of unsupported data for parameter learning. In this case, the sequence no longer contains zero values, instead indications are made only for the first date during an admission where a glucose test was ordered, and a value that corresponds with the the number of days blood glucose was contiguously tested.

For two patients in the data set, an example of the DT and SC sequences used for parameter learning appear in Figure 7.5. I provide an example to further describe the process. Patient x_i, with lor = 15 has their first glucose test indicated on January 1, subsequent testing on January 5th through the 8th, and again on the 13th through the 15th. The discrete time learning sequence would be represented by the measurement sequence

$$x_i = [(t_1, 1), (t_2, 0), (t_3, 0), (t_4, 0), (t_5, 1), (t_6, 1), (t_7, 1), (t_8, 1), (t_9, 0), (t_{10}, 0), (t_{11}, 0), (t_{12}, 0), (t_{13}, 0), (t_{14}, 1), (t_{15}, 1)]$$

CT models do not need to force the representation of data that is unknown,

which is indicated by a '0' in the discrete time representation. Also, they do not require a fixed time interval that is the length of the smallest time granularity. Not only does this allow for a new representation, it more explicitly captures informative features of a measurement sequences including, the contiguous measurements, their density, and frequency. The new representation for x_i that we use for continuous-time modeling is x_i', and represented by the more succinct vector

$$x_i' = [(t_1, 1), (t_5, 4), (t_{14}, 2)]$$

In contrast to the DT model, this approach results in a measurement sequence that is equal to the number of each patient's testing 'episodes' contained in a patient's record. Since we need only to represent the data that is known, intervening time slices where no tests were observed are not required to be represented.

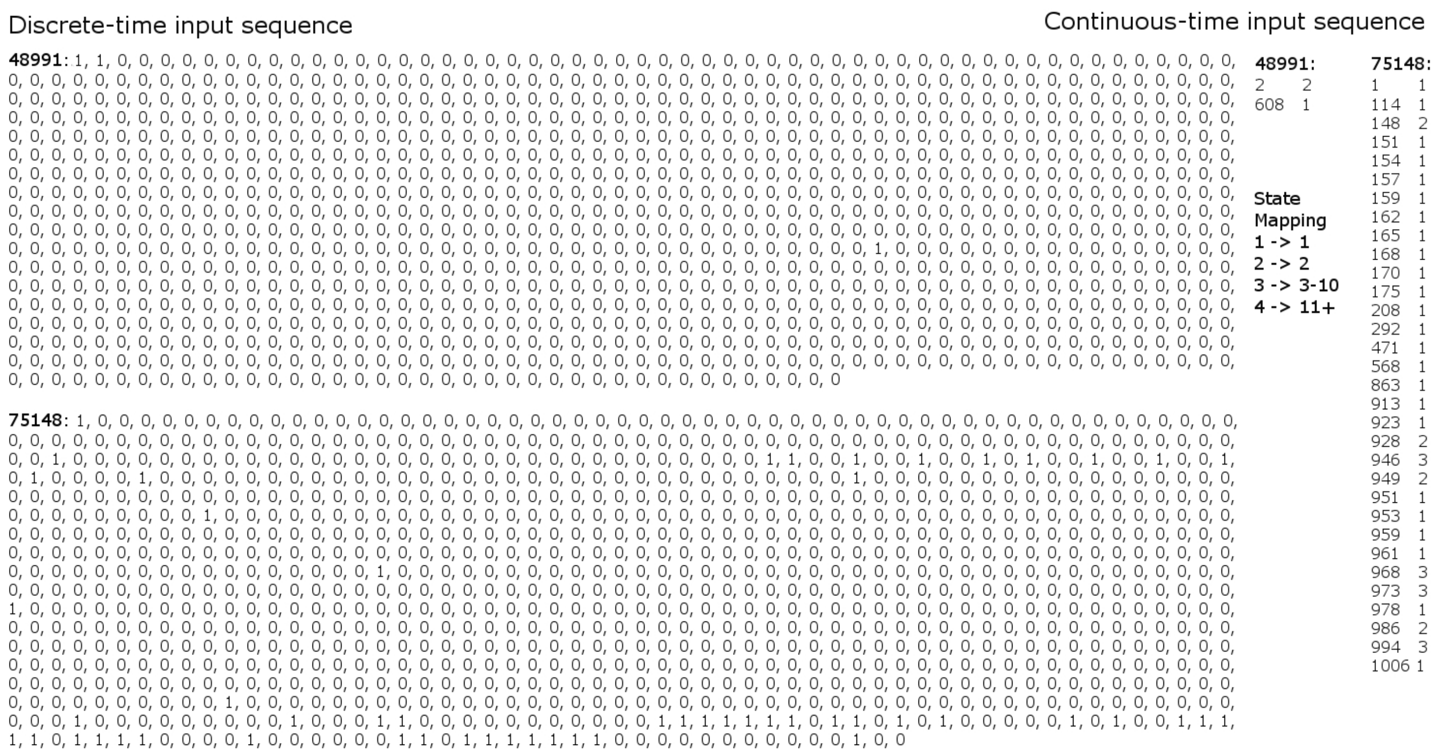

Figure 7.5: Measurement sequence transformations for daily glucose test patterns

Model Structure

In contrast to the hepatitis model, expert information indicating the model states, their finite values, and their conditional dependencies is not unavailable. Also, for the glucose model we are representing a latent or "hidden" state variable. In this section I describe how the CT-HMM was created. More details of discrete-time modeling of the glucose data can found in previous work [67] and a description of HMM models and their variants can be found in Section 3.4.1.

Since there are more model aspects (e.g., the number and representation of states) that need to be learned than with the hepatitis data, the model structure used here is guided by the theoretical framework provided by MSMs, which are used by biostaticians for modeling chronic disease dynamics in continuous-time. Two characteristics are relevant: the state semantics and the construction of the emission matrix.

Consistent with MSMs, disease states reflect an interval that corresponds with their acuity. In this representation, the lowest state indicates the absence normal blood sugar values, and the highest state corresponds with the most risk of a glycemic complication. Also, in MSMs the number of states in the transition intensity and emission matrix are equal. The emission matrix, or the additional model descriptor used to model the latent state is considered an error, or misclassification matrix, when the observation is correlated with a diagnosis [33], and I use this convention for modeling the glucose data.

State Estimation

The input to the estimator was the collection of all observation values from the transformed sequence. In the continuous-time representation this not limited to a day to day representation of the 0/1 measurement sequence, but instead consists of values that correspond with the length of glucose testing periods as described earlier in Section 7.3.4

A summary of the input values for CT-BN state estimation is shown in Figure 7.6. Since this distribution is dramatically skewed, we show three plots, two of which zoom in particular ranges, including (1,25), (2,10) and all observation values (1, 300+). This figure indicates what we'd expect to see, many isolated test orders for which there is no follow-up the consequent day, and what appears to be a power law between the state values and their frequency. As noted in the data description, blood glucose tests are commonly ordered in conjunction with tests like a metabolic panel, or for diabetic screening, and do not indicate the presence of diabetes. However, a series of contiguous testing, especially for more that two days, likely corresponds with a diabetic experiencing problems with their blood glucose management.

Three, four and five state mappings for the data were identified and the four state model is reported. The structure provides a mapping for each "observation", or instance of contiguous glucose measurements to one of the four disease states. The structure of this model is shown in Figure 7.7.

In MSMs, states indicate a risk associated interval between normal, and an acute or censoring state. In Figure 7.7 this corresponds to the state values. Also, I

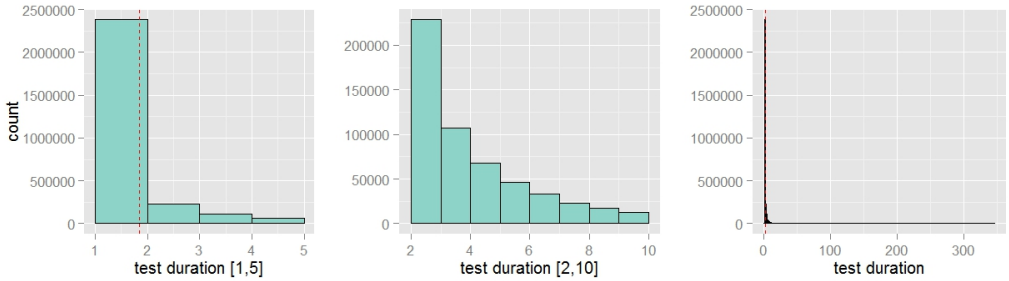

Figure 7.6: Distribution of contiguous glucose testing durations for all patients

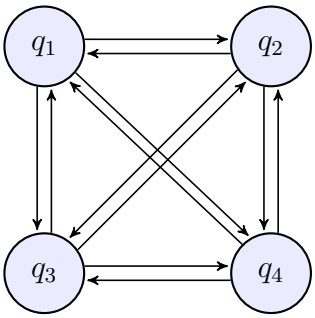

Figure 7.7: Four-state CT-BN, where states q_1 through q_4 indicate increasing risk of a glycemic complication and the transitions among states.

base the emission (misclassification) matrix on the MSM framework, assigning it the same number of states as the underlying state model.

Figure 7.5 shows the state sequence mapping for the CT model, the two different representations for discrete and continuous time abstraction from two patients in the data set #48991 and #75148. For the first sequence, #48991, what would consist of over one thousand iterations of a learning algorithm is reduced to two for a CT-BN. In the second case it is reduced to less than thirty iterations. This figure also shows the mapping that was learned from density estimation for observation values to corresponding disease states.

Model Learning

Using the four-state CT-BN shown in Figure 7.7, where q_1 through q_4 indicate the increasing risk of a hypoglycemic event, each patient's model parameters are learned from the observed data, which serves at the training sequence for the discrete and continuous-time HMMs.

For learning each patient's model from their measurement sequence, a forward-backward algorithm is used. For the discrete-time HMMs this algorithm can be found in Section 3.4.3. For the extension to continuous-time setting the Kolmogorov differential equations can be used. More about CT parameter learning appears in Section 5.2.3.

7.3.5 Nonparametric Clustering

Again, we apply two alternative nonparametric algorithms for clustering the embedded temporal models. I give a brief description of each below. More details on implementation of the spectral and Bayesian clustering can be found in the clustering description for the chronic hepatitis data set in Section 6.4.

Many varieties of spectral clustering algorithms exist. I use an implementation first proposed by Ng et al. [44], which builds upon existing work by normalizing the Laplacian affinity matrix before eigenvalue decomposition and selection of k largest eigenvalues. The best value for k is determined with the eigengap method.

To extend semiparametric temporal clustering to the Bayesian setting, I use *Dirichlet process Gaussian mixture modeling*, a technique with historical roots as

a density estimation method. A Dirichlet process Gaussian mixture model defines a Dirichlet Process Mixture Model (DPMM) by taking the limit of the number of mixture components, k, as the number of components in a hierarchical Gaussian Mixture Model (GMM) approaches infinity. More details on the algorithm can be found in Section 5.2.3.

Similar to the procedure for hepatitis data, to compute the model likelihoods and posterior distribution of the clusters I use a mean variational inference for the infinite Gaussian mixture model instead of Gibbs sampling. First I cluster the glucose test data for alternative values of α. For runs where the number of clusters are equal, I select the lowest score based on the BIC score.

7.4 Results

Since gold standard results are not available for extrinsic evaluation, I assess intrinsic qualities of the generated clusters to assess cluster quality. The silhouette value, which evaluates goodness based on a heuristic, is used to identify good clusters.

In addition, temporal series statistics and visualization techniques are used to further examine component clusters produced by the alternative settings, discrete and continuous-time abstraction, and Spectral Clustering (SC) and nonparametric Bayesian Clustering (BC) methods.

7.4.1 Feature-based Clustering

To select the best model for the different approaches, I first looked for a dramatic drop in the clustering procedure's objective function. If no dramatic drop could be observed, or more than one distinct elbow resulted, BIC was used to select among competing models. Although k-means does not generate a likelihood, the sum of the squared error was used as a pseudolikelihood value.

Also, some of the temporal features available for clustering are highly correlated. To select the best set for clustering, I applied feature selection using different variable combinations and values of k. The final model consisted of the following three features: the length of the longest gap (Eq.7.5), the entropy of the measurement sequence (Eq.7.2) and the patient's total number of visits (Eq.7.3).

To compare the quality of clusters generated by a simpler feature-based clustering approach, with that of more sophisticated semiparametric methods, I used the silhouette validation technique described in Section 4.2.1 to intrinsically validate discovered clusters. Silhouette values are a heuristic commonly used to assess the goodness of clusters, providing information on both inter-cluster compactness and the level of distinction between different clusters.

Cluster assignments for k=5, shown in Figure 7.8, were evaluated and show poor results. Only a small fraction (18%) of patients showed the minimum criteria for a good silhouette value. Applying spectral clustering instead of k-Means for clustering the feature vector resulted in improved silhouette values, with over 60% of patients reporting good silhouette values. However, when patients were grouped by cluster, both approaches failed to show distinction in terms of the time

series summary statistics that were not used for clustering.

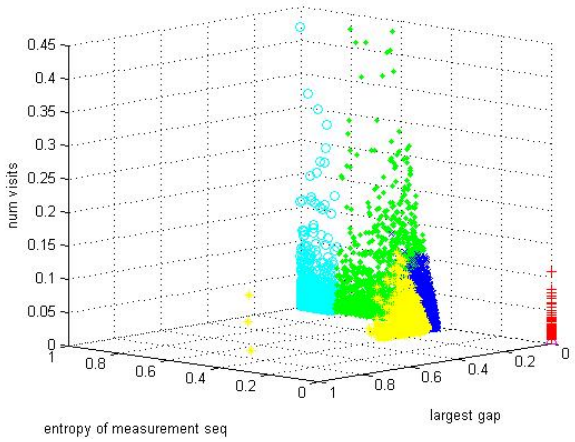

Figure 7.8: k-means clustering: $k = 5$

7.4.2 Semiparametric Clustering Methods

The need to establish methods to compare discrete and continuous-time models has been noted by researchers [45, 46]. The two representations have different assumptions and can motivate the use of different abstraction or representational aspects of the experiment. For example, my continuous-time abstraction method learns model parameters more directly from the time series, and does not force the representation of data without support, resulting in a much more succinct observation vector for parameter learning that appears as Figure 7.5.

Although we compare the two abstraction methods in terms of silhouette values, it is important to consider the challenges in comparing two different types of models. For that reason, aggregate time series statistics (described in Sec-

tion 7.2.1) are used to further characterize clusters, and interpret their contextual meaning in relation to the problem task.

Intrinsic Validation

For discrete-time model-based abstraction, a three-state HMM was used to learn each patient's model parameters, and paired with nonparametric clustering. As noted previously, silhouette values close to 1 indicate a strong level of cluster compactness and distinction from other clusters.

Using DT HMM abstraction, the clustering model where k=4 indicated that 75% of patients report a value of 0.9 or greater, and outperforming the second best candidate where k=9. The silhouettes for the clustering results appear in Figure 7.9. Compared to the silhouettes generated from feature-based, there are notable improvements, and almost 94% of patients reported the minimum criteria for a good silhouette.

For the continuous time models with spectral clustering, the best candidate for k is 7. Based on the assignments for this value, only two clusters report a high mean silhouette, 89%, and 93%, but their membership constitutes 17% of the sample. The second best value is $k = 9$, which has two clusters with the same membership as the top clusters from the $k = 7$ model. The silhouettes for these two models appear as Figures A.1 and A.2 in the Appendix.

In contrast to the discrete-time abstraction, silhouette values for continuous-time abstraction paired with nonparametric clustering are lower. This suggests that in terms of the silhouette values, which aim to measure compactness and distinc-

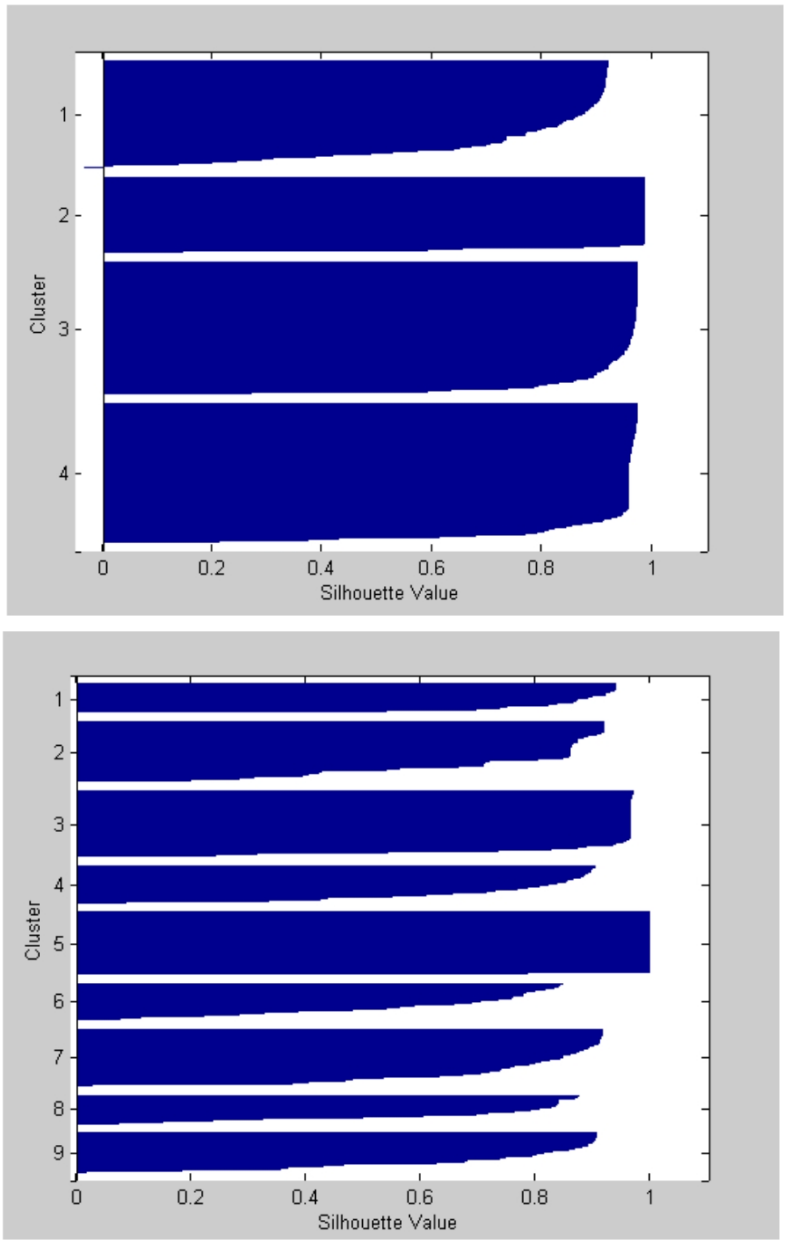

Figure 7.9: DT-SC silhouettes for $k = 4$ and $k = 9$

tion of the discovered clusters based on casting clustering as a simple partitioning problem, the discrete-time HMM presents modeling advantages.

Since good intrinsic validation does not guarantee high extrinsic quality, we further analyse the clustering assignments generated by both methods using aggregate time series statistics.

Aggregate Time Series Statistics

To further compare the results of discrete and continuous-time abstraction, we use external measures that are not used as features for clustering in this experiment, and represent aggregate time series statistics averaged by cluster. Preliminary work [67] for this thesis demonstrated the relevance of these statistics for capturing temporal features from the glucose data. These statistics are used here to provide information about the entropy of the cluster's time series sequences, and the duration and frequency of glucose testing.

Figure 7.10 shows clusters formed by Discrete-Time (DT) HMM abstraction and 7.11 shows Continuous-Time (CT) HMM abstraction with spectral clustering. All DT and CT charts are plotted on the same scale. The colored markers indicate different clusters and their size is proportional to total membership.

In Figure 7.10, the y-axis, labeled 'Ordered Tests', shows the average number of glucose tests ordered by a physician for patients in each cluster. The left side of the x-axis, labeled 'Test Series', indicates the average number of states recorded for patients by cluster. More information on the mapping for state sequence representation was described earlier in Section 7.3.4.

On the right side of Figure 7.10 the x-axis shows the average sequence entropy per cluster. As noted earlier, the entropy of a patient's raw time series sequence has

been shown to be an informative feature for clustering in previous work, and helps to show distinction between discovered clusters in this work. More information on the calculation of aggregate time series statistics can be found in Section 7.2.1.

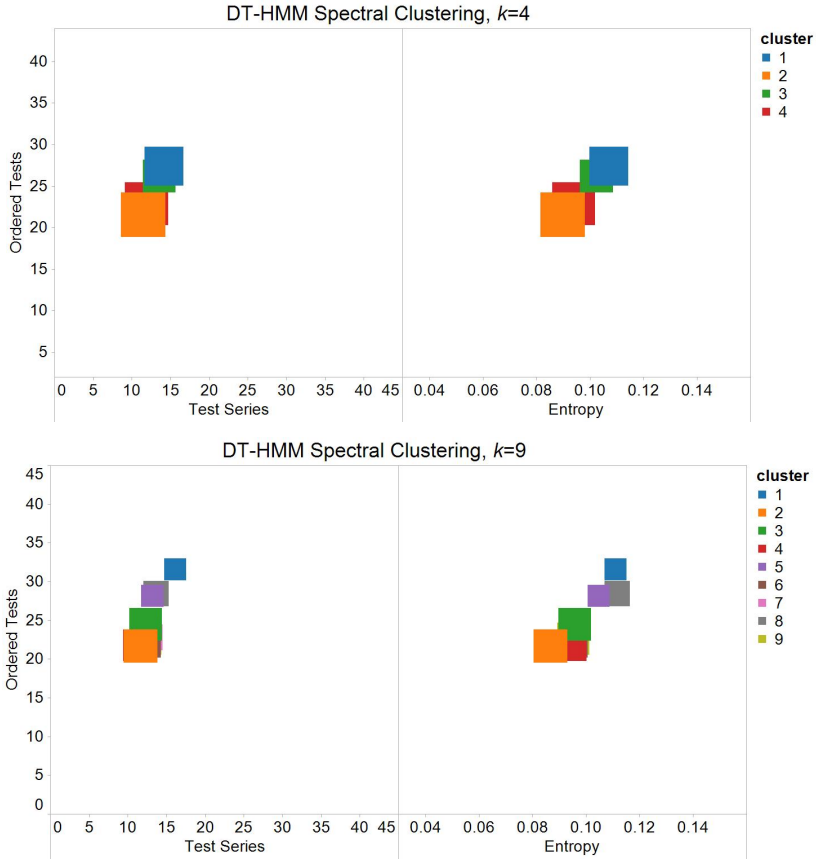

Figure 7.10: Patient clusters generated using DT abstraction and spectral clustering by time series statistics

The relationship between a cluster's average ordered tests, and the average test series length provides information about the corresponding set of patients' risk of a glycemic complication. When these two values are equal, it suggests that physician ordered glucose tests are in isolation, and so are rarely repeated after the

initial test. As the number of visits becomes greater than the number of states, this is indicative of more repeated testing, and a cluster composition that is at higher risk for glycemic complications.

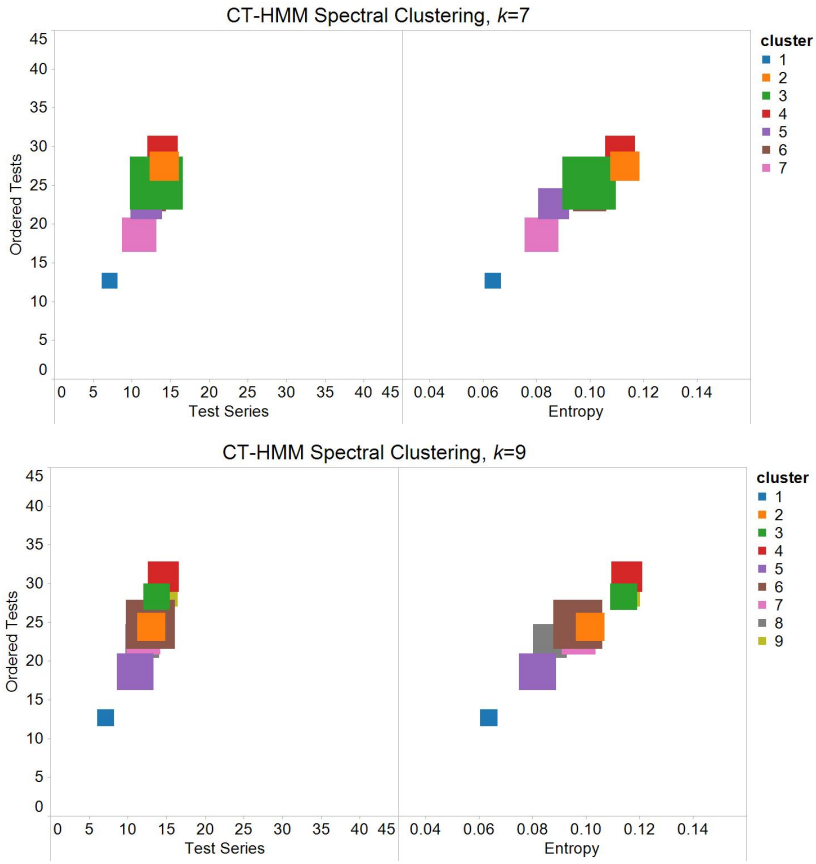

Figure 7.11: Patient clusters generated using CT abstraction with spectral clustering

Based on the cluster averages for the time series statistics that appear in Figure 7.10, discrete-time abstraction does not appear to provide a meaningful clustering result. For all clusters the number of total tests, test series, and the sequence entropy is similar. For the CT abstraction models, there is more variation between

the clusters formed, and the distinctness of clusters is better. From Figure 7.11, it can more easily inferred what clusters are associated with patients at low and high risk for hypoglycemic events. Figures 7.10 and 7.11 are inconsistent with what is indicated by their cluster silhouettes, and suggests the opposite of what would be expected.

The inconsistencies between the goodness of silhouette values and time series statistics points to the complexity of the cluster validation problem [26]. In my experiment, the models were trained on sequences that had a different representation and process model. Also, for the discrete time learning case, estimates are updated many more times than in the continuous-time case, and reflect a probability distribution. In the continuous time model, the matrix is no longer a probability matrix, rather it is an intensity matrix. These findings may suggest that that the discrete-time abstraction method overfits the model, and could be learning the structure of the training sequences instead of parameters related to the dynamics of the underlying process.

Computing Time

Another aspect for which DT and CT abstraction types can be compared is computational efficiency. Based on my experiments, of the two computational steps that are performed serially, abstraction and embedded model clustering, learning the parameters for each patients model is substantially more resource intensive. Since the rate of technological developments can quickly make a system's computational resources obsolete, I compare the relative efficiency of alternative learning

methods used in this thesis instead of providing system specific parameters.

The average fraction of days measured for the sample is 0.02. Since a CT model is trained only on data that is observed, for DT-HMM learning over 1000 more iterations of the forward-backward algorithm was executed. Although the computational time for running the forward-backward algorithm for a continuous-time model is longer than for a discrete-time model on the same data, when the data is sparse it is more efficient.

The abstraction step entails using each patients observation sequence, or singleton time series, to learn their models parameters with the either a DT or CT Baum-Welch algorithm. For the glucose data set with record lengths of 1000-1025 days, the DT method, which requires that the model likelihoods are updated at each Δt, takes 13.36 times more clock cycles than CT model learning. Although updating the model likelihoods using the CT extension of Baum-Welch results more computational steps, they are only required when data is observed, resulting in many less model updated.

It is important to note that DT learning becomes increasingly burdensome when clustering larger samples. For example, training a CT model for every patient with a length of record of one year or greater (n=161,276) required just over one day on a quad-core machine using all processors. Based on our relative computational time estimate, this would take a DT method over two weeks. It is not only resources that make DT abstraction unattractive for large datasets that are sparse, but the number of model updates that must occur based on data that is forced without support.

For the glucose sample that consists of patients with record lengths in the range of 1000-1025 days, clustering CT embedded models took k-means, spectral clustering and nonparametric Bayesian clustering under thirty seconds. For the subset of patients with record lengths of one year or greater, nonparametric Bayesian clustering took almost twice the amount of time as k-Means. Spectral clustering, as described in Section 4.1.2, will not scale. However, an alternative method that maintains the integrity of spectral methods but is designed for large data sets may be used to produce cluster assignments [76].

7.4.3 Comparing Nonparametric Clustering Methods

In this section, I further examine the application of continuous-time semiparametric clustering, and compare the results of cluster assignments generated by spectral clustering and Bayesian clustering. Similar to my comparison of DT and CT models, I use silhouette values for intrinsic validation of the clusters and the aggregate time series statistics for further evidence of good clusters.

Silhouettes

To compare performance of CT abstraction with Bayesian clustering instead of spectral clustering, Figure 7.12 shows the silhouettes by cluster for the four-state glucose model used for continuous-time abstraction that generated *five* clusters (CT-HMM BC $k = 5$). The silhouettes for CT-SC $k = 7$ and $k = 9$ appear as Figures A.1 and A.2 in the Appendix.

Each observation's silhouette is grouped by cluster along the y-axis, and mem-

bers are sorted within each cluster by each assignment's value. As stated earlier, 0.6 is generally the threshold for a good cluster in terms of similarity to other members of its own cluster and distinction from members of other clusters.

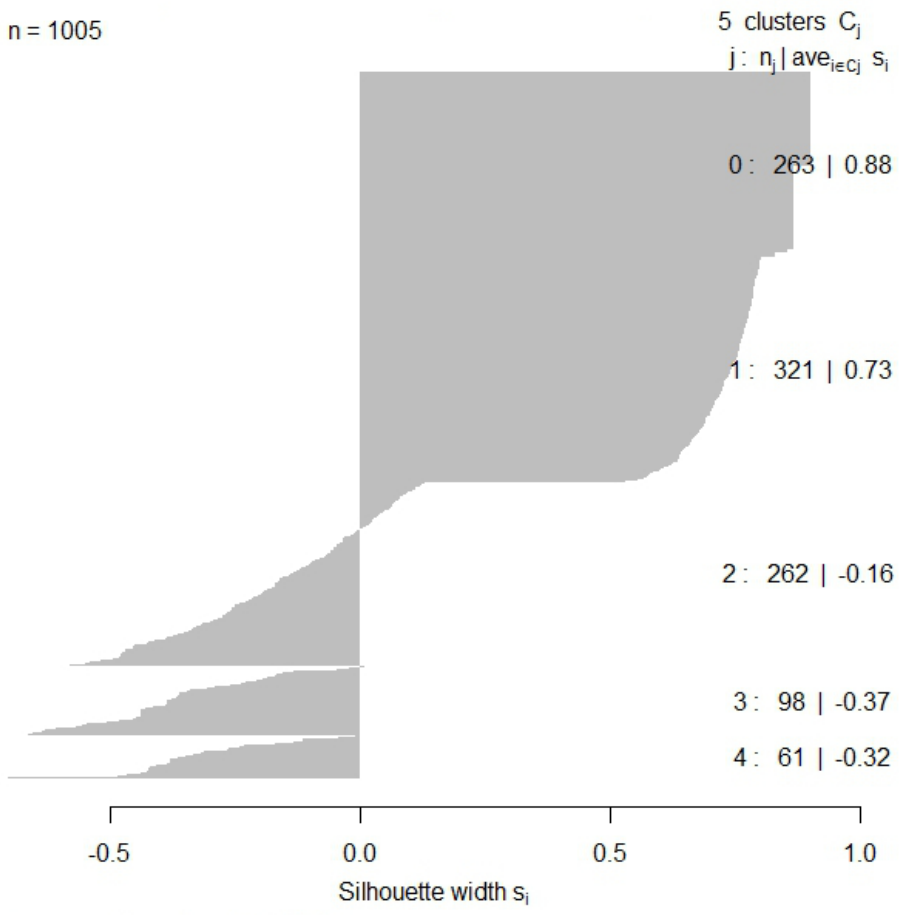

Figure 7.12: CT-BC silhouette by cluster for 4-state model.

Based on intrinsic validation with silhouettes, the k=5 CT-HMM BC assignment suggests the best clustering model. Table 7.1 shows each cluster's size,

mean and average silhouette value, and suggests that c_0 and c_1 are good based on intrinsic criterion.

Table 7.1: Size, mean and median silhouette values by cluster

	c_0	c_1	c_2	c_3	c_4
size	263	321	262	98	61
mean	0.8841	0.7267	-0.1573	-0.3669	-0.3167

Aggregate Time Series Statistics

Since a good silhouette score is not a guarantee of a good clustering, I assess cluster assignments using the same time series statistics described in the previous section.

Figure 7.13 shows aggregate time series statistics by cluster for the results of continuous-time abstraction paired with Bayesian clustering. The results paired with spectral clustering appear in Figure 7.11 of the previous section and are reported on the same scale. On the y-axis is the average number of total tests per cluster. On the left side of the x-axis is the average number episodes, or periods of continuous testing, and on the right side of the x-axis is the average sequence entropy per cluster.

Although the differences between CT-SC and CT-BC methods in Figure 7.13 aren't striking, it appears that the Bayesian clustering results present more complete and distinctive clusters. More noticeable is the contrast between the the DT-SC approach shown in Figure 7.10 with that of CT-BC.

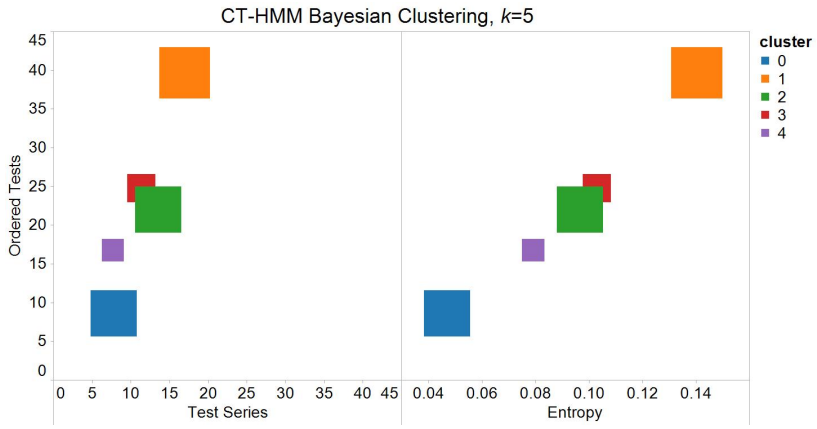

Figure 7.13: Patient clusters generated using CT abstraction with Bayesian clustering by time series statistics.

Consistent with the findings from intrinsic validation with silhouettes, the chart for CT-BC k=5 suggests two good quality clusters. Table 7.2 reports additional time series statistics for the clustering assignment that are useful for the description of discovered clusters, and includes the number of total tests (Tests), the number of contiguous blocks of daily tests (Episodes), the entropy of the measurement sequence (Entropy), and the fraction of days measured (Fraction) by cluster. The calculation and a discussion of these aggregate metrics appear in Section 7.2.1.

Clusters with patients at the lowest risk for glycemic complications will exhibit less tests overall, and with a number of total tests that is equivalent to the number of entries in their state sequence representation, or test series. This suggests that few tests are ordered, and when they are, for these patients, they are typically not repeated on the next day. In contrast, when the average number of episodes for a cluster is relatively lower than the average test number, it is more

Table 7.2: Time series statistics aggregated by cluster

CLUSTER	N	TESTS	EPISODES	ENTROPY	FRACTION
c_0	263	8.60	7.81	0.05	0.01
c_1	321	39.64	16.98	0.14	0.04
c_2	262	22.00	13.53	0.10	0.02
c_3	98	24.77	11.38	0.10	0.02
c_4	61	16.74	7.69	0.08	0.02
TOTAL	1005	24.08	12.57	0.10	0.02

indicative of repeated daily glucose testing during the admission.

Based on the total number of visits Table 7.2 suggests that c_0 and c_1 represent patients with the lowest risk and highest risk for glycemic complications. With only one exception of c_4 reporting the lowest score for the number of episodes, c_0 reports the lowest scores for all measures, and c_1 the highest. Although the number of episodes for c_4 is, on average, low (7.69), the number of visits is almost twice that of c_1 (16.74) and indicative of higher risk patients, whereas in c_1 these two values are approximately equal (8.60 to 7.81), consistent with a lower risk profile.

7.4.4 Clinical Relevance

The goal of the temporal mining task was to discover patient groups that are useful or meaningful to:

- stratify patients into groups that correspond with their propensity towards glycemic complications
- characterize temporal patterns associated with any clinically significant groups.

Similar to our hepatitis experiments, the models that serve as input to clustering characterize patient disease-state dynamics, or their instantaneous risk of transitioning to and from diabetes related disease states. These subpopulation models can be useful for describing the dynamics of clinically significant groups. In addition to assessing results with intrinsic validation, I examine the intensity matrixes generated by nonparametric Bayesian clustering, and demonstrate how the difference among discovered groups can be visualized using Q matrix plots. Also, I show how additional clinical insights about the types of patients that make up individual clusters can be facilitated by sequence heatmaps.

Q Matrix Plot

The values for a characteristic Q matrix for each cluster is shown below. As noted earlier, the four-state model reflects increasing disease severity, with q_1 indicating normal blood sugar levels and q_4 a high risk of a glycemic complication. Each q matrix entry, q_{ij}, corresponds with the instantaneous risk of moving from state i to state j.

In order to help interpret the clusters, I use a Q-matrix plot to visualize the clusters. Figure 7.15 corresponds with the first row of entries in the Q matrix, which corresponds with transitions from the state q_1, normal blood sugar levels, to those at higher risk of a hypoglycemic event. Individual clusters are indicated by color and number along the y-axis, and the x-axis represents the z-score. All of the q_{ij} entries of the 4×4 intensity matrix Q appear in Figure A.3 in the Appendix.

The plot for q_1 transitions shows a temporal trend that is consistent with the

$$Q_0 = \begin{pmatrix} 0 & -25.55 & -11.09 & -14.54 \\ 14.66 & 0 & -4.74 & 1.55 \\ 12.72 & 3.84 & 0 & -1.49 \\ 11.09 & 3.38 & -13.87 & 0 \end{pmatrix} \quad Q_1 = \begin{pmatrix} 0 & -2.28 & -2.83 & -5.3 \\ -0.61 & 0 & -2.24 & -4.19 \\ -0.76 & -1.61 & 0 & -3.72 \\ -0.52 & -2.84 & -2.59 & 0 \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} 0 & -0.65 & -12.45 & -11.92 \\ 4.93 & 0 & -7.30 & -8.62 \\ 8.00 & 0.37 & 0 & -5.60 \\ 7.15 & -1.31 & -1.52 & 0 \end{pmatrix} \quad Q_3 = \begin{pmatrix} 0 & -20.04 & -4.64 & -8.17 \\ 8.98 & 0 & -0.36 & -0.46 \\ 2.95 & -2.13 & 0 & -7.31 \\ 5.52 & 0.75 & -2.35 & 0 \end{pmatrix}$$

$$Q_4 = \begin{pmatrix} 0 & -2.76 & -0.39 & -3.85 \\ -13.73 & 0 & -5.50 & -7.00 \\ 0.35 & 0.92 & 0 & -2.66 \\ 6.38 & -5.97 & 2.22 & 0 \end{pmatrix}$$

Figure 7.14: Intensity matrices for 4-state, 5 cluster glucose model

patients that have less risk of a hypoglycemic event in c_0 (indicated as a 1 on the y-axis) and poorer health in c_1 (indicated as a 2 on the y-axis). The negative value indicated that patients in c_0, are more likely to maintain normal blood sugar levels than transition to higher states compared with the average. Patients in c_1, show the opposite, and are more likely to transition to a state associated with a higher

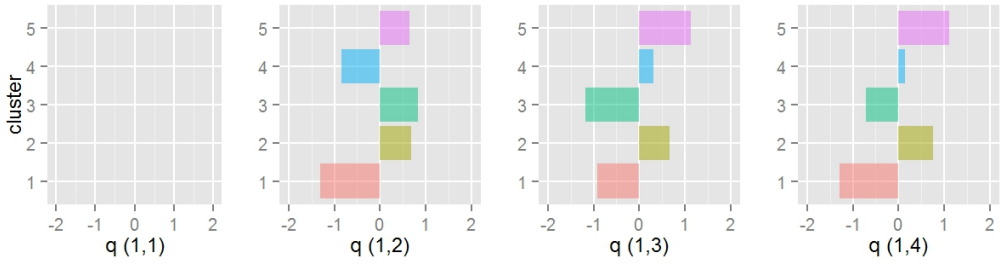

Figure 7.15: Characteristic Q matrices by cluster

risk of hypoglycemic event.

For the clusters that report low silhouette scores, one shows a clear trend, c_4 (indicated as a 5 on the y-axis). Based on the aggregate time series statistics, c_4 has a high total visit to total episode average, indicating more repeated daily testing, also at high risk for transitioning to higher disease states. Other clusters, do not show a consistent trend as the states increase.

Heatmaps

Although they do not quantify cluster goodness in a way that is easily comparable among clusters or to benchmark performance, visualization tools for assessing cluster goodness provides additional insight that is not captured by silhouettes, aggregate time series statistics, or even extrinsic validation measures. Simple visuals, such as heat maps for states in the sequence, are not only more intuitive, but can allow for the possibility of interactive exploration.

To further analyze c_0 and c_1, the two clusters that report high silhouette values and distinction in terms of aggregate time series statistics, I visualize the state sequence for patients in each of these clusters in Figure 7.16 using a heatmap. The values of states increase with the saturation of green, with yellow indicating the absence or very low probability of diabetes, and dark green the highest in severity.

This heat map shows two important and distinctive trends. That is c_0 patients do not show any indication of higher disease states, or that they start in a higher state, but after the initial episode there is no longer a documentation pattern that

Figure 7.16: Comparison of c_0 (left) and c_1 state sequences.

suggest they are at a high risk for glycemic complications. However, patients in c_1 exhibit higher states, and they do so more frequently.

7.4.5 Generalization

To determine if results were unique to the study sample, I replicated the semiparametric Bayesian clustering on two additional patient samples with *shorter and longer time series* than the study sample. The shorter sample (n=1041) consisted of patients with medical record durations of 975-999 days. The longer sample (n=1033) consisted of patients with medical record durations of 1026-1050 days.

I first used Multidimensional Scaling (MDS), which is an exploratory visualization technique that can be used to determine the similarity of individual cases

in a high-dimensional dataset in a way that is easier to comprehend by humans. The MDS algorithm requires the calculation of a distance matrix, d. Using D, the MDS generates a coordinate matrix with a configuration that minimizes a loss function. In comparison with principal component analysis, its strength is that it not only provides the ability to transform a higher-dimensional space to lower dimensions for surface matching, but it preserve distances between points as much as possible.

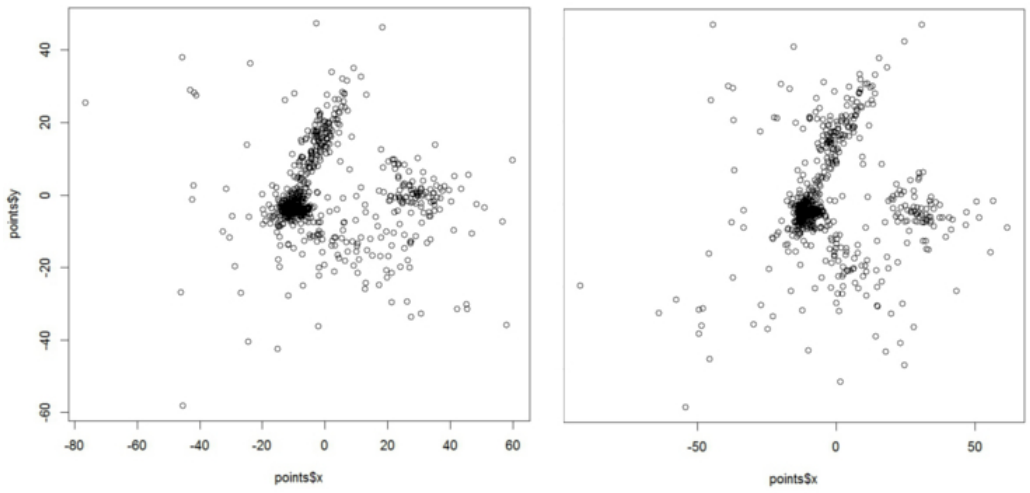

Figure 7.17: Multidimensional Scaling: shorter (left) and longer (right) sequence comparison.

In Figure 7.17 I show the scatterplot for MDS applied to a Euclidian distance matrix that was generated from calculating the dissimilarities among the CT embedded models. On the left the shorter sample appears, and on the right is the longer duration sample. Despite the absence of the intermediate interval, the study sample of patients with record lengths of 1000-1025 days, the embedded model

values show very similar patient mixies. This suggests that the ability of the CT abstraction method to generalize to other samples from the dataset is good.

In Figure 7.18 I show cluster means for the same time series statistics reported in Figure 7.13 for the sample of patients with record lengths of 1000-1025 days. In addition to the MDS visualization, these plots suggest that similar patient characteristics are revealed by the method from all three population samples, and that the abstractions are useful for discovering clinically significant groups that correlate with a group-level risk of glycemic complications.

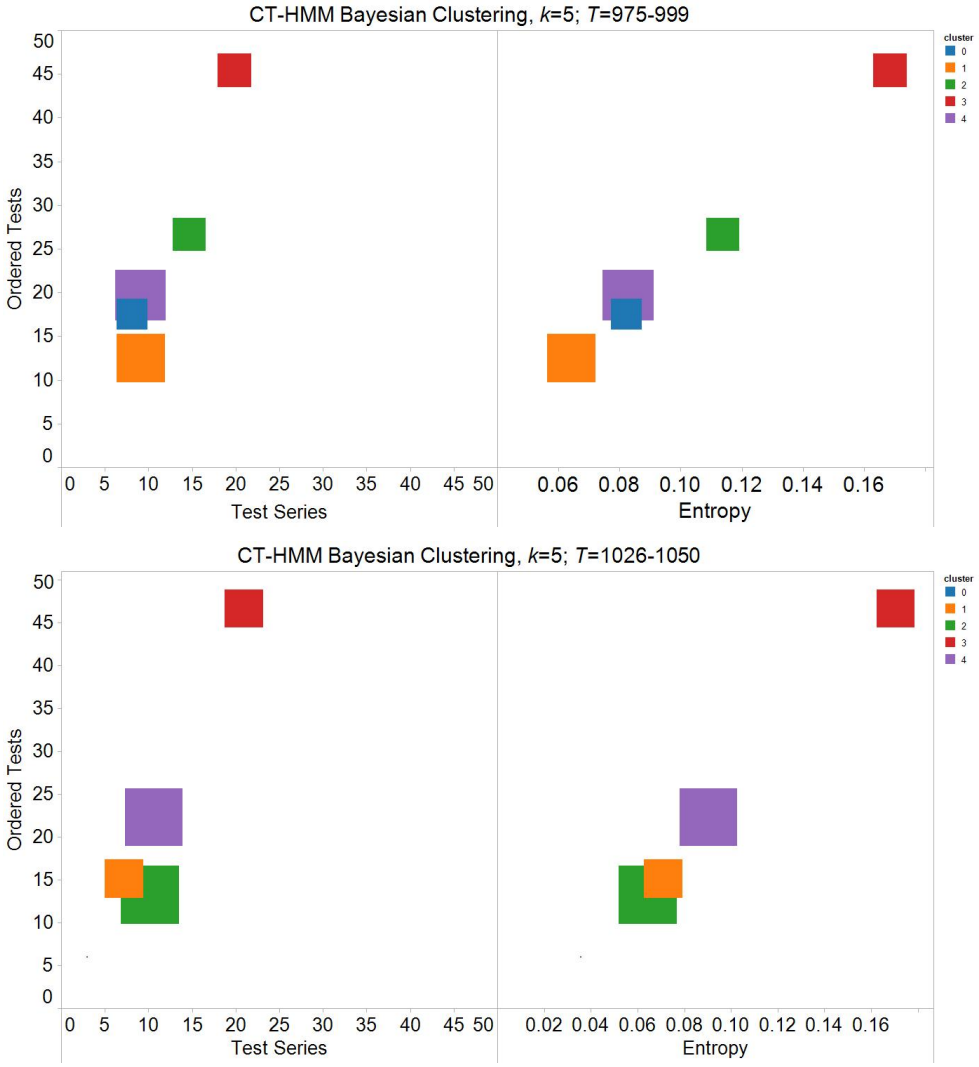

Figure 7.18: Semiparametric Bayesian clustering applied to patient samples with shorter and longer record durations.

Chapter 8

Conclusion

8.1 Brief Summary

The growth of temporal data sets has outpaced the development of effective methods to process them. It is not only the abundance of data that is challenging, but the arbitrary sampling schemes, irregularities and noise that is associated with many real world data sets that serve a secondary purpose as research instruments.

Using digital health data as a case study, I developed a new exploratory analysis methods to facilitate the meaningful use of temporal data. Specifically, I develop a new method for embedding Continuous-Time (CT) Bayesian networks for temporal clustering, extending semiparametric temporal clustering for clustering large, sparse, arbitrarily sampled observational datasets. Parametric abstraction is paired with nonparametric Bayesian clustering, to avoid the problem of having to prespecify the appropriate number of clusters.

In this thesis, I demonstrate that my clustering approach can produce meaningful clusters from noisy, incomplete patient data by validating clusters. For multivariate temporal data, my approach gives over a 20% relative improvement in the B-cubed measure over a state-of-the-art system. In the case when gold-standard results are unavailable, I use intrinsic validation measures, and time series statistics that have been shown as informative for feature based temporal clustering to present observable differences in cluster-level time series sequences and provide evidence that discovered patient groups are clinically meaningful. By applying semiparametric Bayesian clustering to shorter and longer temporal sequences, I show that my method can generalize to model objects of shorter and longer temporal durations.

Also, to facilitate the interpretation of model-based clustering results so that they can be interpreted by end-users of my clustering application, I normalized cluster-level characteristics using z-scores. Visualizations I call Q-matrix plots help to make cluster level comparisons in terms of disease-related risk.

There are two key advantages to my method over current applications for semiparametric clustering. When there are no natural time slices, continuous-time models can be used to more directly reflect sequential dependencies. They avoid the requirement of discretizing the time intervals, and representing data without support. Also, CT models can provide efficiency gains for sparse data sets. Although updating the model to reflect new temporal information requires more calculations, there are fewer model updates making learning more efficient.

Another benefit is presented by the use of nonparametric Bayesian methods,

which define the clustering problem as identifying the components of infinite mixture, where k is a random variable in the model. Relative to other nonparametric clustering methods such as spectral clustering, Bayesian clustering does not require that k is known in advance. Also, nonparametric Bayesian clustering provides a generative model that can describe group structure at the population and subpopulation level, and helps to describe the underlying data generating phenomena.

One benefit over the continuous-time abstraction over the discrete-time abstraction method that should be noted is convergence. For HMM based abstraction, we were able to learn each patient's model priors. Of the 1024 patients sampled, a CT-BN abstraction method learned 1005. The difference is attributed to models that did not converge, which was associated with two types of patients: those with very few observations, and patients with sequences that have parameter estimates that are dramatically different that the priors that are determined from the naive estimation method.

8.2 Lessons Learned

For both data sets, the goal was to learn clinically significant groups of patients from unlabeled temporal data. Although I show that my method can be used to detect meaningful patient clusters, it appears that it works best on those patients that are at very high, or at minimal risk of disease related complications. For example, Section 6.5.6 discusses the results of my method applied to chronic hepatitis

patients. When compared to the gold-standard, one cluster, c_4, shows a positive predictive value (PPV) of 93% for identifying patients with little or no fibrosis. Also, the two clusters that are the most medically complex, c_2 and c_3, can be combined to report at PPV of 96% for patients with serious liver conditions such as bridging fibrosis or cirrhosis.

Section 7.4.4 discusses the clinical relevance of the results that were generated by applying my method to the hepatitis data set. A gold standard was not available to perform extrinsic validation. However, two clusters that were discovered show high intrinsic validity. Cluster-level transition intensities, and visual evidence, indicates that these clusters likely correspond with patients that are at high, and at low risk of glycemic complications.

Another lesson that was learned from these experiments is related to nonparametric Bayesian clustering. A key benefit of this clustering method in contrast to spectral methods is that k does not have to be known in advance. However, I found that the scaling factor, α, needs tuning to produce a reasonable cluster size. Also, variational methods, which offer an alternative to MCMC, require that the upper-bound for the number of clusters is provided.

8.3 Limitations

In this section I discuss important limitations of my work. These include those limitations that are specific to the new method that I have developed, and problems related to clustering temporal data sets more generally.

Although my method helps to avoid issues associated with the discrete time modeling, it does not address all of the challenges posed by incomplete data. Instead of forcing the assumption of data without support, the CT model only models observed data. This approach avoids the need to incorporate a default or imputed value, but does not explicitly address the problem of data that is not missing at random.

For learning abstraction models, the methods used to learn the parameters of patient models may not converge, resulting in the inability to provide cluster assignments for all observations. This is more likely to happen when the initial probability matrix is not a good estimate of the posteriors.

In term of clustering, determining the similarity of HMMs is an active research area [23, 34]. This work may benefit form exploring a variety of distance metrics for computing the similarity between individual patient models.

Standard evaluation methods are based on the simplification of clustering as a partitioning problem that is independent of the method used or deeper problem context. For example, the state-of-the-art system with which I compare the results of semiparametric clustering is a hierarchical method. It is possible that a specific type of clustering algorithm is more useful for deriving meaning from clusters, but may not score as well in terms of intrinsic or extrinsic criteria.

External validation metrics assume that erroneous assignments are equivalent. However, for many important clustering problems clusters are ordinal, and express a discretization of an interval such as risk. In the case where the true classes are on a scale of one through five, misplacing an observation that should be in

cluster one into cluster five is much worse than assigning that same patient into cluster two. When the clustering can also be viewed as a stratification problem, additional formal constraints that metrics should satisfy would likely provide a better estimation of cluster quality.

8.4 Future Work

The introduction of continuous-time modeling frameworks, and their use in clustering provides numerous opportunities for future work. In terms of temporal abstraction, a learning theory would be more powerful if it could capture the self-selection mechanism that occurs in many real-world observational data sets into the temporal abstraction step, and other informative sequence features.

For modeling population-level disease dynamics, biostaticians have already developed continuous-time models to capture auto-regressive properties, integrated piecewise linear constant functions that more precisely examine risk based on the duration in a model state, and additional clinical knowledge to estimate risk in the presence of informative sampling times. It is possible that the work in this field can lend useful methods, or at least inform the development of new machine learning methods for temporal clustering.

There are several future directions for parameter learning. Markov chain Monte Carlo sampling and methods of moments algorithms, which provide an alternative to EM based learning, have shown high accuracy for estimating model parameters [20] and may provide additional benefits for this work.

Other areas of future work that would be especially useful for popularizing temporal clustering include the development of a clearer theory, or taxonomy for identifying the best type of temporal model for abstraction. This would be of tremendous value to researchers. Also, visualization tools have the potential to play an important role in the temporal clustering process. They can assist in the interpretation of key findings to domain experts, and in terms of cluster validation, allow the perusal of clustering assignments, especially for which there is no gold-standard results, for which qualitative judgements can be derived based on problem context.

Appendix A

Supplementary Materials

Table A.1: Description of blood and urine tests

Laboratory Tests for Liver Disease	
Platelets (PLT)	Platelet counts for hepatitis patients with cirrhosis are low, but can be the result of other causes. This lab test measures the platelet number in the blood.
Bilirubin (D-BIL)	This test measures the bilirubin level in the blood, and is produced by the breakdown of hemoglobin. If the liver is not functioning correctly, it will not be properly excreted.
Cholinesterase (ChE)	This family of enzymes are manufactured in the liver and can reflect liver function.
Albumin (ALB)	This protein is made in the liver and can be used to detect liver disease.
Zinc turbidity test (ZZT)	This test is used to determine the chronicity of liver disease.

Table A.2: Comparison of clustering techniques for sequential data

Method	**Strengths and Limits**	**Description**
Relocation Clustering	More scalable than eigenvalue decomposition based techniques; limited to equal length sequences; shows lower performance on time series data.	Procedure begins with and initial clustering, C, with k known $a\ priori$. For each t_i the dissimilarity matrix is calculated and stored to find a clustering C', such that C' is better than C in terms of the generalized Ward criterion.
Agglomerative Hierarchical Clustering	Can works with unequal length sequences but reports poor performance; reveals hierarchical properties of observations; k is not a parameter; does not scale well to long time series.	Clustering begins by placing each object in its own cluster, then clusters are merged into larger clusters, until stop criteria is satisfied. At each step, sum-of-squares variance is computed for all possible mergers, and the smallest value selected.
Divisive Hierarchical Clustering	Can work with unequal length sequences but does not scale well and reports poorer performance; reveals hierarchical properties; k is not a parameter.	Clustering generates a nested hierarchy of similar groups of according to a pairwise distance matrix.
k-Means and Fuzzy c-Means	simple to implement, intuitive and provides good performance when cluster shapes are convex; extends for fuzzy partitioning (c-means); cannot work with unequal length sequences; sensitive to the initialization of cluster means	k-means begins with an initial assignment of cluster means that is iteratively optimized to minimize the objective function, which is typically the total distance between all points from their respective cluster centers.
Self-Organizing Maps (SOM)	Provide good performance, but does not work well with time series of unequal length due to the difficulty involved in defining the dimension of weight vectors.	Sometimes called Kohonen maps, are a class of neural networks with neurons arranged in a low-dimensional structure. Begins by assigning small random values to the weight vectors, w, of the neurons and iteratively updates w until convergence on local estimates.
Bayesian Hierarchical Clustering (BHC)	Like spectral clustering, Bayesian Hierarchical Clustering avoids unnecessary parametric assumptions; doesn't scale and cannot incorporate tree uncertainty	BHC is an extension of agglomerative hierarchical clustering that uses marginal likelihoods of a probabilistic model instead of distance measures for deciding on cluster merges and to control over-fitting.

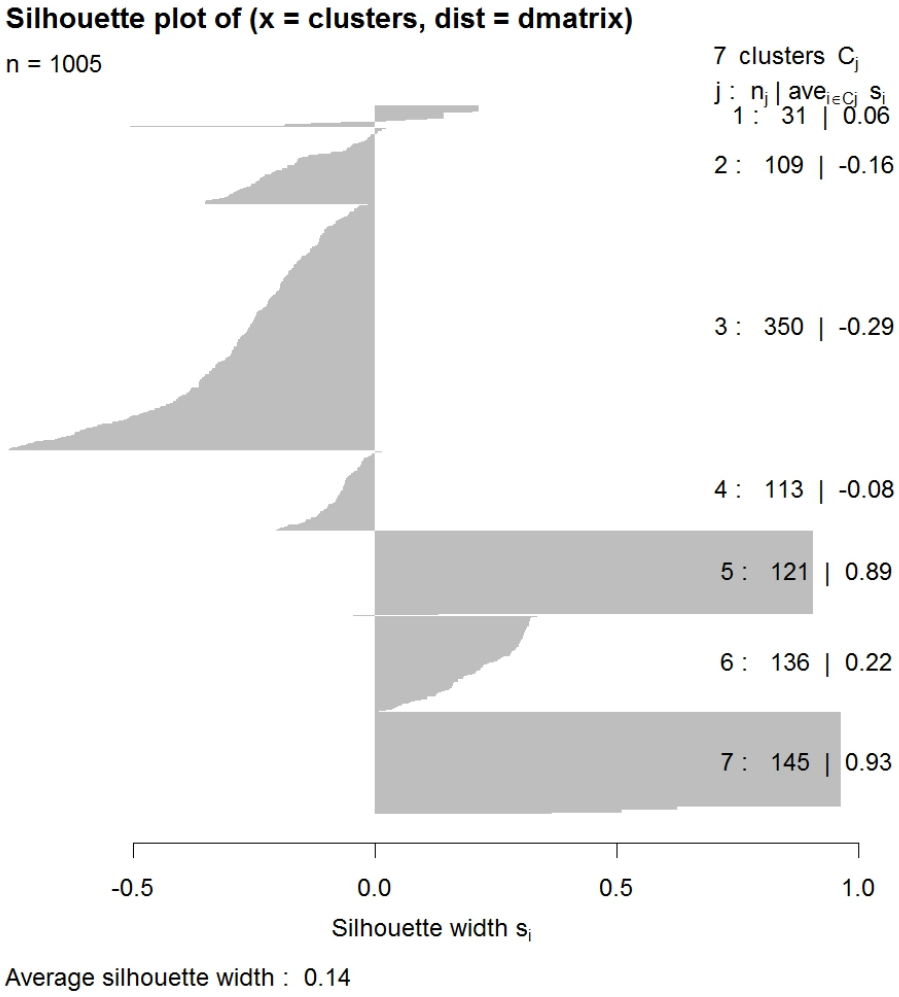

Figure A.1: Silhouette values for CT-SC method where $k = 7$

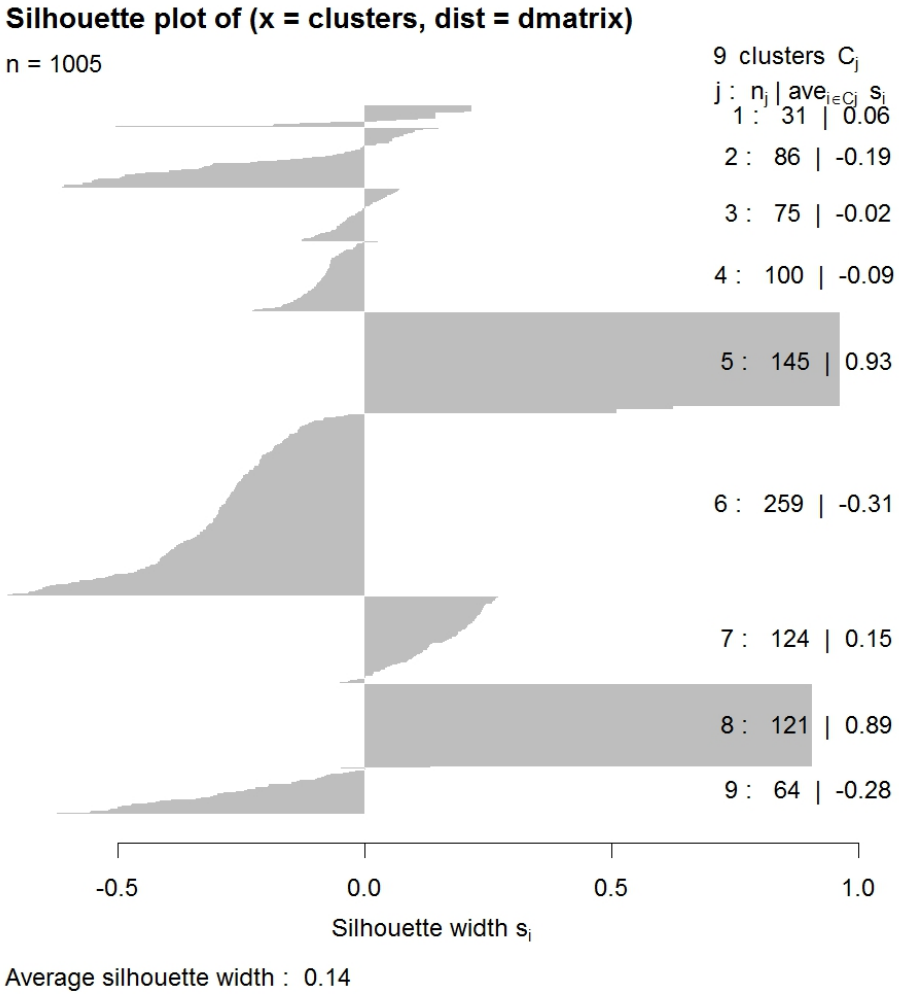

Figure A.2: Silhouette values for CT-SC method where $k = 9$

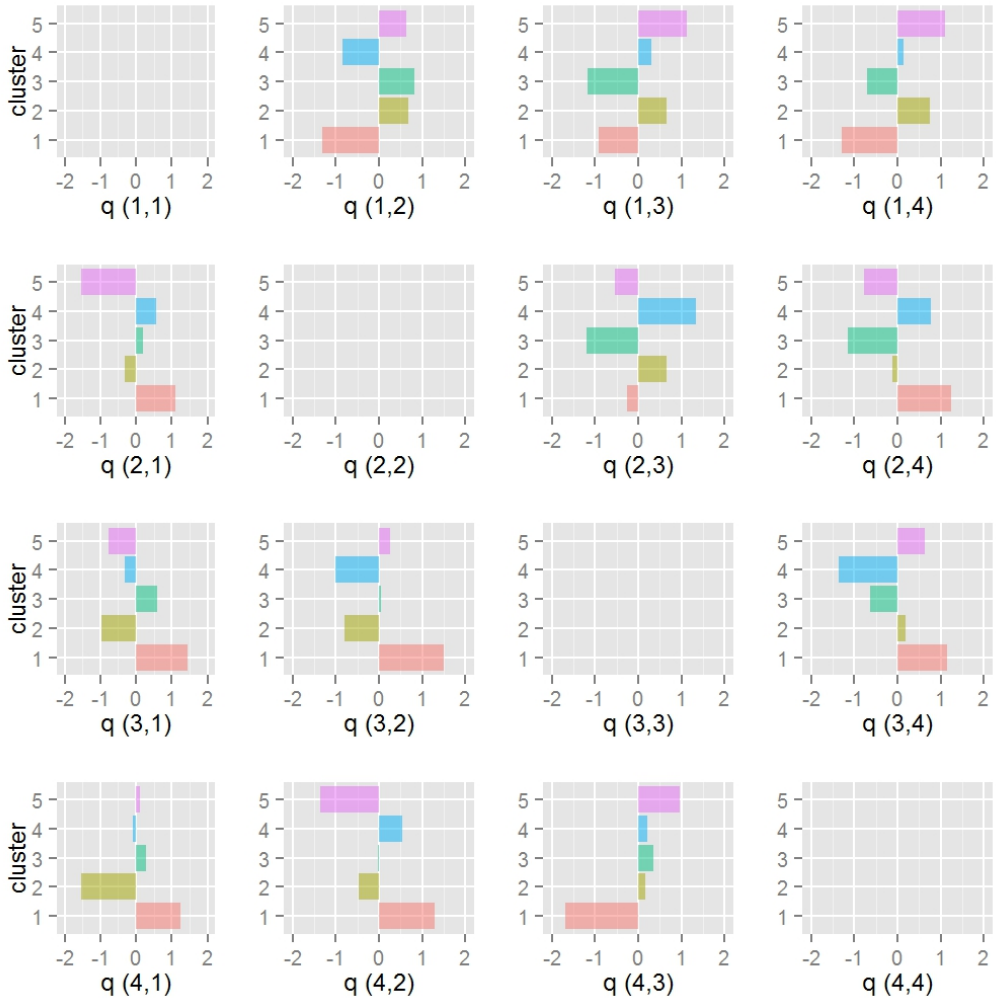

Figure A.3: Characteristic Q matricies by cluster

Bibliography

[1] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

[2] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Prentice-Hall, Englewood Cliffs, N.J., 1976.

[3] A. Bagga and B. Baldwin. Algorithms for scoring coreference chains. In *Proceedings of the* 1^{st} *International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, 1998.

[4] A.J. Bagnall and G.J. Janacek. Clustering time series from ARMA models with clipped data. In *Proceedings of the* 10^{th} *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 49–58, 2004.

[5] R. Bellazzi, C. Larizza, P. Magni, S. Montani, and M. Stefanelli. Intelligent analysis of clinical time series: an application in the diabetes mellitus domain. *Artificial Intelligence in Medicine*, 20, 2000.

[6] M. Berlingerio, F. Bonchi, F. Giannotti, and F. Turini. Mining clinical data with a temporal dimension: A case study. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 429–436, 2007.

[7] D.M. Blei. Variational methods for the Dirichlet process. In *Proceedings of the International Conference on Machine Learning*, 2004.

[8] D.M. Blei and J.D. Lafferty. Dynamic topic models. In *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, pages 113–120, 2006.

[9] D.M. Blei and J.D. Lafferty. Topic models. *Text Mining Theory and Applications*, 3(4-5):1–24, 2009.

[10] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[11] A. Blum. Thoughts on clustering. In *Advances in Neural Information Processing Systems, Workshop on Clustering Theory*, 2009.

[12] L. Castera. Noninvasive methods to assess liver disease in patients with hepatitis B or C. *Gastroenterology*, 142(6):1293–1302, 2012.

[13] E.J. Cooke, R.S. Savage, P Kirk, R Darkins, and D.L. Wild. Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements. *BMC Bioinformatics*, 12(1):399, 2011.

[14] D.R. Cox and H.D. Miller. *The theory of stochastic processes*. Methuen London, 1965.

[15] T.L. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.

[16] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistics Society, Series B*, 39(1):1–38, 1977.

[17] J.D. Dunham. Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):67–75, 1986.

[18] N. Durand and A. Soulet. Emerging overlapping clusters for characterizing the stage of liver fibrosis. In *Proceedings of the* 16^{th} *European Conference on Machine Learning*, 2005.

[19] J.A. Ewald, T.M. Downs, J.P. Cetnar, and W.A. Ricke. Expression microarray meta-analysis identifies genes associated with Ras/MAPK and related pathways in progression of muscle-invasive bladder transition cell carcinoma. *Public Library of Science ONE*, 8(2):e55414, 2013.

[20] Y. Fan and C.R. Shelton. Learning continuous-time social network dynamics. In *Proceedings of the* 25^{th} *on Uncertainty in Artificial Intelligence*, pages 161–168, 2009.

[21] Centers for Disease Control. *National diabetes fact sheet*. U.S. Department of Health and Human Services, 2011.

[22] S. Fruhwirth-Schnatter. Panel data analysis: a survey on model-based clustering of time series. *Advances in Data Analysis and Classification*, 5:251–280, 2011.

[23] D. García-García, E. Parrado-Hernández, and F. Diaz-de Maria. A new distance measure for model-based sequence clustering. *IEEE Transansactions on Pattern Analysis and Machine Intelligence*, 31(7):1325–1331, 2009.

[24] D. García-García, E. Parrado-Hernández, and F. Diaz-de Maria. State-space dynamics distance for clustering sequential data. *Pattern Recognition*, 44:1014–1022, 2011.

[25] R.C. Gentleman, J. F. Lawless, J. C. Lindsey, and P. Yan. Multi-state Markov models for analysing incomplete disease history data with illustrations for HIV disease. *Statistical Medicine*, 13(8):805–821, 1994.

[26] A. Guyon, U. von Luxburg, and R.C. Williamson. Clustering: Science or art? In *Advances in Neural Information Processing Systems, Workshop on Clustering Theory*, 2009.

[27] F.J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51 – 83, 1978.

[28] K.A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the* 22^{nd} *International Conference on Machine Learning*, pages 297–304, 2005.

[29] S. Hirano and S. Tsumoto. Temporal analysis of platelet data in chronic viral hepatitis dataset. In *Proceedings of the* 16^{th} *European Conference on Machine Learning*, 2005.

[30] S. Hirano and S. Tsumoto. Cluster analysis of trajectory data on hospital laboratory examinations. *Proceedings of the AMIA Annual Symposium*, pages 324–328, 2007.

[31] S. Hirano and S. Tsumoto. Unsupervised grouping of trajectory data on laboratory examinations for finding exacerbating cases in chronic diseases. *Proceedings of the* 18^{th} *European Conference on Machine Learning, Workshop on Mining Complex Data*, pages 324–328, 2007.

[32] F. Hoeppner. Time series abstraction methods - a survey. In *GI Jahrestagung*, pages 777–786, 2002.

[33] C. Jackson. Multi-state models for panel data: The msm package for R. *Journal of Statistical Software*, 38(8):1–28, 2011.

[34] T. Jebara, Y. Song, and K. Thadani. Spectral clustering and embedding with hidden markov models. In *Proceedings of the European Conference on Machine Learning*, 2007.

[35] M.I. Jordan. Graphical models. *Statistical Science*, 19, 2003.

[36] D. Klimov, U. Shahar, and M. Taieb-Maimon. Intelligent visualization and exploration of time-oriented data of multiple patients. *Artificial Intelligence in Medicine*, 49(1):11–31, 2010.

[37] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[38] J.M. Lewis. Finding a better k: A psychophysical investigation of clustering. In *Advances in Neural Information Processing Systems, Workshop on Clustering Theory*, 2009.

[39] T.W. Liao. Clustering of time series data a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[40] J. Lin and E. Keogh. Clustering of streaming time series is meaningless. In *In Proceedings of the 8th ACM SIGMOD workshop on Research issues in*, pages 56–65. ACM Press, 2003.

[41] P. Marcello. What is a cluster: Perspectives from game theory. In *Advances in Neural Information Processing Systems, Workshop on Clustering Theory*, 2009.

[42] B. Marlin, D. Kale, R. Khemani, and R. Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the* 2^{nd} *ACM SIGHIT International Health Informatics Symposium*, pages 389–398, 2012.

[43] S.N. Murphy, G.M. Weber, M. Mendis, V. Gainer, H.C. Chueh, S. Churchill, and I.S. Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130, 2010.

[44] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Conference on Advances in Neural Information Processing Systems*, pages 849–856, 2001.

[45] U. Nodelman, C.R. Shelton, and D. Koller. Continuous time Bayesian networks. In *Proceedings of the* 18^{th} *Conference on Uncertainty in Artificial Intelligence*, pages 378–387, 2002.

[46] U. Nodelman, C.R. Shelton, and D. Koller. Learning continuous time Bayesian networks. In *Proceedings of the* 19^{th} *Conference on Uncertainty in Artificial Intelligence)*, pages 451–458, 2003.

[47] P. Orbanz and Y.W. Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. Springer, 2010.

[48] C. Pamminger and S. Frhwirth-Schnatter. Model-based clustering of categorical time series. *Bayesian Analysis*, 5(2):345–368, 2010.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[50] A. Post and J. Harrison. Temporal data mining. *Clinics in Laboratory Medicine*, 28:83–100, 2008.

[51] A. M. Presanis, D. De Angelis, A. Goubar, O. N. Gill, and A. E. Ades. Bayesian evidence synthesis for a transmission dynamic model for HIV among men who have sex with men. *Biostatistics*, 12(4):666–681, Oct 2011.

[52] L.R. Rabiner. Readings in speech recognition. chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann Publishers Inc., 1990.

[53] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.

[54] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

[55] J.C. Salazar, F.A. Schmitt, L. Yu, M.M. Mendiondo, and R.J. Kryscio. Shared random effects analysis of multi-state Markov models: application to a longitudinal study of transitions to dementia. *Statistical Medicine*, 26(3):568–580, 2007.

[56] S. Saria, K. Koller, and A. Penn. Learning individual and population level traits from clinical temporal data. In *Advances in Neural Information Processing Systems, Predictive Models in Personalized Medicine workshop*, 2010.

[57] S. Saria, U. Nodelman, and D. Koller. Reasoning at the right time granularity. In *Proceedings of the* 23^{rd} *Conference on Uncertainty in Artificial Intelligence*, 2007.

[58] S.P. Shah, K.J. Cheung, N.A. Johnson, G. Alain, R.D. Gascoyne, D.E. Horsman, R.T. Ng, and K.P. Murphy. Model-based clustering of array CGH data. *Bioinformatics*, 25:i30–i38, June 2009.

[59] M.J. Shensa. The discrete wavelet transform: wedding the a'trous and Mallat algorithms. *IEEE Transactions on Signal Processing*, 40(10):2464–2482, 1992.

[60] J. Shi and Malik J. Normalized cuts and image segmentation. *IEEE Transactions on Patern Analysis and Machine Learning*, 22(8):888–905, 2000.

[61] P. Smyth. Clustering sequences with hidden Markov models. In *Advances in Neural Information Processing Systems*, pages 648–654, 1997.

[62] P. Smyth. Belief networks, hidden Markov models, and Markov random fields: a unifying view. *Pattern Recognition Letters*, 18:1261–1268, 1998.

[63] P. Smyth, D. Heckerman, and M.. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2):227–269, 1997.

[64] M. Srivastava, P. Khurana, and R. Sugadev. Lung cancer signature biomarkers: tissue specific semantic similarity based clustering of digital differential display data. *BioMed Central Research Notes*, 5:617, 2012.

[65] B. Steckemetz, A. Schliep, B. Knab, and B. Wichern. Model-based clustering with hidden Markov models and its application to financial time-series data. In *Between data science and applied data analysis: University of Mannheim, July 22 - 24, 2002*. Springer, 2003.

[66] M. J. Sweeting, V. T. Farewell, and D. De Angelis. Multi-state Markov models for disease progression in the presence of informative examination times: an application to hepatitis C. *Statistical Medicine*, 29(11):1161–1174, 2010.

[67] S. Tamang and S. Parsons. Using semi-parametric clustering applied to electronic health record time series data. In *Proceedings of the* 17^{th} *ACM SIGKDD Workshop on Data Mining for Medicine and Healthcare*, pages 72–75, 2011.

[68] W. T. Tay, G. A. Evans, L. M. Boykin, and P. J. De Barro. Will the real Bemisia tabaci please stand up? *PLoS ONE*, 7(11):e50550, 2012.

[69] A. C. Titman and L. D. Sharples. Semi-Markov models with phase-type sojourn distributions. *Biometrics*, 66(3):742–752, 2010.

[70] S. Tsumoto and S. Hirano. Multidimensional temporal mining in clinical data. In *Proceedings of the* $2nd$ *ACM SIGHIT International Health Informatics Symposium*, pages 563–572, 2012.

[71] H. TuBao, N. DucDung, K. Saori, and N. TrongDung. Extracting knowledge from hepatitits data with temporal abstraction. In *Proceedings of the* 13^{th} *European Conference on Machine Learning*, 2002.

[72] Z. Uhry, G. Hedelin, M. Colonna, B. Asselain, P. Arveux, A. Rogel, C. Exbrayat, C. Guldenfels, I. Courtial, P. Soler-Michel, F. Molinie, D. Eilstein, and S. W. Duffy. Multi-state Markov models in cancer screening evaluation: a brief review and case study. *Statistical Methods in Medical Research*, 19(5):463–486, 2010.

[73] C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*, 2012.

[74] S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SIAM International Conference on Data Mining*, 2005.

[75] E.P. Xing, K.A. Sohn, M.I. Jordan, and Y.W. Teh. Bayesian multi-population haplotype inference via a hierarchical Dirichlet process mixture. In *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, volume 23, 2006.

[76] D. Yan, L. Huang, and M. Jordan. Fast approximate spectral clustering. Technical report, EECS Department, University of California, Berkeley, 2009.

[77] J. Yin and Q. Yang. Integrating hidden Markov models and spectral analysis for sensory time series clustering. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 506–513, 2005.

[78] Y. Zeng and J. Garcia-Frias. A novel HMM-based clustering algorithm for the analysis of gene expression time-course data. *Computational Statistics and Data Analysis*, 50:2472–2494, 2006.