TIME SERIES MODELING OF IRREGULARLY SAMPLED MULTIVARIATE CLINICAL DATA

by

Zitao Liu

B.Eng. in Software Engineering, Wuhan University, 2010

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of Arts and Sciences in partial

fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Zitao Liu

It was defended on

June 2nd 2016

and approved by

Milos Hauskrecht, PhD, Professor, University of Pittsburgh

Rebecca Hwa, PhD, Associate Professor, University of Pittsburgh

Jingtao Wang, PhD, Assistant Professor, University of Pittsburgh

Christos Faloutsos, PhD, Professor, Carnegie Mellon University

Dissertation Director: Milos Hauskrecht, PhD, Professor, University of Pittsburgh

TIME SERIES MODELING OF IRREGULARLY SAMPLED MULTIVARIATE CLINICAL DATA

Zitao Liu, PhD

University of Pittsburgh, 2016

Building of an accurate predictive model of clinical time series for a patient is critical for understanding of the patient condition, its dynamics, and optimal patient management. Unfortunately, this process is challenging because of: (1) *multivariate behaviors*: the real-world dynamics is multivariate and it is better described by multivariate time series (MTS); (2) *irregular samples*: sequential observations are collected at different times, and the time elapsed between two consecutive observations may vary; and (3) *patient variability*: clinical MTS vary from patient to patient and an individual patient may exhibit short-term variability reflecting the different events affecting the care and patient state.

In this dissertation, we investigate the different ways of developing and refining forecasting models from the irregularly sampled clinical MTS data collection. First, we focus on the refinements of a popular model for MTS analysis: the linear dynamical system (LDS) (a.k.a Kalman filter) and its application to MTS forecasting. We propose (1) a regularized LDS learning framework which automatically shuts down LDSs' spurious and unnecessary dimensions, and consequently, prevents the overfitting problem given a small amount of data; and (2) a generalized LDS learning framework via matrix factorization, which allows various constraints can be easily incorporated to guide the learning process. Second, we study ways of modeling irregularly sampled univariate clinical time series. We develop a new two-layer hierarchical dynamical system model for irregularly sampled clinical time series prediction. We demonstrate that our new system adapts better to irregular samples and it supports more accurate predictions. Finally, we propose, develop and experiment with two personalized

forecasting frameworks for modeling and predicting clinical MTS of an individual patient. The first approach relies on model adaptation techniques. It calibrates the population based model's predictions with patient specific residual models, which are learned from the difference between the patient observations and the population based model's predictions. The second framework relies on adaptive model selection strategies to combine advantages of the population based, patient specific and short-term individualized predictive models. We demonstrate the benefits and advantages of the aforementioned frameworks on synthetic data sets, public time series data sets and clinical data extracted from EHRs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

PREFACE

I spent six fabulous years in Pittsburgh. I would like to thank the people who accompanied me and made my journey of pursing Ph.D. possible and pleasurable.

First of all, I want to sincerely thank my research advisor Dr. Milos Hauskrecht. This dissertation would be impossible to complete without the help from Milos. He not only taught me the advanced machine learning and data mining techniques but guided me through the scientific research process. His high professional standards and rigorous attentions to details helped me solve real-world clinical problems, publish top conference and journal papers, obtain the Andrew Mellon Predoctoral Fellowship for the school year 2015-2016 and gradually shape my logical thinking and problem solving skills. Thank you, Milos!

I would also like to thank my Ph.D. committee members, Dr. Rebecca Hwa, Dr. Jingtao Wang and Dr. Christos Faloutsos for their valuable suggestions and insightful discussions during my proposal and dissertation defenses. I want to thank our post-doc Lei Wu, with whom I worked during my first year of Ph.D. research and also other members of Milos' machine learning group: Shuguang Wang, Quang Nguyen, Dave Krebs, Eric Heim, Charmgil Hong, Salim Malakouti, Siqi Liu and Zhipeng Luo.

I was privileged to work as an intern in Google Inc., eBay Research Lab, Yahoo! Labs and Alibaba Group with amazing colleagues and mentors: Laura Werner from Google Inc; Nish Parikh, Gyanit Singh, and Neel Sundaresan from eBay Research Lab; Chris Yan Yan, Jimmy Jian Yang, Pengyuan Wang, Wei Sun, James Li, and Zheng Wen from Yahoo! Labs; Jian Xue, Shenghuo Zhu, Sen Yang, Jian Tan and Rong Jin from Alibaba Group. The internship experience taught me both the research & development paradigm in the industry. I learned how to quickly adapt in new environments and how to openly communicate with others.

I am very grateful to have so many wonderful friends throughout my educational odyssey. I need to mention Xiangmin Fan, Rui Wu, Lingjia Deng, Jiannan Ouyang, Ka Wai Yung, Wencan Luo, Lanfei Shi, Mengmeng Li, Huichao Xue, Wenting Xiong, Yingze Wang, Yao Sun and Yu Du for marvelous times we spent together in Pittsburgh. I made great friends at both Pitt and CMU, with whom I would like to keep in touch including Lailuyun Xu,

Rongqian Ma, Shou Li, Shicheng Lv, Yangzhan Yang, Haifeng Xu, Xuelian Long, Bo Luan, Yingjun Su, Yun Wang, Rui Liu, Guimin Lin, Guangyu Xia, Xi Chen and others.

I would also like to thank colleagues and friends who I met during academic conferences and internships. We often exchanged research ideas interdisciplinarily, which broadened my sight and encouraged me to move forward. In particular, I would like to thank Huan Liu, Jieping Ye, Hanghang Tong, Fei Wang, Jiliang Tang, Xia Hu, Bing Hu, Chen-Yu Lee, Zixuan Wang and Shumo Chu.

Lastly, it is most important to thank my parents Tiejun and Lihua, whose unlimited patience, love, encouragement and support helped assure that I would complete this most difficult journey.

Thank you all!

1.0 INTRODUCTION

1.1 MOTIVATION

Recent advances in data collection, data storage and information technologies have resulted in enormous collections of time series data in various aspects of our everyday life, such as sequences of weather temperature measurements reflecting the changes of the climate, clinical observations showing the health conditions of patients, or stock price series indicating the dependences and variations of the capital market. The emergence and availability of time series data provide us with a unique opportunity to gain novel insights into the processes generating the data and let us build models we can utilize for making future decisions. For example, understanding how the supply and demand change over time provides better strategies for supply chain and inventory management and planning [Aburto and Weber, 2007]. Time series analysis is the field of research that attempts to analyze these rich time series data in order to extract their meaningful statistics and infer their future behaviors.

As one important type of time series data, clinical multivariate time series (MTS) record the values of many clinical variables over time. In general these variables include various laboratory tests, physiological measurements, or treatments and are highly related to patient condition and outcomes. With the recent development of advanced data technology, large temporal electronic health record repositories emerge and become highly available. They reflect different responses and behaviors of individual patients whether this is in context of chronic or acute clinical condition, or their combination. Clinical time series data provides us with a unique opportunity to gain novel insights into the dynamics of the patient state, dynamics of the disease, or efficacy of its treatments.

1.2 TIME SERIES RELATED TASKS

With the emergence and availability of the huge amount of time series data, various of time series analysis tasks are researched and studied largely in any domain of applied science and engineering, which involves temporal measurements, such as econometrics [Zellner and Palm, 1974], signal processing [Cohen, 1995], mathematical finance [Taylor, 2007]. In the following, we briefly list several major types of time series analysis tasks which are appropriate for different purposes.

- **Time series classification.** Time series classification is to build a classification model based on labeled time series and then use the model to predict the label of unlabeled time series. There are many practical applications of time series classification, such as classifying electroencephalography signals [Xu et al., 2004], personal motion trajectories [Shotton et al., 2013], speech recognition [Rabiner, 1989] and more.
- **Time series segmentation.** In time series segmentation, the goal is to split time series data into sequences of segments by identifying the segment boundary points, and to characterize the dynamical properties associated with each segment. A typical application of time series segmentation is speaker diarization, in which an audio signal is partitioned into several pieces according to who is speaking at what times [Tranter and Reynolds, 2006].
- **Time series outlier detection.** Time series outlier detection is similar to event detection but focuses on finding the observation that appears to deviate markedly from other observations in the time series. Outliers may occur due to various reasons, such as machine malfunctioning, networking disturbances, or human inappropriate operations. A practical application scenario of time series outlier detection is that in clinical decision support systems, temporal outlier detection algorithms can identify unusual clinical management patterns in individual patients and raise alarms if wrong treatments are detected [Hauskrecht et al., 2013].
- **Temporal pattern abstraction.** Temporal pattern abstractions aim to convert time series variables into time-interval sequences of abstract states or temporal logic to represent temporal interactions among multiple states and define and construct temporal

patterns from these abstract representations. Temporal patterns provide appealing abstractions of the original time series and improve the performance for other time series tasks like time series classification [Batal et al., 2011], event detection [Batal et al., 2012].

- **Temporal dependence/causal discovery.** Uncovering the temporal dependent or causal relationship among MTS data is a major task in data mining, which easily finds applications in many domains. For example, in the climatology, the causal relationships between climate time series variables help identify the factors that impact the climate patterns of certain regions. In social networks, the temporal dependence improves the pattern identification of influence among users and how topics activate or suppress each other [Bahadori and Liu, 2013, Cheng et al., 2014].
- **Time series forecasting.** Time series forecasting is the use of a model to predict future values based on previously observed values, which has extensive applications in many domains. For example, in the clinical domain, accurate predicting the patients' lab tests values from previous measurements observed by physicians will help detect an adverse event or a disease in its early stages, thus allowing clinicians to identify the most effective treatment [Osorio et al., 1998, Richman and Moorman, 2000, Liu and Hauskrecht, 2015a]. In this dissertation, we mainly focus on the task of time series forecasting, especially the forecasting problems in clinical domain. With a wide adoption and availability of electronic health records (EHRs), the development of forecasting models of clinical MTS and tools for their analysis is becoming increasingly important for meaningful applications of EHRs in computer-based patient monitoring, adverse event detection, and improved patient management [Bellazzi et al., 2000, Clifton et al., 2013, Lasko et al., 2013, Liu and Hauskrecht, 2013, Liu et al., 2013, Schulam et al., 2015, Ghassemi et al., 2015, Durichen et al., 2015].

1.3 CHALLENGES

A large spectrum of temporal models have been developed and successfully applied in time series analysis [Du Preez and Witt, 2003, Ljung and Glad, 1994] and many of them have

been applied recently to support predictions or inferences on clinical and biomedical data. Example applications include detection and early warning of patient deteriorations [Clifton et al., 2013], discovery of phenotypes and endotypes [Lasko et al., 2013,Schulam et al., 2015], assessment of severity of patient's illness [Ghassemi et al., 2015], models for active motion compensation to precisely radiate tumors in the liver or lung [Durichen et al., 2015]. However, none of the aforementioned methods can be directly applied into forecasting problems in real-world clinical MTS data. Building forecasting models from EHRs encounters numerous challenges due to three practical characteristics of real-world clinical time series data: *multivariate behaviors*, *irregular samples* and *patient variability*, which make conventional methods inadequate to handle them.

1.3.1 Multivariate Behaviors

A univariate time series is a sequence of measurements of the *same* variable collected over time while a multivariate time series (MTS) consists of sequences of measurements of *multiple* variables over time and exhibits complex temporal behaviors. MTS data appear in a wide variety of fields, such as health care [Sacchi et al., 2007, Hauskrecht et al., 2010a, Ho et al., 2003], economics [Kling and Bessler, 1985], motion capture [Li et al., 2009], astronomy [Scargle, 1982], weather forecasting [Gneiting and Raftery, 2005], earthquake prediction [Scholz et al., 1973] and many more. MTS not only show the temporal dependent behaviors within each time series but exhibit interactions and co-movements among different time series. For example, in economics, forecasting consumer price index usually depends on money supply, index of industrial production and treasury bill rates collectively [Kling and Bessler, 1985]. In clinical domain, a large number of clinical variables might be measured for a single patient (e.g., white blood cell counts, creatinine values, cholesterol levels, etc.) [Batal et al., 2012].

A large number of hidden variable models are proposed in past decades to model such complex dependent MTS, such as hidden Markov models (HMM) [MacDonald and Zucchini, 1997], factorial HMM [Ghahramani and Jordan, 1997], hierarchical Bayesian models [Berliner, 1996], Markov switching models [McCulloch and Tsay, 1994, Kim, 1994]. Hidden variables empower the models to capture more variabilities in the MTS and let human

knowledge easily be incorporated in the modeling process. However, since the observational sequences in MTS data may exhibit strong interactions and co-movements, given the MTS sequences, it is difficult to seek the intrinsic dimensionality of the hidden variables. Open questions arise such as *how many hidden variables are needed to sufficiently represent the MTS well?*, *what is the compact representations of the observation sequences?* Furthermore, after introducing the hidden variables, it becomes challenging to incorporate constraints in the model learning process to achieve desired properties, such as smoothness, stability, etc. Questions emerge such as *Can we easily guide the learning process by adding constraints?*

1.3.2 Irregular Samples

We say that the time series is regularly sampled if the time elapsed between consecutive observations is uniform (the same for all pairs of consecutive observations), while the irregularly sampled time series means sequential observations are collected at different times, and the time elapsed between two consecutive observations may vary [Adorf, 1995].

Usually we obtain regularly sampled time series through sensor devices which regularly collect observations at some fixed sampling frequency. For example, due to the advances of health care sensor technologies, we can easily record the regularly sampled electrocardiogram (ECG) and electroencephalogram (EEG) signals (depicted in Figure 1). In the climatology, weather stations set climate sensors to collect the outside temperature, wind speed, humidity at regularly sampled time stamps.

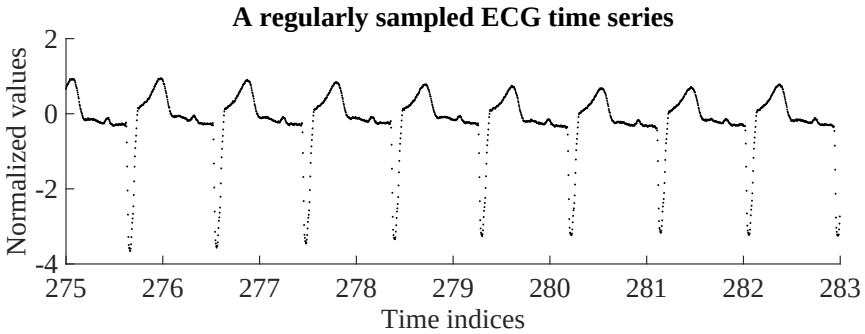


Figure 1: A regularly sampled ECG time series fragment.

However, in many situations we observe irregularly sampled time series, which is very different from typical regularly sampled time series domains. For example, in clinical domain, the observations are obtained whenever a patient visits a healthcare facility and the time intervals between consecutive visits tend to vary greatly. Even during a patient's hospitalization, there is no guarantee that the physician can order lab tests regularly. An irregularly sampled mean corpuscular hemoglobin concentration (MCHC) lab test time series from a patient is shown in Figure 2.

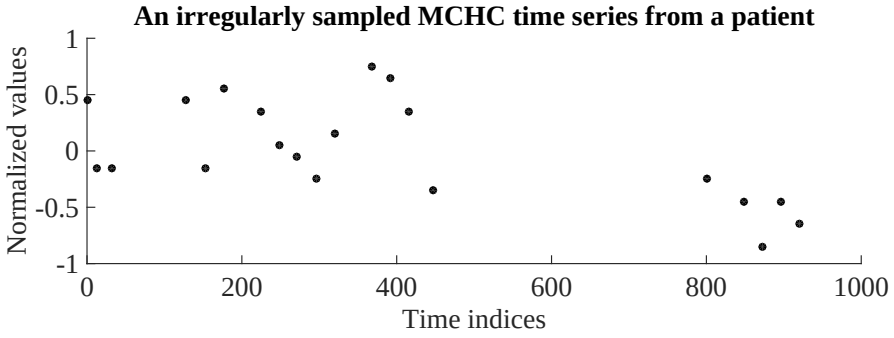


Figure 2: An irregularly sampled MCHC lab test time series.

This irregularly sampled data preclude the applications of a large class of time series modeling techniques that require regularly sampled observations. Modeling irregular sampled questions gives rise to numerous important questions like *Can we still use existing discrete time models to model the irregularly sampled time series?*, *Is it possible to model the irregular sampled data directly?*

1.3.3 Patient Variability

Clinical MTS exhibits large patient variability. First, the number of observations in each patient sequence is limited and the duration they span may vary a lot from patient to patient. As we discussed in Section 1.3.2, nowadays we can easily obtain a long-span time series via sensor devices by either increasing the sampling frequency or keeping it recording for a longer period of time. However, compared to the long-span time series data, patients are usually hospitalized for short periods of time (often less than two weeks), which produces

relatively short-span clinical sequences (often less than 50) [Liu et al., 2013]. Second, within each patient specific clinical MTS sequence, values of various laboratory tests, physiological measurements, or treatments are all recorded. They reflect different responses and behaviors of individual patients and contain lots of short-term variability due to different causes [Schulam et al., 2015]. For example, the blood tests may be affected by events like infection, bleeding, transfusion, or a particular medication treatment.

All such patient variability poses two hard modeling problems of supporting accurate predictions for each patient. First, given a complex length-varying MTS collection, *how can we learn a population based forecasting model without overfitting to such short-span temporal data?* Second, patient-to-patient variability is typically large and population based models derived or learned from many different patients are unable to capture short-term variability in each individual patient. Given a patient specific prediction task, *how can we adapt the population based model to provide accurate personalized predictions?*

1.4 CONTRIBUTIONS

In this dissertation we focus on the time series forecasting of clinical data with large patient variability (Section 1.3.3). To better understand the forecasting challenges and have a clearer overview of completed work discussed in this dissertation, we introduce four categories of forecasting problems formed by the intersection of two of the three characteristics discussed previously (depicted in Figure 3). Categories are defined by whether they consider univariate or multivariate time series and whether they consider regularly or irregularly sampled data. Below we explain the corresponding problem in each category and highlight our contribution in those categories.

Forecasting regularly sampled univariate time series is the simplest case depicted in the top left in Figure 3. Observations within each time series are obtained at a fixed sampling frequency and forecasting is conducted individually. Many existing forecasting methods can be applied to such time series data, such as ARIMA [Box and Pierce, 1970, Makridakis and Hibon, 1997], exponential smoothing [Gardner, 1985], polynomial regression [Theil, 1992].

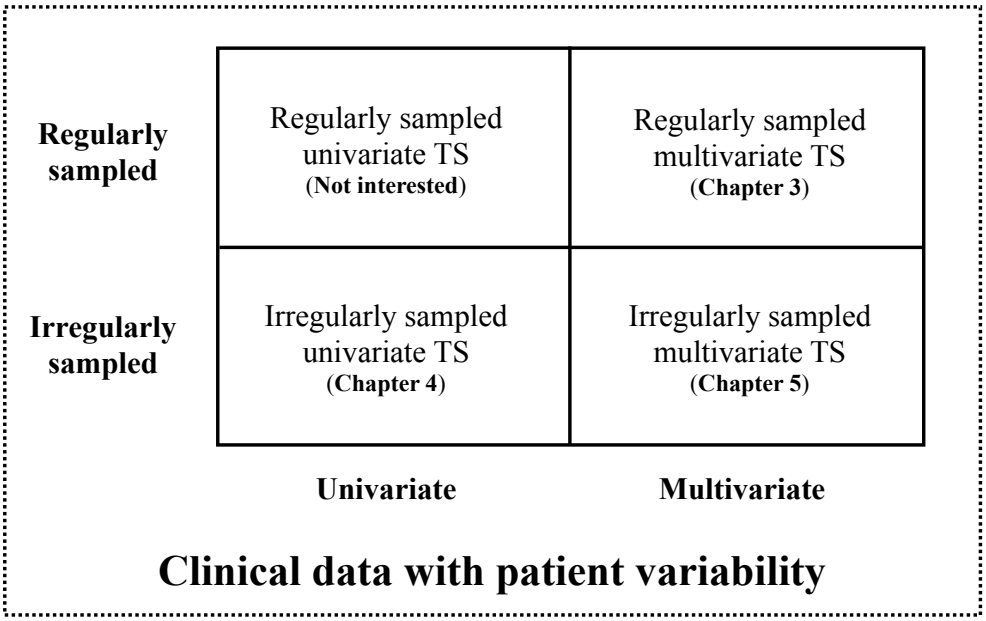


Figure 3: The four categories of clinical time series forecasting problems.

Furthermore, methods from other categories generally can be applied to model regularly sampled univariate time series with few or no modifications. Therefore, in this dissertation, we will focus more on forecasting problems in other categories in Figure 3.

In Chapter 3, we develop two frameworks to learn temporal models from regularly sampled multivariate time series data. Our work focuses on the refinements of a popular model for MTS analysis: the linear dynamical system (LDS) (described in Section 2.2.1). More specifically, the first framework, regularized linear dynamical system, aims to automatically identify the intrinsic dimensionality of the hidden state space of LDS given a limited number of MTS data and consequently, prevents the overfitting problem and performs more accurate forecasting. We develop a maximum a posteriori learning framework to learn the regularized LDS models from a small amount of complex MTS data. In our learning framework, we choose parameter priors to bias the model towards a low-rank solution. We propose three strategies for choosing the parameter priors that lead to three instances of our regularized LDS. The second framework is developed for learning LDS models from a collection

of MTS data based on matrix factorization, which is different from traditional EM learning and spectral learning algorithms. In our generalized LDS learning framework, each MTS sequence is factorized as a product of a shared emission matrix and a sequence-specific (hidden) state dynamics, where an individual hidden state sequence is represented with the help of a shared transition matrix. One advantage of our generalized framework is that various types of constraints can be easily incorporated into the learning process. Furthermore, we propose a novel temporal smoothing regularization approach for learning the LDS model, which stabilizes the model, its learning algorithm and predictions it makes. We demonstrate the benefits of our methods on a number of time series data sets.

In Chapter 4, we focus on challenges in forecasting from irregularly sampled data. Since observations that form clinical time series are usually are made initiated (ordered) by a clinician, no fixed sampling frequency can be guaranteed. In this chapter, we propose and develop a novel hierarchical dynamical system framework for modeling clinical time series that combines advantages of the two temporal modeling approaches: the linear dynamical system and the Gaussian process(GP). We model the irregularly sampled clinical time series by using multiple GP sequences in the lower level of our hierarchical framework and capture the transitions between GPs by utilizing the LDS. The experiments are conducted on the complete blood count (CBC) panel data of 1000 post-surgical cardiac patients during their hospitalization. We show that our model outperforms multiple existing models in terms of the mean absolute prediction error and the absolute percentage error. Our method achieved a 3.13% average prediction accuracy improvement on ten CBC lab time series when it was compared against the best performing baseline. A 5.25% average accuracy improvement was observed when only short-term predictions were considered.

In Chapter 5, we develop and study personalization strategies for building improved forecasting models that better mimic patient specific behaviors from irregularly sampled multivariate clinical data. This problem is rather challenging due to the characteristics of clinical MTS and the computational and modeling trade-offs arising from them. Briefly, when the time series of past observations for the patient are short, it may be hard to learn a patient specific model, and the population based model may be a better option. On the other hand, when the observed data for the target patient are sufficiently long, a patient

specific time series model may better reflect the future behavior. In this dissertation, we develop two approaches to address the above issues. Our first approach builds upon model adaptation. It first learns a population based model from all the available patients and then re-calibrates the population based model into personalized models through patient specific residual models. The patient specific residual models are learned from multivariate residual time series, which is the difference between the patient observations and the population based model's predictions. The second approach relies on adaptive model selection strategies to combine advantages of the population based, patient specific and short-term individualized predictive models. We build a pool of high quality forecasting models for clinical MTS and their variety assures the coverage of many different modes and behaviors. Our approach is designed to pick the most appropriate predictive model for each patient at every time stamp. Both proposed approaches are evaluated on a real-world clinical time series data set. The results demonstrate that our approaches are superior on the prediction tasks for irregularly sampled multivariate clinical time series, and they outperform pure population based and patient specific models, as well as, other patient specific model adaptation strategies in terms of prediction accuracy.

1.5 OUTLINE

The rest of this dissertation is organized as follows: Chapter 2 introduces the notation to be used in subsequent chapters and provides a review of the basics of time series models and the personalized predictive methods to guide the precision medicine. Chapters 3, 4 and 5 present the main contributions of this dissertation. Finally, Chapter 6 summarizes the contributions of this dissertation and discusses avenues of future work.

Finally, I would like to note that parts of this dissertation have been previously published in the following conferences and journal: SDM 2013 [Liu et al., 2013], AIME 2013 [Liu and Hauskrecht, 2013], AAAI 2015 [Liu and Hauskrecht, 2015b], AAAI 2016 [Liu and Hauskrecht, 2016a], SDM 2016 [Liu and Hauskrecht, 2016b] and the Artificial Intelligence in Medicine [Liu and Hauskrecht, 2015a].

2.0 BACKGROUND

In this section, we first define notation used in this dissertation. Then, we review the basics of the time series models, in particular, (1) the linear dynamical system, which is a discrete time model used commonly to represent regularly sampled time series data (Section 2.2.1); (2) the Gaussian process model that works with continuous real-valued quantities and lets us model functions of continuous time (Section 2.2.2); and (3) the multi-task Gaussian process model that extends the standard Gaussian process to model the multivariate dependence within multivariate time series (Section 2.2.3). After that, we review various techniques used in biomedical and clinical domains to build predictive patient specific models. These techniques are proposed to entail the delivery of individually tailored clinical decision supports that leverage information about each person's unique characteristics, which can be summarized into three categories: subpopulation models (Section 2.3.1), model adaptation (Section 2.3.2) and adaptive model selection (Section 2.3.3).

2.1 NOTATION

In the following, we introduce the notation that be used in the subsequent.

- We denote time series data $\mathcal{D}$ as a collection of N multivariate time series sequences $\mathcal{D} = \{\mathbf{Y}^1, \mathbf{Y}^2, \cdots, \mathbf{Y}^N\}$. Each $\mathbf{Y}^l$ consists of a sequence of T_l past observation-time pairs $(\mathbf{y}_i^l, t_i^l)$, i.e., $\mathbf{Y}^l = \{(\mathbf{y}_i^l, t_i^l)_{i=1}^{T_l}\}$, such that T_l is the number of past observations for sequence l, $0 < t_i < t_{i+1}$, and $\mathbf{y}_i^l$ is a n-dimensional observation vector made at time (t_i). n is the number of clinical variables in the MTS.

- Let $\mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$ be a multivariate normal distribution with the mean vector $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$. Let $\mathbb{E}_{\mathbf{z}}[f(\cdot)]$ denote the expected value of $f(\cdot)$ with respect to $\mathbf{z}$.

- Special norms used throughout this work include: $\|\cdot\|_F$, $\|\cdot\|_*$, $\|\cdot\|_2$ and $\|\cdot\|_1$ which is the matrix Frobenius norm, the matrix nuclear norm, the vector Euclidean norm and the vector/matrix ℓ_1 norm.

- For both vectors and matrices, the superscript $\top$ denotes the transpose. vec$(\cdot)$ denotes the vector form of a matrix; and $\otimes$ represents the Kronecker product. *Tr* is the *trace* operator and I_d is the $d \times d$ identity matrix.

For the sake of notational brevity, we omit the explicit sample index ("l") and describe our methods by using a MTS sample for the rest of this section. However, it is worth to note that methods we developed can be applied to data of multiple time series samples with few or no modifications.

2.2 TIME SERIES MODELS

A large spectrum of models have been developed and successfully applied in time series modeling and forecasting [Hamilton, 1994], such as ARIMA [Box and Pierce, 1970, Makridakis and Hibon, 1997], exponential smoothing [Gardner, 1985], etc. However, the majority of existing models are focused on regularly sampled univariate time series. In the following, we first review the basics of the linear dynamical system, which is used commonly to represent multivariate time series data (Section 2.2.1). Then, we review the basics of the Gaussian process model that works with continuous real-valued quantities and lets us model functions of continuous time (Section 2.2.2). After that, we introduce the multi-task Gaussian process model, which is an extension of Gaussian process model for multivariate time series (Section 2.2.3).

2.2.1 Linear Dynamical System

The linear dynamical system (LDS) is a classical and widely used model for real-valued sequence analysis [Kalman, 1963], that is applicable to many real-world domains, such as engineering, astronautics, bioinformatics, economics [Lunze, 1994,Liu and Hauskrecht, 2013]. This is due to its relative simplicity, mathematically predictable behavior, and the fact that exact inference and predictions for the model can be done efficiently [Martens, 2010].

The LDS is an MTS model that represents observation sequences indirectly with the help of hidden states. Similarly $\mathbf{y}_i$ and $\mathbf{Y}$ we introduced in Section 2.1, let $\mathbf{z}_i$ be a $d \times 1$ vector representing the values of d dimensional hidden states at time stamp t_i corresponding to $\mathbf{y}_i$ and denote $\mathbf{Z}$ as a $d \times T$ matrix representing the entire values of hidden states along the time span T. The LDS model is a discrete time model which assumes all the time stamps within a sequence are evenly spaced, i.e., $t_{i+1} - t_i = \Phi$ and Φ is the constant representing the fixed length of time interval. The LDS models the dynamics of these sequences in terms of the state transition probability $p(\mathbf{z}_i|\mathbf{z}_{i-1})$, and state-observation probability $p(\mathbf{y}_i|\mathbf{z}_i)$. These probabilities are modeled using the following equations:

$$\mathbf{z}_i = A\mathbf{z}_{i-1} + \boldsymbol{\epsilon}_i \tag{2.1}$$

$$\mathbf{y}_i = C\mathbf{z}_i + \boldsymbol{\zeta}_i \tag{2.2}$$

where the transitions among the current and previous hidden states are linear and captured in terms of a $d \times d$ transition matrix A. The stochastic component of the transition, ϵ_i, is modeled by a zero-mean Gaussian noise $\epsilon_i \sim \mathcal{N}(\mathbf{0}, Q)$ with a $d \times 1$ zero mean vector and a $d \times d$ covariance matrix Q. The observations sequence is derived from the hidden states sequence. The dependencies in between the two are linear and modeled using an $n \times d$ emission matrix C. A zero mean Gaussian noise $\boldsymbol{\zeta}_i \sim \mathcal{N}(\mathbf{0}, R)$ models the stochastic relation in between the states and observations. In addition to A, C, Q, R, the LDS is defined by the initial state distribution for $\mathbf{z}_1$ with mean $\boldsymbol{\xi}$ and covariance matrix Ψ, i.e., $\mathbf{z}_1 \sim \mathcal{N}(\boldsymbol{\xi}, \Psi)$. The complete set of the LDS parameters is $\Lambda = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$. The graphical representation of the LDS is shown in Figure 4.

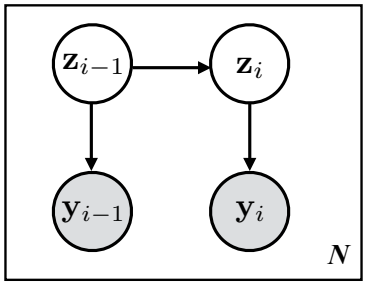


Figure 4: The graphical representation of the LDS. Shaded nodes $\mathbf{y}_t$ and $\mathbf{y}_{t-1}$ denote observation made at current and previous time steps. Unshaded nodes $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$ denote the corresponding hidden states. The links represent dependences among the observations and hidden states. The plate represents a collection of N sequences.

2.2.1.1 Applications The LDS model is a powerful tool in the analysis of the evolution of a dynamical model in time and is commonly used time series model for real-world engineering and financial applications [Isard and Blake, 1998, Kazemi et al., 2008, Victor and Alberto, 2011, Rogers et al., 2013]. In the following, we describe two important applications of the LDS models, *Visual Tracking* and *Biomedical Signal Processing*.

Visual Tracking The LDS models show numerous successful applications in visual tracking (a.k.a, object tracking), which is the problem of estimating the positions (coordinates) and other relevant information of moving objects from a collection of noisy observations [Lee et al., 1995, Isard and Blake, 1998, Funk, 2003, Li et al., 2004, Weng et al., 2006]. In visual tracking, the LDS is robust to the noise caused by rotation, illumination changes, occlusions, etc. and the time update step and measure update step in the *Kalman filtering* algorithm (see Appendix A) is able to filter out the noise from the signal measurements while retain the true trajectories (state sequences).

Biomedical Signal Processing The LDS models are widely used in tasks of analyzing biomedical signals, such as electroencephalogram (EEG), electrocardiogram (ECG) [Georgiadis et al., 2005, Georgiadis et al., 2007, Khan and Dutt, 2007, Kazemi et al., 2008, Sayadi and Shamsollahi, 2008]. Examples like Kazemi et al. [Kazemi et al., 2008] utilize Kalman filtering algorithm to remove the periodic noises (such as electricity grid induced noises)

from ECG signals. Georgiadis et al. [Georgiadis et al., 2005, Georgiadis et al., 2007] propose a Kalman filter based approach to dynamically estimate the event related potentials, which are the voltage changes of brain electric activity due to stimulation. Sayadi et al. [Sayadi and Shamsollahi, 2008] build a modified extended Kalman filter structure to conduct the ECG signals denoising and compression. Khan et al. [Khan and Dutt, 2007] use the hidden state estimates of LDS models to detect event-related desynchronization and synchronization, which are used to describe the decrease and increase in activity in an EEG signal.

2.2.1.2 Learning Linear Dynamical Systems While in some LDS applications the model parameters are known a priori, in the majority of real-world applications the model parameters are unknown, and we need to learn them from MTS data. This can be done using standard LDS learning approaches such as the Expectation-Maximization (EM) [Ghahramani and Hinton, 1996] or spectral learning algorithms [Katayama, 2005, Van Overschee and De Moor, 1996, Doretto et al., 2003].

Expectation-Maximization The EM algorithm is an iterative procedure for finding model parameters that maximizes the likelihood of observations in the presence of hidden variables. In practice, instead of maximizing the data likelihood directly, EM algorithm usually maximizes a $\mathcal{Q}$ function, which is the expectation of the joint probability of both observed and hidden variables with respect to the distribution of hidden variables. The $\mathcal{Q}$ function is a lower bound of the true data likelihood and maximizing it will improve the data likelihood. Under the setting of learning standard LDS defined by eq.(2.1) and eq.(2.2), the $\mathcal{Q}$ function is defined as follows:

$$\mathcal{Q} = \mathbb{E}_{\mathbf{Z}}\Big[\log p(\mathbf{Z}, \mathbf{Y})\Big] = \mathbb{E}_{\mathbf{Z}}\Big[\log p(\mathbf{z}_1)\Big] + \mathbb{E}_{\mathbf{Z}}\Big[\sum_{i=1}^{T} \log p(\mathbf{y}_i|\mathbf{z}_i)\Big] + \mathbb{E}_{\mathbf{Z}}\Big[\sum_{i=2}^{T} \log p(\mathbf{z}_i|\mathbf{z}_{i-1})\Big] \quad (2.3)$$

The EM algorithm alternates between maximizing the $\mathcal{Q}$ function with respect to the parameters Λ and with respect to the distribution of hidden states, holding the other quantity fixed. The E-step depends on $\mathbb{E}[\mathbf{z}_i|\mathbf{Y}]$, $\mathbb{E}[\mathbf{z}_i\mathbf{z}_i^\top|\mathbf{Y}]$ and $\mathbb{E}[\mathbf{z}_i\mathbf{z}_{i-1}^\top|\mathbf{Y}]$, which are sufficient statistics to compute eq.(2.3). Detailed algorithms for computing the sufficient statistics are provided in Appendices A and B. The M-step re-estimate each of the parameter in Λ by

taking the corresponding partial derivative of the expected log likelihood, setting to zero and solving.

Spectral Learning Spectral learning methods provide a non-iterative, asymptotically unbiased LDS estimation solution in closed form. They estimate the parameters of an LDS by using singular value decomposition (SVD) to find Kalman filter estimates of the underlying state sequence [Katayama, 2005, Van Overschee and De Moor, 1996, Doretto et al., 2003]. Spectral learning methods approximate the observation matrix $\mathbf{Y}$ or its variants (hankel matrix) [Boots et al., 2007] into $U\Sigma V'$ by SVD, where $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{T\times d}$ have orthonormal columns $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ and $\Sigma = \text{diag}\{\delta_1, \cdots, \delta_d\}$ contains the singular values. The emission matrix and state sequence are estimated as $\hat{C} = U$ and $\hat{\mathbf{Z}} = \Sigma V'$ and the transition matrix is obtained by solving the least square of $\|A\mathbf{Z}_{1:T-1} - \mathbf{Z}_{2:T}\|_F^2$ where $\mathbf{Z}_{a:b}$ represents a subsequence of $\mathbf{Z}$ inclusively from the time t_a to time t_b, i.e., $\mathbf{Z}_{a:b} = [\mathbf{z}_a, \mathbf{z}_{a+1}, \cdots, \mathbf{z}_{b-1}, \mathbf{z}_b]$.

Due to iterative re-estimation the EM is slower than spectral methods that do not iterate. However, the maximum likelihood solution found by EM might provide more accurate parameter estimation than spectral learning methods, especially when the amount of training data is small, but is subject to local optima. In practice, the estimates from spectral learning are used as the initialization of the EM algorithm [Boots et al., 2007].

However, even though the standard EM and spectral methods are maturely developed, learning LDS from short-span low-sample clinical MTS data set encounters a number of questions. First, both EM and spectral methods require to know the intrinsic dimensionality of an LDS's hidden state space in advance, which in general is difficult. The dimensionality plays an important role in the performance of LDS models due to the fact that a small number of hidden states may not be able to model the complexities of a MTS, while a large number of hidden states can lead to overfitting. In Section 3.1 of this dissertation, we address the above issue by presenting a regularized LDS framework to recover the intrinsic dimensionality of MTS and consequently prevent model overfitting given short MTS data sets. Second, neither the EM algorithm nor spectral methods are able to constrain the LDS learning process in the sense of leading the learned models to achieve desired properties, such as stability. In Section 3.2 of this dissertation, we propose and develop a generalized

LDS learning framework in which various constraints are easily incorporated and parameter optimizations are efficiently conducted.

2.2.1.3 Irregularly Sampled Data Discretization In general, there are two ways to handle irregularly sampled time series data and convert them to observation sequences one can model and analyze using the discrete time models: (1) direct value interpolation (DVI) approach and (2) window-based segmentation (WbS) approach. In the following we briefly summarize these two approaches.

Direct Value Interpolation The DVI approach assumes that all observations are collected regularly with a pre-specified sampling frequency r. However, instead of actual readings the values at these time points are estimated from readings at time points closest to them using various interpolation techniques [Adorf, 1995, Dezhbakhsh and Levy, 1994, Åström, 1969]. The interpolated (regular) time series, i.e., $\tilde{\mathbf{Y}} = \{(\tilde{\mathbf{y}}_i, \tilde{t}_i)_{i=1}^{\tilde{T}}\}$, is then used to train a discrete time model such as LDS. The approach is illustrated in Figure 5. We put a tilde sign ($\tilde{\cdot}$) over $\mathbf{Y}$ and $\mathbf{y}_i$ to indicate the discretized observations. $\tilde{t}_i$ is time stamps of discretized observations and $\tilde{T}$ is the length of discretized sequence. In terms of predictions of future values, one has to first use trained discrete time model to predict the values at time points closest to the target time, and after that, apply the interpolation approach to estimate the target value.

The DVI approach converts the time series with irregular observations to discrete time observation sequences. The quality of the conversion depends on the number of observations actually seen and the sampling frequency parameter r. One straightforward way to set r is to use internal cross-validation approach. Briefly, we divide the time series data used for training the models into folds and use them to built multiple internal training and testing datasets. The models built with different sampling frequencies r are tested on the internal test sets, and the best r that leads to the best prediction accuracy on the internal test data (averaged over different folds) is selected.

Window-based Segmentation The WbS approach is slightly different. Instead of values at pre-specified regularly sampled time points, the approach first segments time series to fixed-sized windows. The behavior in the window is summarized in terms of its statistics

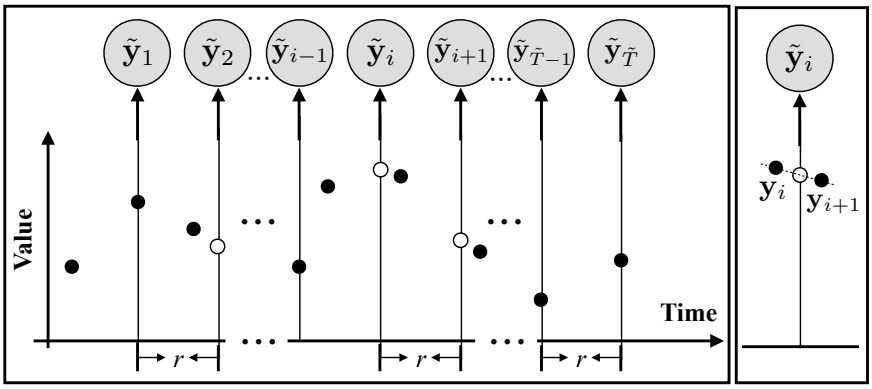


Figure 5: Transformation of an irregularly sampled time series $\mathbf{Y} = \{(\mathbf{y}_i, t_i)_{i=1}^{T}\}$ to a discrete time series $\tilde{\mathbf{Y}} = \{(\tilde{\mathbf{y}}_i, \tilde{t}_i)_{i=1}^{\tilde{T}}\}$ by using DVI. The empty circles denote the interpolated values with no readings. The right panel illustrates the linear interpolation process.

γ, such as, the mean, or the last value observed within that time interval [Chu, 1995, Das et al., 1998, Yi and Faloutsos, 2000, Keogh and Pazzani, 2000, Keogh et al., 2001, Smyth and Keogh, 1997]. The values generated by the different windows define sequences of γ statistics. The discrete time model is then used to represent how the summary statistics γ in two consecutive windows change, that is, a sequence of statistics calculated over these intervals are considered to be observations of the discrete time model. Predictions at future times for the window-based approach are made using the discrete time model by identifying the time interval the target time point falls into.

We would like to note that in order to learn the parameters of the window-based discrete time model from irregularly sampled data one has to either assure that every time interval has at least one reading that is sufficient to calculate the summary statistics; or impute the statistics for the window with missing values from its neighbors using, for example, interpolation methods. Figure 6 illustrates the process of filling statistics in intervals with missing values by using interpolations. Briefly, after segmentation of time series to windows of a fixed size (step 1), the summary statistics γ_i for each window i are calculated (step 2), and for windows with no readings, the statistics are interpolated from windows next to it (step 3). Once the missing statistics are imputed, the discrete time models, such as LDS,

can be learned from complete sequences $\gamma_1, \gamma_2, \cdots, \gamma_m$ of summary statistics derived from time series data.

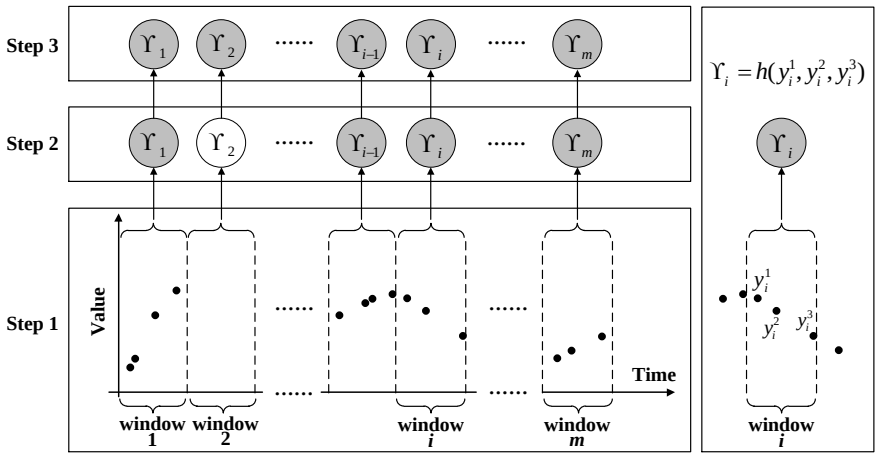


Figure 6: Transformation of irregularly sampled time series $\mathbf{Y} = \{(\mathbf{y}_i, t_i)_{i=1}^T\}$ to a discrete time series $\boldsymbol{\gamma} \equiv \{\gamma_i\}$ by WbS. The shaded nodes denote summary statistics calculated from the corresponding windows, such as γ_i in step 2. The regular (unshaded) nodes denote empty summary statistics corresponding to windows with no readings, such as γ_2. h in the right panel denotes the summary statistics estimation function.

The discrete time models (once they are learned) can be used for prediction by taking an initial sequence of observations for a new instance and predicting values at an arbitrary future time t^*. This is accomplished by first applying the WbS to observed data for the new instance and by calculating or imputing the statistics γ for each window. The value at some future time t^* is predicted by using the time series model (like LDS) to predict the statistics γ^* for the window the future time falls into and after that infer the value for target time t^* from γ^*. We note the simplest implementation of step 3 is to predict the value directly with the summary statistic. Briefly, if the summary statistic reflects the value of observations in the respective time window, we may directly use this value to predict the value for any time that falls within the corresponding window.

The above window-based approach can be further refined by overlapping two consecutive windows that generate the statistic γ in time. This means some of the observations can be

shared by two windows and may influence the statistics in two consecutive steps. Overlapping the two windows helps to smooth the transitions in statistics. In addition, it helps to generate longer sequences one can use to train better models. The idea of window overlap is illustrated in Figure 7. Considering windows and their overlaps, the segmentation of the time series is induced by two parameters: the window size $\mathcal{W}$ and the overlap size $\mathcal{O}$. These are additional parameters of the WbS approach, and if needed, they can be optimized using the internal cross-validation approach.

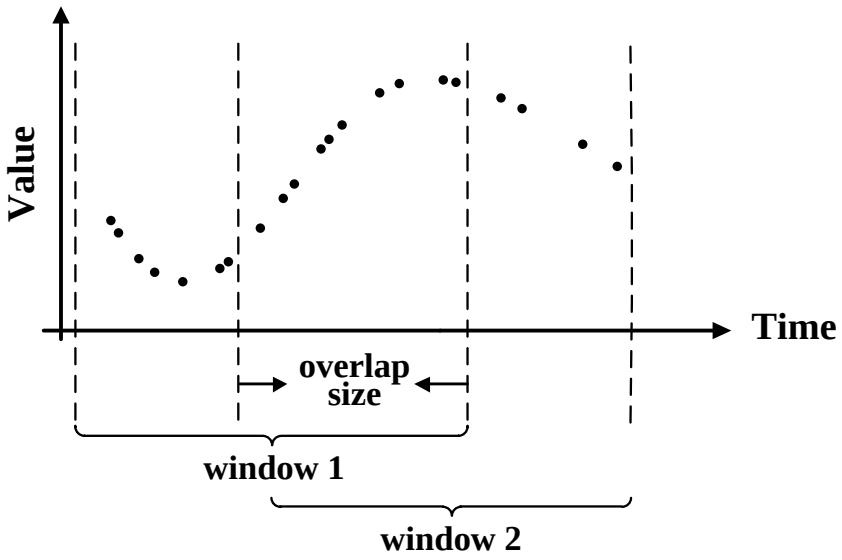


Figure 7: The graphical illustration of WbS with overlaps on the irregularly sampled time series data.

2.2.2 Gaussian Process

The Gaussian process (GP) is a popular nonparametric nonlinear Bayesian model in statistical machine learning [Rasmussen and Williams, 2006]. A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution. The GP is best viewed as an extension of the multivariate Gaussian to infinite-sized collections of real-valued variables defining the distribution over random functions. Table 1 summarizes the relationship between Gaussian distribution, multivariate Gaussian distribution and the GP.

Table 1: Relationship between Gaussian distribution, multivariate Gaussian distribution and Gaussian process.

	Mean type	(Co)variance type
Gaussian distribution	Scalar	Scalar
Multivariate Gaussian distribution	Vector	Matrix
Gaussian process	Function	Function

A GP is represented by the mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and the covariance function $K^G(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$, where $f(\mathbf{x})$ is a real-valued process and $\mathbf{x}$ is the input vector. The mean function $m(\mathbf{x})$ indicates the central tendency of the process, and the covariance function controls the variation in terms of the similarity or distance of the two input vectors $\mathbf{x}$ and $\mathbf{x}'$.

The GP can be used to calculate the distribution $p(f(\mathbf{X}^*))$ of f values for an arbitrary set of inputs $\mathbf{X}^*$. The distribution $p(f(\mathbf{X}^*))$ is a multivariate Gaussian defined as follows.

$$f(\mathbf{X}^*) \sim \mathcal{N}(m(\mathbf{X}^*), K^G(\mathbf{X}^*, \mathbf{X}^*)) \tag{2.4}$$

Eq.(2.4) defines the prior distribution of $f(\mathbf{X}^*)$. In addition, the GP can be used to calculate the posterior distribution $p(f(\mathbf{X}^*)|(\mathbf{X}, \mathbf{Y}))$ of f values for inputs $\mathbf{X}^*$, given a set of observed values $\mathbf{Y}$ for $\mathbf{X}$, where $\mathbf{Y} = f(\mathbf{X}) + \epsilon$, assuming additive independent identically distributed Gaussian noise ϵ with variance σ^2, $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The posterior is again a multivariate Gaussian $p(f(\mathbf{X}^*)|(\mathbf{X}, \mathbf{Y}))$ defined as follows.

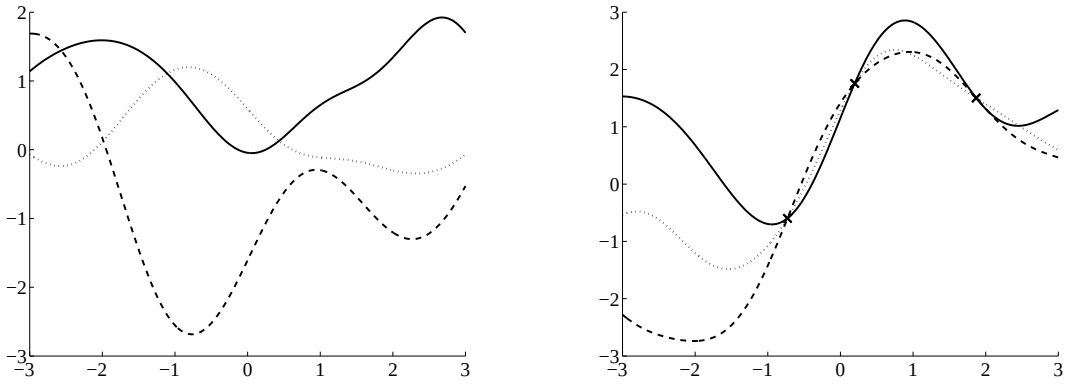
$$f(\mathbf{X}^*)|(\mathbf{X}, \mathbf{Y}) \sim \mathcal{N}(m(\mathbf{X}^*|(\mathbf{X}, \mathbf{Y})), Cov(\mathbf{X}^*|(\mathbf{X}, \mathbf{Y}))) \tag{2.5}$$

where the mean and covariance expressions are:

$$m(\mathbf{X}^*|(\mathbf{X},\mathbf{Y})) = m(\mathbf{X}^*) + K^G(\mathbf{X}^*,\mathbf{X})\left[K^G(\mathbf{X},\mathbf{X}) + \sigma^2 I\right]^{-1}(\mathbf{Y} - m(\mathbf{X})) \tag{2.6}$$

$$Cov(\mathbf{X}^*|(\mathbf{X},\mathbf{Y})) = K^G(\mathbf{X}^*,\mathbf{X}^*) - K^G(\mathbf{X}^*,\mathbf{X})\left[K^G(\mathbf{X},\mathbf{X}) + \sigma^2 I\right]^{-1} K^G(\mathbf{X},\mathbf{X}^*). \tag{2.7}$$

Figure 8 illustrates the examples of functions drawn from the GP prior and posterior in a 1-D space; Figure 8(a) shows functions drawn from the prior distribution function values at $\mathbf{X}^*$. Figure 8(b) shows functions drawn from the posterior distributions given that some data points $(\mathbf{X},\mathbf{Y})$ are observed.



(a) Three functions drawn at random from the zero-mean GP prior.

(b) Three random functions drawn from the GP posterior given three observations.

Figure 8: The graphical illustration of GP prior and posterior. In this example, we create X^* as a linearly spaced vector from -3 to 3 with step size 0.01. We set the mean function $m(\cdot) = 0$ and covariance function $K^G(x,x') = \exp(-(x-x')^2/2)$.

2.2.2.1 Applications Due to the function view of GP methodology and its corresponding flexible nature, GP has a variety of applications in solving temporal modeling problems. In the following, we describe two major applications of GP in time series domain.

GP as A Function of Time As we discuss in Section 2.2.2, GP can be viewed as an extension of the multivariate Gaussian distribution in the function space (infinite space) which can be directly applied to time series modeling problems by representing observations

as a function of time [Roberts et al., 2013, Girard et al., 2003, Brahim-Belhouari and Bermak, 2004]. As a result, there is no restriction on when the observations are made and whether they are regularly or irregularly spaced in time and it can be easily applied to make future time prediction. Given any time index t^* we can calculate its posterior mean with eq.(2.6), and use it to predict the values at that time. Figure 9 illustrates this step.

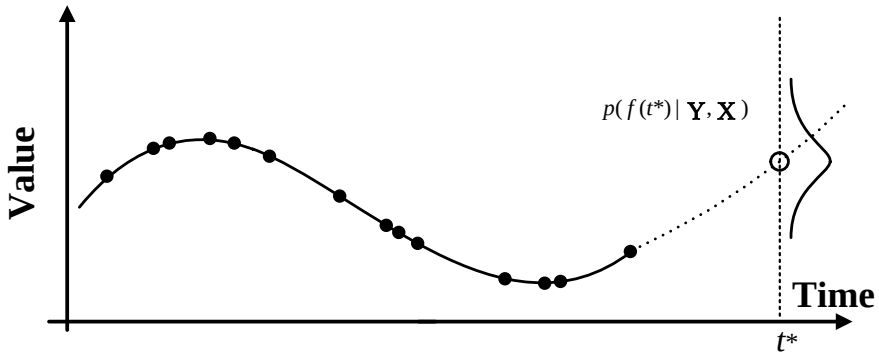


Figure 9: The graphical illustration of the prediction problem on a GP model on irregularly sampled time series data. The solid line denotes the GP we learned from the data and the dotted line indicates the GP's predictions of future values for future time t^*. The posterior distribution of $f(t^*)$ at time t^* is shown and the empty circle is the mean of that distribution, which is the value predicted by the GP.

GP as A Non-linear Transformation Instead of using GP as a function of time, we can choose to use GP as a non-linear transformation operator and substitute GP for the linear transformations in traditional temporal models. For example, in the LDS model defined by eq.(2.1) and eq.(2.2), we can replace the transition matrix A and emission matrix C, which are linear transformation operators, with two GPs $r(\cdot)$ and $u(\cdot)$. This leads to the following discrete-time Gaussian process dynamical system [Turner et al., 2010].

$$\mathbf{z}_i = r(\mathbf{z}_{i-1}) + \boldsymbol{\epsilon}_i \tag{2.8}$$

$$\mathbf{y}_i = u(\mathbf{z}_i) + \boldsymbol{\zeta}_i \tag{2.9}$$

The transition function $r(\cdot)$ and the observation function $u(\cdot)$ represent stochastic transitions and observations, and are represented with the help of Gaussian processes. $\boldsymbol{\epsilon}_i$ and

$\boldsymbol{\zeta}_i$ are the same as in eq.(2.1) and eq.(2.2). Briefly, the LDS assumes linear dependencies among latent states and observations, while the GP based model replaces the linear dependencies with more general nonlinear functions $r(\cdot)$ and $u(\cdot)$. Please note that if $\mathbf{z}_i$ states are observed then the model collapses to an autoregressive model which is represented by a single GP. [Turner et al., 2010] introduced the GPIL algorithm for inference and learning in the above discrete-time Gaussian process dynamical system based on the EM framework. Similar ideas appear in [Wang et al., 2005, Wang et al., 2008, Deisenroth et al., 2009, Ko and Fox, 2011] for building nonlinear dynamic systems by utilizing GP.

2.2.2.2 Learning Gaussian Process Models The parameters of the GP are formed by parameters defining the mean and covariance functions. The mean function is the function of time and the covariance function measures the similarity of two function values based on corresponding input time stamps.

The prior mean function is considered as the expectation function, prior to any observation. Usually, we are equally unsure whether the time series trend is up and down and this symmetry of ignorance leads to constant-offset mean functions [Roberts et al., 2013]. While in some cases, we do have a prior domain knowledge of the long-term trend of the time series, we can easily incorporate the specific function form into the Gaussian process models and the mean function's parameters can be optimized by using gradient based approaches. In the clinical setting, where the focus of this thesis lies, we want to learn a function that fits many patients and their clinical time series. Since the patients may be encountered at different age and under different circumstances, there is no good way to align their time origins. Hence the only way to feasibly align them is to set their mean functions equal to a constant $m(t) = M$, which makes the mean function of a GP time invariant. To obtain M, we can average all the observations from all the patients and use that averaged value as the constant M for the mean function. This gives us a constant mean which reflects many patients and their clinical time series.

To learn the parameters of the covariance function, we seek Θ that can maximize the marginal likelihood $p(\mathbf{Y}|\mathbf{X})$ [Rasmussen and Williams, 2006]. The log marginal likelihood

for GP is shown in eq.(2.10).

$$\log p(\mathbf{Y}|\mathbf{X}) = -\frac{1}{2}\mathbf{Y}^\top K_{\mathbf{Y}}^{-1}\mathbf{Y} - \frac{1}{2}\log|K_{\mathbf{Y}}| - \frac{T}{2}\log 2\pi \tag{2.10}$$

where $\mathbf{Y}$ denotes all the training observations. $K_{\mathbf{Y}} = K^G + \sigma^2 I$ is the covariance matrix for the noisy observations $\mathbf{Y}$ and K^G is the covariance matrix for noisy-free function values from function f, $\mathbf{Y} = f(\mathbf{X}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$. n is the number of observations.

The partial derivatives of the marginal likelihood with respect to each parameter θ_i in Θ is shown in eq.(2.11).

$$\frac{\partial}{\partial\theta_i}\log p(\mathbf{Y}|X,\Theta) = -\frac{1}{2}\mathrm{Tr}\left[K_{\mathbf{Y}}^{-1}\frac{\partial K_{\mathbf{Y}}}{\partial\theta_i}\right] + \frac{1}{2}\mathbf{Y}^\top K_{\mathbf{Y}}^{-1}\frac{\partial K_{\mathbf{Y}}}{\partial\theta_i}K_{\mathbf{Y}}^{-1}\mathbf{Y} \tag{2.11}$$

where Θ represents the entire set of parameters in covariance function, $\Theta = \{\theta_i\}$.

Once we have the partial derivatives with respect to each parameter, any well developed gradient based methods can be directly applied to maximize $p(\mathbf{Y}|\mathbf{X})$.

In summary, the advantage of the GP model is that it lets us represent functions of time and their distributions, which has no restriction on when the observations are made and whether they are regularly or irregularly spaced in time. However, this approach also comes with limitations; the most serious one is that the mean function of the GP is a function of time and in order to make the GP independent of the time origin we need to set it to a constant value. However, this significantly limits our ability to represent changes or different modes in time series dynamics. In Chapter 4 of this dissertation, we propose a new hierarchical dynamical system for modeling irregularly sample univariate time series, which combines the advantages of the LDS and GP models. A combination of the two appears as the best solution to offset their limitations.

2.2.3 Multi-task Gaussian Process

One limitation of applying GP to clinical MTS is that each clinical time series is modeled independently within a patient and the interactions between multiple clinical variables are neglected. To address this issue and capture the multivariate behaviors within the clinical MTS, the multi-task Gaussian process (MTGP) is proposed [Bonilla et al., 2007]. The MTGP

is an extension of GP to model multiple tasks (e.g., multivariate time series) simultaneously by utilizing the learned covariance between related tasks. MTGP uses K^C to model the similarities between tasks and uses K^G to capture the temporal dependence with respect to time stamps. The covariance function of MTGP is shown as follows:

$$K^M = K^C \otimes K^G + D \otimes I_T \tag{2.12}$$

where K^C is a positive semi-definite matrix and $K^C_{j,k}$ measures the similarity between time series j and time series k. D is an $n \times n$ diagonal matrix in which $D_{j,j}$ is the noise variance δ_j^2 for the jth time series. $\otimes$ is the Kronecker product. Usually the MTGP model has the computation limitation that it has $\mathcal{O}(n^3T^3)$ compared with $n \times \mathcal{O}(T^3)$ for standard GP models. However, this limitation is not as relevant in our application setting, given that the number of clinical observations is very limited and clinical time series are usually short span.

The parameters of the GP based models are formed by parameters defining the mean and covariance functions. Typically, the covariance function makes sure the function values for two nearby times tend to have high covariance, while values from inputs that are far apart in time tend to have a low covariance. The parameters can be learned from data that consist of one or many examples of time series. The predictions of values at future times correspond to calculation of posterior distribution for these times.

The advantages of GP based models is that (1) with the reasonable choice of the covariance function, GP based models are capable of capturing the short-term rapid changes in clinical time series [Clifton et al., 2013, Ghassemi et al., 2015]; and (2) GP based models can be applied to time series modeling problem by representing observations as a function of time. As a result, there is no restriction on when the observations are made and whether they are regularly or irregularly spaced in time.

2.3 INSTANCE-SPECIFIC MODELING

Building predictive models from available data is a fundamental task in machine learning. Typically, a single model is learned from a collection of training instances. After that, the

learned model is applied to all future instances. In this dissertation, we call such a model a *population based model*, which is optimized to have good predictive performance on average on all the future instances.

In spite of the huge successful applications of population based models, recent research has demonstrated that learning specific models to particular instances can improve the performance [Visweswaran and Cooper, 2004, Gottrup et al., 2005, Visweswaran et al., 2015]. Different from the population based model learned from the entire training data, such specific models are either trained on a particular instance or a group of specific instances or adjusted from the population based model according to the specific instance. In this dissertation, such a model is referred to as an *instance-specific model*. Recently, building and using instance-specific models have been shown great success in genetics, pharmacology, and other important aspects of healthcare such as personal preferences, nutrition, lifestyle, and disease, recapturing the importance of personalized health [Jørgensen, 2009, Swan, 2009, Schleidgen et al., 2013, Karkar et al., 2015, Wiley et al., 2016].

In general an instance-specific model can be achieved by:

- building instance-specific models for each instance. The instance-specific model is learned from a selected collection of similar examples out of the entire population. We refer these models to as *Subpopulation Models* (Section 2.3.1).
- adjusting the population based model to fit better the specific instance. This usually includes two steps: first learn a population based model from all available data and then calibrate the population based model according on the unique characteristics of each instance. We refer this approach to as *Model Adaptation* (Section 2.3.2).
- instance-dependently combining a pool of predictive models which are built either from the entire population or a subpopulation of instances. We refer this technique to as *Adaptive Model Selection* (Section 2.3.3).

Please note that models from the above three categories are complementary and they can be combined in the prediction process. For example, the model adaptation techniques can be applied to both population based models and subpopulation models. Moreover, both subpopulation models and adaptive models can be candidate models in the pool of the

adaptive model selection approaches. In the following, we briefly review the three approaches to build the personalized model.

2.3.1 Subpopulation Models

The data available for model building (learning) purposes may cover a wide variety of past patients and their conditions. However, using all of them may bias the model towards the population mean. The most common way to alleviate the problem and build a patient specific model is to identify a subpopulation of patients most similar to the target patient and learn a model using only examples from this subset. The subpopulation approaches usually rely on some pre-defined similarity measures to evaluate similarity between the target example (the patient that needs to be predicted) and all training examples (all available past patients), that is, a past patient is used to build a model for the target patient only if it is highly similar to the target patient.

The main challenge when adopting the subpopulation approach is to define proper similarity among patients and their respective time series. The majority of approaches in the literature assume the the similarity among patients relies on some atemporal patient specific information (such as demographics of the patient) to guide the personalized strategies. Deriving the similarity of two time series or mixed atemporal and temporal information is more complex. To measure the similarity of time series sequences of equal length, Euclidean distance, Pearson correlation, cosine distance and their variants are typically used. These types of similarities were used, for example, for classifying high-grade gliomas with gene expression profiles [Nutt et al., 2003], detection of seizure onsets [Qu and Gotman, 1997] and anomalous patient traces [Huang et al., 2014]. For symbol based time series, which have discrete values at each time stamp, edit distance based similarity has been be successful in several clinical applications, such as predicting length of stay for clinical treatment process [Huang et al., 2013], finding similar patient traces for clinical pathway analysis [Huang et al., 2014], etc. For real-valued time series of different lengths, the similarity can be computed either explicitly by using dynamic time warping [Berndt and Clifford, 1994, Müller, 2007] or implicitly by using the likelihood of generative probabilistic models defining the time

series [Liao, 2005]. Dynamic time warping explicitly computes distance between sequences by aligning the two series so that their difference is minimized. On the other hand, implicit approaches consider that each time series is generated by some kind of model and time series are considered similar when the models characterize individual series. For example, [Huang et al., 2015] models and clusters medical inpatient journeys and the similarity between two inpatient journeys are computed based on the sequences' log likelihood of their respective hidden Markov models.

One limitation of subpopulation based approaches is that a subpopulation from which we start and learn a subpopulation model from may still be very large and exhibit a lots of patient specific variations. So it may be necessary to further explore methods that can adapt the prediction model closer to the current patient.

2.3.2 Model Adaptation

Model adaptation methods try to bridge a possible gap in between population (or subpopulation models) and the target patient by adjusting the population model to fit better the specific patient. Broadly speaking, there are two types of mechanisms and strategies to modify the population based model to reflect the instance-specific characteristics, *model parameter adaptation* and *instance-specific residual modeling*.

Model parameter adaptation approaches achieve the personalized prediction results by modifying the model parameters of population based models based on instance-specific features. For example, [Berzuini et al., 1991, Berzuini et al., 1992] proposed a general Bayesian network model for individualized therapeutic monitoring. More specifically, each reading in patient specific white blood cell counts time series is modeled by a Gaussian distribution. The mean of the Gaussian changes with time and is modeled by the piece-wise linear function whose parameters (slope and bias) have a population based prior. The population based prior is estimated from all related past patients. The forecasting is made by adapting this population based prior to patient specific posteriors by using patient specific covariates (i.e., patient's age, gender, etc.) and patient specific recent observations.

Different from model parameter adaptation approaches, instance-specific residual based

techniques add additional models to support the personalized predictive outcomes. In such approaches, residuals are defined as the difference between the true outcomes of the specific instance and the predictive results of the population based models. Adding extra individualized models based on residuals tries to capture the specific deviations and offset the insufficient ability of the population based models. For example, Hou et al. [Hou and Zhang, 2007] present a spectral residual model for visual saliency detection where the spectral residual is represented as the difference from the log spectrum of an image and its corresponding approximation from a local average filter.

In spite of the successful applications of the model adaptation techniques, they have some limitations. For model parameter adaptation approaches, designing and deriving adaptation is very difficult and varies from model to model. Even under Bayesian adaption framework with simplified distribution assumptions, approximations are need for such tasks. Furthermore, for model parameter adaptation approaches, they usually require larger instance-specific features or observations. However, time series observed for one patient are often too short to support inadequate adaptation.

To utilize the advantages and flexibility of model adaptation based approaches and to overcome the limitations, in Section 5.1 of this dissertation, we explicitly model the gaps (residuals) between population based models and each specific patient and develop a two-stage adaptive forecasting model. Our model benefits from the population trend extracted from past data collection and at the same time adapt to patient specific data, thus allowing one to make more accurate MTS predictions.

2.3.3 Adaptive Model Selection

The adaptive model selection approaches the personalized prediction problems by assuming a pool of candidate predictive models that may contribute to the prediction. In adaptive model selection, a different model or combination of models may support the predictions at the different time. Briefly, each of the candidate models is associated with weight that reflects how much it contributes to the final solution. Two different strategies: ensemble (Section 2.3.3.1) and online (Section 2.3.3.2) methods are commonly be used to choose (optimize) the

weights in the machine learning literature.

2.3.3.1 Ensemble Methods Ensemble methods are general techniques in machine learning for combining several models to create a more accurate prediction [Caruana et al., 2004]. Related research work focuses on either creating more candidate models, such as bagging [Breiman, 1996], boosting [Freund and Schapire, 1997] or by wisely optimizing their combination weights, such as exponential weighting, stacking [Smyth and Wolpert, 1999], etc. In medical and clinical practice, the ensemble methods can often significantly boost the performance of individual models. [Moon et al., 2007] distinguishes disease subtypes on lymphoma patients and lung cancer patients by using a robust classification algorithm that is developed for high-dimensional data based on ensembles of classifiers. [Jiang et al., 2012] develops a data-driven approach to utilize individualized confidence intervals to select the most "appropriate" model from a pool of candidates to predict patient's specific clinical condition. [Visweswaran and Cooper, 2004] performs a selective Bayesian model averaging for each individual patient where the prediction is made by first searching for models having the greatest impact on the target prediction and then averaging the predictions from selected models.

2.3.3.2 Online Algorithms Online prediction problems have been studied extensively in the theoretical machine learning literature [Littlestone and Warmuth, 1994, Blum, 1998]. In online prediction problems, various techniques, such as the weighted majority algorithm [Littlestone and Warmuth, 1994], hedge algorithm [Freund and Schapire, 1997] are proposed to select the best model from the candidate pool based on the knowledge of the past. The models with poorer performance receive larger penalties and become less likely to be picked in the future. There have been many papers that aim to apply online learning to solving real-world problems, for example, classifying handwritten digits [Crammer et al., 2006], detects malicious Web sites [Ma et al., 2009], but as far as we know few or none work has been applied to time series forecasting in clinical settings.

Although many ensemble and online methods have been proposed, the majority of them require error feedback over longer periods of time to optimize the combination weights.

However, clinical time series are usually too short to obtain effective weights for those algorithms. Furthermore, weight updating rules are often based on the overall performance of each model on all previously observed data and hence the recent errors are smoothed out by the errors made in the early stage of the process. Since clinical MTS may contain short-term variability standard weight updating rules are not able to respond to these changes quickly enough. To address the above problems, in Section 5.2 of this dissertation, we develop a new online model switching strategy to put more penalties on recent errors. Our approach helps predictive models that perform well recently but do not perform well initially to be selected as soon as possible.

3.0 LEARNING LINEAR DYNAMICAL SYSTEMS FROM REGULARLY SAMPLED MULTIVARIATE TIME SERIES

In Chapter 3, we develop two frameworks to learn temporal models from regularly sampled multivariate time series data with complex dependence. Our work focuses on the refinements of a popular model for MTS analysis: the linear dynamical system (described in Section 2.2.1). The first framework focuses on learning regularized LDS models from short-span and low-sample MTS collections to avoid the overfitting problems (Section 3.1). The second work provides a generalized framework to learn LDS from a collection of MTS sequences via matrix factorization. By using our approach, constraints can be easily incorporated into the LDS learning process, which drives the dynamics to meet the modeling expectations (Section 3.2). We note that the material of the regularized LDS learning framework presented in Section 3.1 was originally published as [Liu and Hauskrecht, 2015b] and the material of the generalized framework for learning LDS with constraints presented in Section 3.2 was originally published as [Liu and Hauskrecht, 2016b].

3.1 REGULARIZED LINEAR DYNAMICAL SYSTEMS

Though the LDS is a classical and widely used model for real-valued sequence analysis, learning an LDS model from short-span MTS may encounter the following challenges:

1. The observational sequences in MTS data may exhibit strong interactions. This raises a question on *how many hidden states are needed to represent the system dynamics well given a MTS sequences?*;

2. The number of parameters representing transitions among hidden state components (a.k.a transition matrix) is quadratic in the dimensionality of the hidden space, which raises a question on *how we can prevent the overfit of the model parameters when the training size is small?*

In this section, we present work we have completed that identifies these challenges by studying learning methods that impose regularization penalties on the transition matrix of the LDS model and propose a regularized LDS learning framework (rLDS) which aims to (1) automatically shut down LDSs' spurious and unnecessary dimensions, and consequently, address the problem of choosing the optimal number of hidden states; (2) prevent the overfitting problem given a small amount of MTS data; and (3) support accurate MTS forecasting.

3.1.1 The Regularized Framework

In our rLDS framework, the LDS has a large implicit state space but a low-rank transition matrix. The rLDS recovers the intrinsic dimensionality of MTS by using the rank of transition matrix rather than the state space size. In order to achieve the low-rank property, we introduce a prior, i.e., $p(A)$ (The choice of $p(A)$ is discussed in the Section 3.1.2.1) for the hidden state transition matrix A. The graphical illustration of our rLDS is shown in Figure 10 and the log joint probability distribution for our rLDS is:

$$\log\Big(p(\{\mathbf{Z}^l\},\{\mathbf{Y}^l\},A)\Big) = \sum_{l=1}^{N}\log p(\mathbf{z}_1^l) + \sum_{l=1}^{N}\sum_{i=1}^{T_l} p(\mathbf{y}_i^l|\mathbf{z}_i^l) + \sum_{l=1}^{N}\sum_{t=2}^{T_l}\log p(\mathbf{z}_i^l|\mathbf{z}_{i-1}^l,A) + \log p(A).$$

3.1.2 EM Learning

We develop an Expectation-Maximization (EM) algorithm for the maximum a posteriori (MAP) estimation of the rLDS which aims at maximizing the $\mathcal{Q} = \mathbb{E}_{\mathbf{Z}}[\log p(\{\mathbf{Z}^l\},\{\mathbf{Y}^l\},A)]$ function:

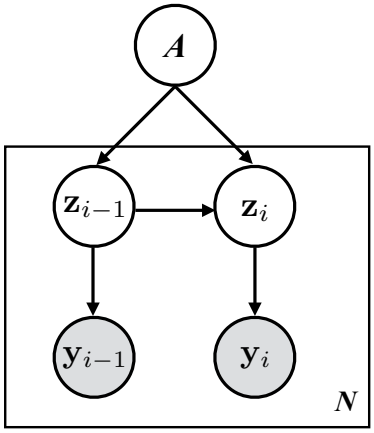


Figure 10: The graphical illustration of our rLDS model. Shaded nodes $\mathbf{y}_i$ denote observations and unshaded nodes $\mathbf{z}_i$ correspond to hidden states. The plate represents a collection of N sequences.

$$\mathcal{Q} = \sum_{l=1}^{N} \mathbb{E}_{\mathbf{Z}^l}\Big[\log p(\mathbf{z}_1^l)\Big] + \sum_{l=1}^{N}\sum_{i=1}^{T_l} \mathbb{E}_{\mathbf{Z}^l}\Big[\log p(\mathbf{y}_i^l|\mathbf{z}_i^l)\Big] + \sum_{l=1}^{N}\sum_{t=2}^{T_l} \mathbb{E}_{\mathbf{Z}^l}\Big[\log p(\mathbf{z}_i^l|\mathbf{z}_{i-1}^l)\Big] + \log p(A). \tag{3.1}$$

The EM algorithm alternatively iterates the E-step and M-step until the it converges. In E-step, since the hidden state Markov chain defined by the LDS is unobserved, we cannot learn our rLDS directly. Instead, we infer the hidden state expectations. The E-step infers a posterior distribution of latent states ($\mathbf{Z}^l$) given the observation sequences ($\mathbf{Y}^l$). The E-step requires computing the value of $\mathcal{Q}$ functions (eq.(3.1)), which depends on three sufficient statistics $\mathbb{E}[\mathbf{z}_i^l|\mathbf{Y}^l]$, $\mathbb{E}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]$ and $\mathbb{E}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top|\mathbf{Y}^l]$. Here we follow the backward algorithm in [Ghahramani and Hinton, 1996] to compute them. The backward algorithm is presented in the Appendix B.

In M-step, we try to find the set of LDS parameters $\Omega = \{A, C, Q, R, \pi, \Psi\}$ that maximizes the likelihood lower bound $\mathcal{Q}$ (eq.(3.1)). As we can see, $\mathcal{Q}$ function's differentiability with respect to A depends on the choice of A's prior, i.e., $p(A)$, while it is differentiable

with respect to $(C, R, Q, \boldsymbol{\xi}, \Psi)$. Therefore, we separate the optimization into two parts, i.e., optimization of A (Section 3.1.2.1) and optimization of $\Omega \backslash A$ (Section 3.1.2.2).

3.1.2.1 Optimization of A In each iteration in the M-step, we need to maximize

$$\sum_{l=1}^{N} \sum_{i=2}^{T_l} \mathbb{E}_{\mathbf{Z}^l} \Big[\log p(\mathbf{z}_i^l | \mathbf{z}_{i-1}^l) \Big] + \log p(A)$$

with respect to A, which is equivalent to

$$\min_A g(A) - \log p(A)$$

,

where

$$g(A) = \frac{1}{2} \sum_{l=1}^{N} \sum_{i=2}^{T_l} \mathbb{E}_{\mathbf{Z}^l} \Big[(\mathbf{z}_i^l - A\mathbf{z}_{i-1}^l)^\top Q^{-1} (\mathbf{z}_i^l - A\mathbf{z}_{i-1}^l) \Big].$$

In order to recover the intrinsic dimensionality from MTS datasets through the rank of transition matrix A rather than the state space size d, we need to choose specific priors which can induce the desired low-rank property. Here we have three choices of inducing a low-rank A: (1) a nuclear norm prior; (2) both univariate and multivariate Laplacian priors; and (3) only a multivariate Laplacian prior as shown in Table 2. A_i represents each row (or column) [1] of A. The prior choices lead to three instances of our rLDS framework, rLDS$_\mathcal{R}$, rLDS$_\mathcal{S}$ and rLDS$_\mathcal{G}$.

Table 2: Prior choices for rLDS.

Prior Name	Model	Prior Form	Regularization
Nuclear norm	rLDS$_\mathcal{R}$	$\propto \exp(-\lambda_N \|A\|_*)$	$\lambda_N \|A\|_*$
Uni/multi-Lap.	rLDS$_\mathcal{S}$	$\propto \exp(-\lambda_U \|A\|_1 - \sum_{i=1}^d \lambda_M \|A_i\|_2)$	$\lambda_U \|A\|_1 + \lambda_M \sum_{i=1}^d \|A_i\|_2$
Multi-variate Lap.	rLDS$_\mathcal{G}$	$\propto \exp(-\lambda_M \sum_{i=1}^d \|A_i\|_2)$	$\lambda_M \sum_{i=1}^d \|A_i\|_2$

rLDS$_\mathcal{R}$: a nuclear norm prior. In rLDS$_\mathcal{R}$, we assume A has a nuclear norm density. In order to avoid overfitting, we add a multivariate Gaussian prior to each A_i, which leads to

[1] Without loss of generality, we will use A_i to represent the row in the following text.

the ridge regularization. Therefore, we combine the nuclear norm prior and Gaussian prior to get a new prior for transition matrix A, which leads to the following log probability:

$$\log p(A|\lambda_N, \lambda_G) = -\lambda_N \|A\|_* - \frac{\lambda_G}{2}\|A\|_F^2 + const, \tag{3.2}$$

and the objective function we want to optimize becomes:

$$\min_A h(A) + \lambda_N \|A\|_* \quad \text{where } h(A) = g(A) + \frac{\lambda_G}{2}\|A\|_F^2 \tag{3.3}$$

Since $h(A)$ is convex and differentiable with respect to A, we can adopt the proximal gradient descent algorithm to minimize eq.(3.3). The update rule is

$$A^{(k+1)} = \text{prox}_{\rho_k}\Big(A^{(k)} - \rho_k \bigtriangledown h(A^{(k)})\Big) \tag{3.4}$$

where ρ_k is the step size at the kth iteration. $\bigtriangledown h(A)$ is the gradient of $h(A)$, which is

$$\bigtriangledown h(A) = Q^{-1}\Big(A(\sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]) - (\sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top|\mathbf{Y}^l])\Big) + \lambda_G A \tag{3.5}$$

The proximal function $\text{prox}_{\rho_k}(A)$ is defined as the singular value soft-thresholding operator,

$$\text{prox}_{\lambda_N\rho_k}(A) = U \cdot \text{diag}((\sigma_i - \lambda_N\rho_k)_+) \cdot V^\top \tag{3.6}$$

where $A = U\text{diag}(\sigma_1, \cdots, \sigma_d)V^\top$ is the singular value decomposition (SVD) of A.

An important open question here is how to set the step size of the proximal gradient method to assure it is well behaved. Theorem 1 gives us a simple way to select the step size while also assures its fast convergence rate.

Theorem 1. *Proximal gradient descent with a fixed step size*

$$\rho \leq 1/(||Q^{-1}||_F \cdot ||\sum_{l=1}^{N}\sum_{i=1}^{T_l-1}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]||_F + \lambda_G) \tag{3.7}$$

for minimizing eq.(3.3) has convergence rate $O(1/k)$, where k *is the number of iterations.*

Proof. The proof appears in the Appendix C. □

Algorithm 1 Proximal descent algorithm for solving eq.(3.3).

INPUT:

- Expectations from E-step, $\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]$ and $\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top|\mathbf{Y}^l]$.
- Hyper-parameters, λ_N and λ_G.
- rLDS parameters from last iteration, Q.

PROCEDURE:

1: Initialize A by solving $\bigtriangledown h(A) = 0$ with $\lambda_G = 0$.
2: Compute step size ρ by eq.(3.7).
3: **repeat**
4: Compute the gradient $\bigtriangledown h(A)$ by eq.(3.5).
5: Update A, $A = \text{prox}_{\lambda_N \rho}(A - \rho \bigtriangledown h(A))$ by eq.(3.4).
6: **until** Convergence

OUTPUT: A.

The optimization procedure for eq.(3.3) is summarized by Algorithm 1.

rLDS$_\mathcal{S}$: univariate and multivariate Laplacian priors. In rLDS$_\mathcal{S}$, we apply both univariate and multivariate Laplacian priors to achieve the within and between row sparsity on the transition matrix. More specifically, we introduce a multivariate Laplacian prior to each row of A, A_i which turns out to be an $\ell_1\ell_2$ regularizer on every row of the transition matrix to enforce a row-level sparsity. Furthermore, we utilize the univariate Laplacian prior and apply it on every element of the trainsition matrix to obtain a within-row sparsity which is equal to imposing an ℓ_1 regularizer on every element of the transition matrix. Similar to rLDS$_\mathcal{R}$, we add a multivariate Gaussian prior to each A_i, which leads to the ridge regularization to avoid overfitting and enhance numerical stability. In this case, our objective function is

$$\min_A g(A) + \lambda_U \|A\|_1 + \lambda_M \sum_{i=1}^{d} \|A_i\|_2 + \frac{\lambda_G}{2}\|A\|_F^2 \tag{3.8}$$

Recently there have been many approaches proposed to solve the joint regularization of ℓ_1 and $\ell_1\ell_2$ norm optimization problem [Bach et al., 2011, Liu et al., 2009, Liu and Ye, 2010]. Along the existing optimization solutions, we introduce an algorithm to solve eq.(3.8). Its foundation is the following theorem:

Theorem 2. *Maximizing eq.(3.8) is equivalent to minimizing the following problem:*

$$\min_{a} \frac{1}{2} a^\top H a - b^\top a + \lambda_U \|a\|_1 + \lambda_M \sum_{i=1}^{d} \|a_{G_i}\|_2 \tag{3.9}$$

where $a = \mathrm{vec}(A^\top)$, $Q^{-1} = LL^\top$, $s = (W^{-1})^\top b$, *and* $\{G_i\}_{i=1}^d$ *is the row membership indicator.* H *and* b *are defined as follows:*

$$H = (Q^{-1} \otimes \sum_{l=1}^{N} \sum_{i=2}^{T_l} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_{i-1}^l (\mathbf{z}_{i-1}^l)^\top] + \lambda_G I_{d^2}) \tag{3.10}$$

$$b = (L \otimes \sum_{l=1}^{N} \sum_{i=2}^{T_l} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l (\mathbf{z}_{i-1}^l)^\top]^\top) \,\mathrm{vec}(L). \tag{3.11}$$

Proof. The proof of this theorem appears in the Appendix D. □

As we can see, eq.(3.9) is a decomposable function such that it can be broken down into a convex and differentiable function $q(a)$ and a penalty function $r(a)$ of two non-smooth norms.

$$\min_{a} \underbrace{\frac{1}{2} a^\top H a - b^\top a}_{q(a)} + \underbrace{\lambda_U \|a\|_1 + \lambda_M \sum_{i=1}^{d} \|a_{G_i}\|_2}_{r(a)} \tag{3.12}$$

The penalty function $r(a)$ is known as the sparse group Lasso penalty, which allows simultaneous within and between group level sparsification in a. We decide to solve the optimization problem (eq.(3.9)) by using the incremental proximal descent methods [Richard et al., 2012, Zhou et al., 2012], which have attracted extensive attentions in machine learning and data mining communities due to its optimal convergence rate among all first order optimization methods and its ability of dealing with non-smooth penalties. The key ingredient of proximal minimization methods lies in the proximal operator and the corresponding proximal operator of eq.(3.9) is defined as follow:

$$\mathcal{P}_{\ell_1,\ell_1\ell_2}(v) = \arg\min_{a} \frac{1}{2} \|a - v\|_2^2 + \lambda_U \|a\|_1 + \lambda_M \sum_{i}^{d} \|a_{G_i}\|_2 \tag{3.13}$$

In eq.(3.13), the minimization is coupled with $\|a\|_1$ and $\|a_{G_i}\|_2$, which makes it difficult to solve. However, we know that the for each individual non-smooth norm, i.e., $\|a\|_1$ (eq.(3.14)) and $\|a_{G_i}\|_2$ (eq.(3.15)), the following proximal operators can be solved analytically [Liu et al., 2009, Liu and Ye, 2009, Liu and Ye, 2010].

$$\mathcal{P}_{\ell_1}(v) = \arg\min_a \frac{1}{2}\|a - v\|_2^2 + \lambda_U \|a\|_1 \tag{3.14}$$

$$\mathcal{P}_{\ell_1\ell_2}(v) = \arg\min_a \frac{1}{2}\|a - v\|_2^2 + \lambda_M \sum_i^d \|a_{G_i}\|_2 \tag{3.15}$$

Hence, we adopt a two-stage incremental proximal descent algorithm to efficiently compute eq.(3.13) by utilizing the proximal operator decomposition property, which leads to Theorem 3.

Theorem 3. *The unique solution, $\hat{a}$, to eq.(3.13) is $\mathcal{P}_{\ell_1\ell_2}(\mathcal{P}_{\ell_1}(a))$, i.e.,*

$$\mathcal{P}_{\ell_1,\ell_1\ell_2}(a) = \mathcal{P}_{\ell_1\ell_2}(\mathcal{P}_{\ell_1}(a))$$

Proof. Similar proof can be found in [Zhou et al., 2012]. □

The optimization procedure for eq.(3.9) is summarized by Algorithm 2.

Algorithm 2 Incremental proximal descent algorithm for solving eq.(3.9).

INPUT:

- Expectations from E-step, $\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]$ and $\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top|\mathbf{Y}^l]$.
- Hyper-parameters, λ_U, λ_M and λ_G.
- rLDS parameters from last iteration, Q.

PROCEDURE:

1: Compute L from decomposition of Q^{-1}, where $Q^{-1} = LL^\top$.
2: Compute H and b based on eq.(3.10) and eq.(3.11).
3: Initialize A by solving $\bigtriangledown h(A) = 0$ with $\lambda_G = 0$.
4: Reshape $a = \text{vec}(A)$.
5: Solve eq.(3.14), $\hat{a} = \mathcal{P}_{\ell_1}(a)$. See [Liu and Ye, 2009].
6: Solve eq.(3.15), $\hat{a} = \mathcal{P}_{\ell_1\ell_2}(\hat{a})$. See [Liu et al., 2009, Liu and Ye, 2010].

OUTPUT: $\hat{a}$.

rLDS$_\mathcal{G}$: multivariate Laplacian priors. In rLDS$_\mathcal{G}$, we drop the univariate Laplacian prior assumption from rLDS$_\mathcal{S}$, which only induce the row level sparsity on the transition matrix A. The optimization problem is similar to the problem in rLDS$_\mathcal{S}$ by setting $\lambda_U = 0$ in eq.(3.9) and we can still use Algorithm 2 to compute the solution efficiently.

3.1.2.2 Optimization of $\Omega \backslash A$ Each of these parameters is estimated similarly to [Ghahramani and Hinton, 1996] by taking the corresponding derivative of the eq.(3.1), setting it to zero, and by solving it analytically. Update rules for $\Omega \backslash A = \{C, R, Q, \boldsymbol{\xi}, \Psi\}$ are as follows:

$$C^{(k+1)} = \Big(\sum_{l=1}^{N}\sum_{i=1}^{T_l} \mathbf{y}_i^l (\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l|\mathbf{Y}^l])^\top\Big)\Big(\sum_{l=1}^{N}\sum_{i=1}^{T_l} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]\Big)^{-1} \tag{3.16}$$

$$R^{(k+1)} = \frac{1}{\sum_{l=1}^{N} T_l}\Big(\sum_{l=1}^{N}\sum_{i=1}^{T_l} \mathbf{y}_i^l (\mathbf{y}_i^l)^\top - C^{(k+1)} \sum_{l=1}^{N}\sum_{i=1}^{T_l} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l|\mathbf{Y}^l](\mathbf{y}_i^l)^\top\Big) \tag{3.17}$$

$$Q^{(k+1)} = \frac{1}{\sum_{l=1}^{N} T_l - N}\Big(\sum_{l=1}^{N}\sum_{i=2}^{T_l} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l] - A^{(k+1)} \sum_{l=1}^{N}\sum_{i=2}^{T_l} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_{i-1}^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]\Big) \tag{3.18}$$

$$\boldsymbol{\xi}^{(k+1)} = \sum_{l=1}^{N} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_1^l|\mathbf{Y}^l] \tag{3.19}$$

$$\Psi^{(k+1)} = \sum_{l=1}^{N} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_1^l(\mathbf{z}_1^l)^\top|\mathbf{Y}^l] - \sum_{l=1}^{N} \mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_1^l|\mathbf{Y}^l](\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_1^l|\mathbf{Y}^l])^\top \tag{3.20}$$

3.1.2.3 Model Learning Summary The entire parameter estimation procedure for rLDS is summarized by Algorithm 3.

Algorithm 3 Parameter estimation in rLDS

INPUT:

- Initialization $\Omega^{(0)} = \{A^{(0)}, C^{(0)}, Q^{(0)}, R^{(0)}, \boldsymbol{\xi}^{(0)}, \Psi^{(0)}\}$.
- Hyper-parameters, λ_N, λ_U, λ_M and λ_G.

PROCEDURE:

repeat
  E-step: estimate $\mathbb{E}[\mathbf{z}_i^l|\mathbf{Y}^l]$, $\mathbb{E}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]$ and $\mathbb{E}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top|\mathbf{Y}^l]$.
  M-step:
  if rLDS$_\mathcal{R}$ **then**
    estimate A by Algorithm 1.
  end if
  if rLDS$_\mathcal{S}$ **then**
    estimate A by Algorithm 2.
  end if
  if rLDS$_\mathcal{G}$ **then**
    estimate A by Algorithm 2 with $\lambda_U = 0$.
  end if
  M-step: estimate $C, R, Q, \boldsymbol{\xi}, \Psi$ by eqs.(3.16 - 3.20)
until Convergence

OUTPUT: Learned rLDS parameters: $\hat{\Omega} = \{\hat{A}, \hat{C}, \hat{Q}, \hat{R}, \hat{\boldsymbol{\xi}}, \hat{\Psi}\}$.

3.1.3 Experiment

In this section, we will (1) verify that our regularized LDS approach indeed results in a low-rank solution and (2) show that our rLDS models are able to alleviate model overfitting by starting the learning process from a large initial hidden state space and by working with small amounts of training data. Experiments are conducted on both synthetic and real-world datasets. We would also like to note that the hyper parameters (λ_N, λ_U, λ_M and λ_G) used in our methods are selected (in all experiments) by the internal cross validation approach while optimizing models' predictive performances.

3.1.3.1 Baselines We compare the three instances of our rLDS framework, i.e., rLDS$_{\mathcal{R}}$, rLDS$_{\mathcal{S}}$ and rLDS$_{\mathcal{G}}$ to the following LDS learning baselines:

- LDS learned using the standard EM learning algorithm (EM) [Ghahramani and Hinton, 1996] that iteratively finds the maximum likelihood solution.
- Subspace identification algorithm (SubspaceID) [Van Overschee and De Moor, 1996]. SubspaceID computes an asymptotically unbiased solution in closed form by using oblique projection and SVD.
- Stable linear dynamical system (StableLDS) [Boots et al., 2007]. StableLDS constrains the largest singular value of the transition matrix to ensure the stability of LDS models.

3.1.3.2 Evaluation Metrics We evaluate and compare the performance of the different methods by calculating the average Mean Absolute Percentage Error (Average-MAPE) of models' predictions. The Average-MAPE is defined as follows:

$$\text{Average-MAPE} = \frac{\sum_{l=1}^{N}\sum_{j=1}^{n}\sum_{i=1}^{T_l}|1-\hat{y}_{ji}^{l}/y_{ji}^{l}|}{n\sum_{l=1}^{N}T_l} \times 100\% \tag{3.21}$$

where y_{ji}^{l} and $\hat{y}_{ji}^{l}$ are the ith true value and prediction of time series j of MTS sequence l.

3.1.3.3 Data

Synthetic Data To get a good understanding of our approach, we first test it on synthetic data. We generate our synthetic MTS dataset of length $T = 200$ using a 20-state LDS

with zero-mean, 0.01 variance Gaussian innovations. A uniform random emission matrix C is used to generate 20 measurements at each time stamp t with i.i.d. zero mean variance 0.01 measurement noise. We uniformly and randomly generate a 20×20 matrix, normalize its SVD decomposition by its largest singular value to ensure its stability and truncate its 10 smallest singular values to obtain an exact 10-rank matrix A.

Clinical Data We also test our rLDS on a MTS clinical data obtained from electronic health records of post-surgical cardiac patients in PCP database [Hauskrecht et al., 2010b, Valko and Hauskrecht, 2010, Hauskrecht et al., 2013]. We take 500 patients from the database who had their *Complete Blood Count* (CBC) tests [2] done during their hospitalization. The MTS data consists of 6 individual CBC lab time series: mean corpuscular hemoglobin concentration (MCHC), mean corpuscular hemoglobin (MCH), mean corpuscular volume (MCV), mean platelet volume (MPV), red blood cell (RBC) and red cell distribution width (RDW). The average length of patient sequence is 17.19 and the data statistics are shown in Table 3. In order to have a comprehensive and fair comparisons of our algorithm against the baseline methods, we conduct two experiments on the clinical data with different training data sizes (50 and 400) and the learned model are test on the same dataset with size 100.

Table 3: Data statistics of a real-world clinical dataset.

Clinical MTS	Unit	Mean	Max	Min	Std
MCHC	g/dL	33.88	37.00	30.50	0.74
MCH	pg/cell	30.46	36.30	22.00	1.45
MCV	fL	89.90	102.70	69.30	3.64
MPV	fL	8.67	15.90	6.00	1.01
RBC	10^{12}/L	3.36	6.38	1.62	0.56
RDW	%	15.51	29.30	11.30	1.78

[2] CBC panel is used as a broad screening test to check for such disorders as anemia, infection, and other diseases.

3.1.3.4 Results

Intrinsic Dimensionality Recovery We train the three instances (rLDS$_{\mathcal{R}}$, rLDS$_{\mathcal{S}}$ and rLDS$_{\mathcal{G}}$) of our rLDS framework on the synthetic data whose real hidden state space size 10. We start our rLDS learning with different initial state space sizes, i.e., $d = 15$, 20 and 30. The results of rLDS$_{\mathcal{R}}$, rLDS$_{\mathcal{S}}$ and rLDS$_{\mathcal{G}}$ for recovering MTS intrinsic dimensionality are shown in Figure 11. The figure shows three different graphs corresponding to three different experiments for 15, 20 and 30 initial state space sizes respectively. We normalized all the singular values by their maximum values to make sure they are in the same scale and between 0 and 1. We can see that the low-rank inducing priors listed in Table 2 lead us to a low-rank transition matrix and that our rLDS framework is able to recover the correct hidden dimension (which is 10) even if the dimensionality of the initial state space is large. Furthermore, we can see our rLDS framework is robust to different initial state space size and it is consistently recovering the intrinsic true dimensinality of the LDSs.

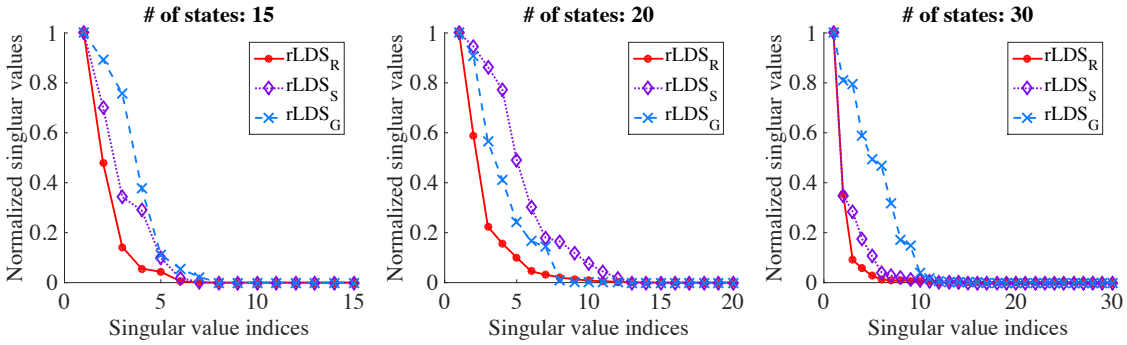


Figure 11: State space recovery on a synthetic dataset.

LDS Overfitting Phenomena We first train the standard EM learning algorithm on the clinical data by considering many initial hidden state space sizes. We vary the hidden state size from 1 to 30. For each trained LDS, we measure its predictive performance on the corresponding test set by Average MAPE (eq.(3.21)). Due to the fact that the ground truth of the exact intrinsic is unknown, we define the "optimal" number of hidden states based on the LDS's predictive performance under this setting and the results are shown in Figure 12. As we can see, the prediction performance varies a lot with the different number of hidden states we use in the model and it difficult to choose the optimal number without

the extensive number of experiments. The results show the model performances vary a lot with different number of hidden states. Finding the optimal number of hidden states by validation checking each candidate state size is time consuming.

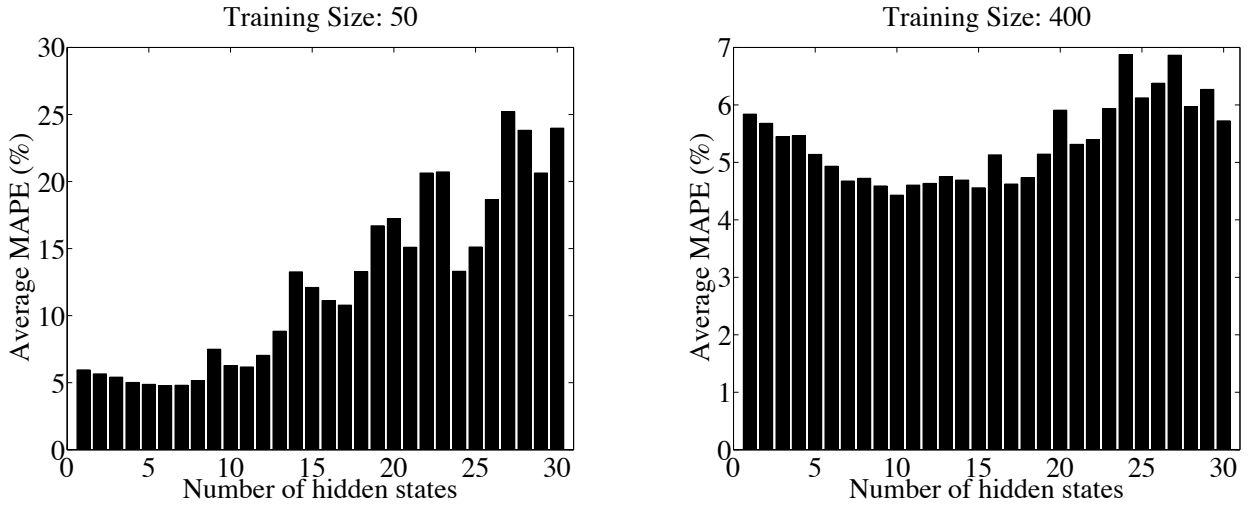


Figure 12: LDS overfitting phenomena with standard EM learning algorithm for the LDS with the different number of hidden states.

Sparsification Effects To have a comprehensive evaluation of the rLDS's ability of shutting down unnecessary dimension, we run the three instances of our rLDS framework on a real-world clinical data with the different number of initial states (10, 20 and 30). The normalized singular values of transition matrices in different experimental settings are shown in Figure 13. As we can see from Figure 13, our approaches are able to consistently seek the intrinsic dimensions and capture the dynamics using a lower-dimensional hidden state space representation. Among the three instances of rLDS, both rLDS$_{\mathcal{R}}$ and rLDS$_{\mathcal{S}}$ prefer lower dimensions compared to rLDS$_{\mathcal{G}}$.

Prediction Performance Prediction performance is an important evaluation metric in time series modeling. In order to gain a more comprehensive insight into rLDS's prediction abilities, we test our rLDS with many initial state space sizes on the clinical data. The results of these experiments are summarized in Table 4. The results show that our rLDS methods are able to outperform all the baselines in terms of their prediction performance

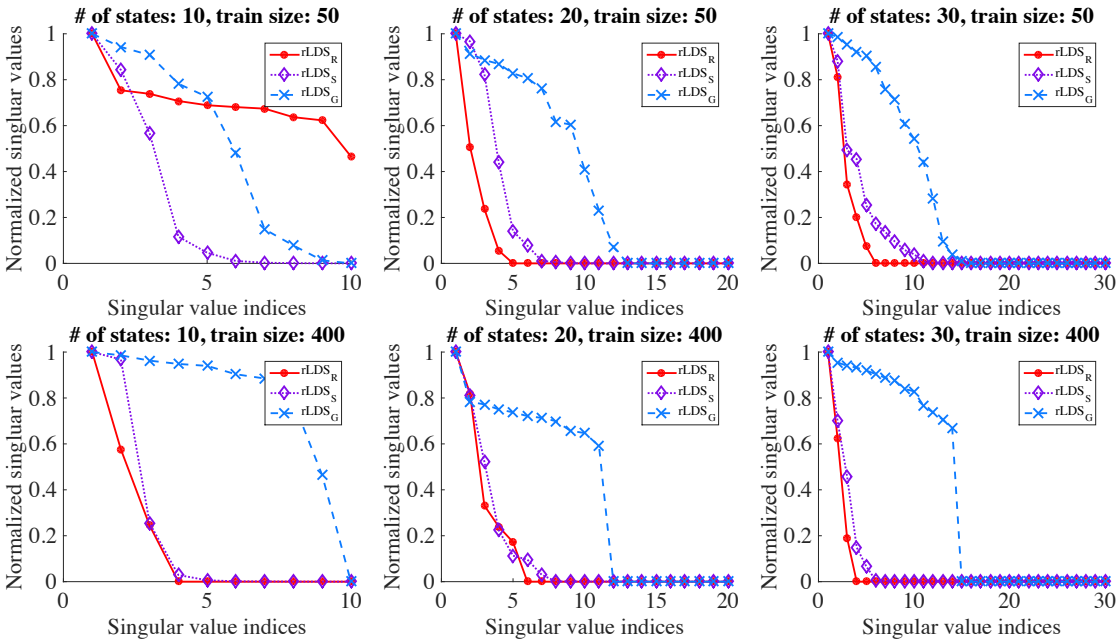


Figure 13: State space recovery on the clinical data.

in nearly all settings. Specifically, we find the following results: (1) when comparing rLDS methods to other baselines, we can see that they do improve the prediction performance by shutting down unnecessary dimensions, especially when the initial state space size goes up. Our rLDS methods are robust and demonstrate a reasonably good prediction quality for various initial state space sizes; (2) comparing *SubspaceID* and *StableLDS*, we can see that they tend to have similar prediction errors in majority cases. This is because in the learning step, *StableLDS* differs from *SubspaceID* by one extra step: it enforces the transition matrix to stay in the stable space. If the transition matrix from *SubspaceID* is already stable, there is no difference between them; and (3) comparing the three instances (rLDS$_{\mathcal{R}}$, rLDS$_{\mathcal{S}}$ and rLDS$_{\mathcal{G}}$) of our framework, we find that rLDS$_{\mathcal{R}}$ and rLDS$_{\mathcal{G}}$ have better performance compared to rLDS$_{\mathcal{S}}$. This is because the hyper parameter tuning process in rLDS$_{\mathcal{S}}$ is more complex compared to rLDS$_{\mathcal{R}}$ and rLDS$_{\mathcal{G}}$. In rLDS$_{\mathcal{S}}$ there are two hyper parameters (λ_U, λ_M) needed to be selected based on the corresponding prediction performance on validation set while in rLDS$_{\mathcal{R}}$ and rLDS$_{\mathcal{G}}$, only one hyper parameter needs to be set. In rLDS$_{\mathcal{S}}$, the sparsification

Table 4: Average-MAPE results on the clinical data with different training sizes.

	Training Size: 50			**Training Size: 400**		
# of states	10	20	30	10	20	30
EM	6.28	17.24	23.98	**4.43**	5.91	5.72
SubspaceID	6.55	6.99	7.44	6.10	6.16	6.27
StableLDS	6.54	6.99	7.40	6.10	6.16	6.27
$\text{rLDS}_{\mathcal{R}}$	**4.65**	**4.95**	5.01	4.65	4.46	4.67
$\text{rLDS}_{\mathcal{S}}$	5.34	5.48	5.67	5.36	5.63	5.70
$\text{rLDS}_{\mathcal{G}}$	4.98	4.97	**4.86**	4.51	**4.25**	**4.35**

ability is jointly controlled by λ_U and λ_M, which makes it more sensitive to their values. On the other hand, $\text{rLDS}_{\mathcal{R}}$ and $\text{rLDS}_{\mathcal{G}}$ are more stable and insensitive to the hyper parameter selections.

3.2 CONSTRAINED LINEAR DYNAMICAL SYSTEMS

Instead of learning ordinary LDS models from time series data collection, we may expect to obtain enhanced LDS models which have desired properties, such as smoothness, stability, etc. This is done by adding constraints into the LDS models. In this dissertation, we refer the enhanced LDS models as to *constrained LDS*.

Various researchers have proposed to incorporate different constraints into either the inference process (estimating hidden states from data given known parameters) or the learning process (estimating parameters from data) to improve LDS performance and its quality. In terms of the LDS inference process, most of the existing refinements try to enforce different types of sparsity constraints on the estimates of the hidden state. For example, in [Angelosante et al., 2009, Carmi et al., 2010, Charles et al., 2011] the hidden states are sparsified

during the Kalman filter inference step. All these approaches assume that all parameters of the LDS are known a priori. Hence they are not directly applicable to the problem of learning LDS models from MTS data. In terms of the LDS learning process, they are typically incorporated to achieve special model properties, such as low-rankness [Liu and Hauskrecht, 2015b], stability [Boots et al., 2007], and others. For example, an algorithm for learning stable LDS is proposed by [Boots et al., 2007]. An LDS with dynamic matrix A is stable if all of A's eigenvalues have magnitude at most 1. The stability is crucial when simulating long sequences from LDS models in order to generate representative data or infer stretches of missing values.

In this work, we propose a new generalized LDS framework, gLDS, for learning LDS models from a collection of MTS data with various constraints. Our learning framework is based on matrix factorization approach, where each MTS sequence is factorized as a product of a shared emission matrix and a sequence-specific hidden dynamics. In contrast to traditional matrix factorization, the hidden factors in gLDS may evolve in time and individual dynamics is modeled with the help of a shared transition matrix. We use alternating minimization to learn the constrained LDS model from data. In such a case, each parameter can be optimized efficiently and the procedure is flexible enough to incorporate various constraints. Furthermore, we propose a temporal smoothing regularization, which penalizes the difference of predictive results from the learned model during the learning phase, to achieve smooth forecasts from the learned LDS models.

3.2.1 A Generalized LDS Framework

Let $\mathbf{Y}^l \in \mathcal{R}^{n \times T_l}$ represent the MTS for the lth patient and $\mathbf{Z}^l \in \mathcal{R}^{d \times T_l}$ is the corresponding hidden state sequence. $\mathbf{Z}^l_+ = [\mathbf{z}^l_2, \mathbf{z}^l_3, \cdots, \mathbf{z}^l_{T_l}]$ and $\mathbf{Z}^l_- = [\mathbf{z}^l_1, \mathbf{z}^l_2, \cdots, \mathbf{z}^l_{T_l-1}]$. We use $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{Z}_+$, and $\mathbf{Z}_-$ to denote the horizontal concatenations of $\{\mathbf{Y}^l\}$, $\{\mathbf{Z}^l\}$, $\{\mathbf{Z}^l_+\}$, and $\{\mathbf{Z}^l_-\}$. Here $\mathbf{Y}$ is an $n \times T$ matrix and $\mathbf{Z}$ is a $d \times T$ matrix. $\mathbf{Z}_+$ and $\mathbf{Z}_-$ are $d \times (T - N)$ matrices where $T = \sum_{l=1}^{N} T_l$.

Based on the linear assumption in LDS that sequential observation vector is generated by the linear emission transformation C from hidden states at each time stamp (eq.(2.2)), we

can formulate the LDS learning problem by using the matrix factorization approach [Koren et al., 2009,Lee and Seung, 1999] that assumes the collection of MTS sequences are generated by a shared emission matrix and their specific hidden factors.

$$\min_{C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 \tag{3.22}$$

However, different from traditional matrix factorization models where the hidden factors are in general time independent, in LDS models, hidden factors evolve in time and are specified by eq.(2.1). Hence, similar to [Boots et al., 2007, Doretto et al., 2003], we estimate the transition matrix A by solving another least square problem as follows:

$$\min_{A,\mathbf{Z}} \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 \tag{3.23}$$

In this work, in order to learn the LDS parameters from the data, we jointly optimize both eq.(3.22) and eq.(3.23). Furthermore, in order to incorporate constraints into the learned LDS models and avoid the overfitting problem, we add regularizations for C, A and $\mathbf{Z}$ into the objective function, shown as follows:

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\mathcal{R}_C(C) + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) + \gamma\mathcal{R}_A(A) \tag{3.24}$$

Intuitively, this formulation of the problem aims to find an LDS model that is able to fit as accurately as possible the time series in the training data by using a simple (less complex) model.

3.2.2 Learning via Matrix Factorization

As we can see from eq.(3.24), the coupling between A, C and $\mathbf{Z}$ makes this problem difficult to solve for A, C and $\mathbf{Z}$ simultaneously, so in this work, we adopt the alternating optimization scheme to find the solution iteratively.

3.2.2.1 Optimization of A, C, and $\mathbf{Z}$ We apply the alternating minimization techniques to eq.(3.24), which leads to the following three optimization problems:

$$\min_{A} \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \gamma/\lambda \mathcal{R}_A(A) \tag{3.25}$$

$$\min_{C} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \alpha \mathcal{R}_C(C) \tag{3.26}$$

$$\min_{\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \beta \mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) \tag{3.27}$$

Since optimization of a hidden state sequence $\mathbf{Z}^l$ is independent of other sequences, we can further decompose the optimization target $\mathbf{Z}$ into $\{\mathbf{Z}^1, \cdots, \mathbf{Z}^l \cdots \mathbf{Z}^N\}$. Due to the asymmetric positions of different $\mathbf{z}_i^l$s in $\mathbf{Z}^l$, we decompose the optimization into three parts: $\mathbf{z}_1^l$, $\mathbf{z}_i^l$ and $\mathbf{z}_{T_l}^l$ $(i = 2, \cdots, T_l - 1)$. The optimization problems for each hidden states sequence $\mathbf{Z}^l$ are defined as follows:

$$\min_{\mathbf{z}_1^l} \|\mathbf{y}_1^l - C\mathbf{z}_1^l\|_2^2 + \lambda \|\mathbf{z}_2^l - A\mathbf{z}_1^l\|_2^2 + \beta \mathcal{R}_{\mathbf{Z}}(\mathbf{z}_1^l) \tag{3.28}$$

$$\min_{\mathbf{z}_i^l} \|\mathbf{y}_i^l - C\mathbf{z}_i^l\|_2^2 + \lambda \|\mathbf{z}_i^l - A\mathbf{z}_{i-1}^l\|_2^2 + \lambda \|\mathbf{z}_{i+1}^l - A\mathbf{z}_i^l\|_2^2 + \beta \mathcal{R}_{\mathbf{Z}}(\mathbf{z}_i^l) \tag{3.29}$$

$$\min_{\mathbf{z}_{T_l}^l} \|\mathbf{y}_{T_l}^l - C\mathbf{z}_{T_l}^l\|_2^2 + \lambda \|\mathbf{z}_{T_l}^l - A\mathbf{z}_{T_l-1}^l\|_2^2 + \beta \mathcal{R}_{\mathbf{Z}}(\mathbf{z}_{T_l}^l) \tag{3.30}$$

3.2.2.2 Optimization of R, Q, ξ and Ψ Once we obtain A, C and $\mathbf{Z}$, the rest of LDS's parameters, R, Q, $\boldsymbol{\xi}$ and Ψ, can be analytically estimated as follows:

$$\hat{Q} = \frac{1}{T-N}(\hat{\mathbf{Z}}_+ - \hat{A}\hat{\mathbf{Z}}_-)(\hat{\mathbf{Z}}_+ - \hat{A}\hat{\mathbf{Z}}_-)^\top \tag{3.31}$$

$$\hat{R} = \frac{1}{T}(\mathbf{Y} - \hat{C}\hat{\mathbf{Z}})(\mathbf{Y} - \hat{C}\hat{\mathbf{Z}})^\top \tag{3.32}$$

$$\hat{\boldsymbol{\xi}} = \frac{1}{N}\sum_{l=1}^{N} \hat{\mathbf{z}}_1^l \tag{3.33}$$

$$\hat{\Psi} = \frac{1}{N}\sum_{l=1}^{N} \hat{\mathbf{z}}_1^l (\hat{\mathbf{z}}_1^l)^\top \tag{3.34}$$

Algorithm 4 Learn the LDS model in gLDS.

INPUT:

- Initialization $A^{(0)}, C^{(0)}, \mathbf{Z}^{(0)}$.
- Hyper-parameters, γ, λ, β and α.
- A collection of MTS sequences $\mathcal{D} = \{\mathbf{Y}^1, \cdots, \mathbf{Y}^N\}$.

PROCEDURE:

// Optimize A, C and $\mathbf{Z}$
repeat
  Update A by solving eq.(3.25).
  Update C by solving eq.(3.26).
  for l: $1 \rightarrow$ N **do**
    Update $\mathbf{z}_1^l$ by solving eq.(3.28).
    for i: $2 \rightarrow T_l - 1$ **do**
      Update $\mathbf{z}_i^l$ by solving eq.(3.29).
    end for
    Update $\mathbf{z}_{T_l}^l$ by solving eq.(3.30).
  end for
until Convergence
// Optimize $\hat{Q}, \hat{R}, \hat{\boldsymbol{\xi}}, \hat{\Psi}$
Compute $\hat{Q}, \hat{R}, \hat{\boldsymbol{\xi}}, \hat{\Psi}$ using eqs.(3.31 - 3.34).

OUTPUT:

- Learned LDS parameters: $\hat{\Omega} = \{\hat{A}, \hat{C}, \hat{Q}, \hat{R}, \hat{\boldsymbol{\xi}}, \hat{\Psi}\}$.

3.2.2.3 Summary

The entire LDS parameter estimation procedure in our gLDS framework is summarized by Algorithm 4.

3.2.3 Relationship to Existing Models

In this section, we describe and show how to formulate both existing stable LDS and regularized LDS as special instances in our gLDS framework.

3.2.3.1 Learning Regularized LDS (gLDS-low-rank)

In order to obtain a low-rank transition matrix of the LDS model, as introduced in Section 3.1, an MAP learning framework is developed and both multivariate Laplacian prior and nuclear norm prior are applied on the A to implicitly shut down spurious and unnecessary dimensions and prevent overfitting problem [Liu and Hauskrecht, 2015b]. In gLDS framework, a low-rank transition matrix A can be easily obtained by setting $\mathcal{R}_A(A) = \|A\|_F^2 + \frac{\lambda}{\gamma}\gamma_A\|A\|_*$ in eq.(3.25), which leads to the following objective function (eq.(3.35)). All the other updates for C, Q, R, ξ, and Ψ remain

the same.

$$\min_A g(A) + \gamma_A \|A\|_* \tag{3.35}$$

where

$$g(A) = \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \gamma/\lambda \|A\|_F^2$$

Since $g(A)$ is convex and differentiable with respect to A, we can adopt the generalized gradient descent algorithm to minimize eq.(3.35). The update rule is

$$A^{(k+1)} = \text{prox}_{\rho_k}\Big(A^{(k)} - \rho_k \bigtriangledown g(A^{(k)})\Big) \tag{3.36}$$

where ρ_k is the step size at kth iteration. The proximal function $\text{prox}_{\rho_k}(A)$ is defined as the singular value soft-thresholding operator,

$$\text{prox}_{\gamma_A \rho_k}(A) = U \cdot \text{diag}((\sigma_i - \gamma_A \rho_k)_+) \cdot V^\top \tag{3.37}$$

where $A = U\text{diag}(\sigma_1, \cdots, \sigma_d)V^\top$ is the SVD of A.

Similar to the optimization problem (eq.(3.3)) in Section 3.1.2.1, it is crucial to set the appropriate step size of the generalized gradient method to assure it is well behaved. Theorem 4 gives us a simple way to select the step size while also assuring its fast convergence rate.

Theorem 4. *Generalized gradient descent with a fixed step size* $\rho \leq 1/2(\|\mathbf{Z}_-\mathbf{Z}_-^\top\|_F + \gamma/\lambda)$ *for minimizing eq.(3.35) has convergence rate* $O(1/k)$*, where* k *is the number of iterations.*

Proof. The proof of this theorem appears in Appendix E. □

3.2.3.2 Learning Stable LDS (gLDS-stable) Stability is a desired property for dynamic and it plays important roles in tasks such as predictions, long term sequence simulation, etc. Boots et al. propose a novel method for learning stable LDS models by formulating the problem as a quadratic program [Boots et al., 2007]. The program starts with a relaxed solution and incrementally add constraints to improve stability. In gLDS framework, by setting $\mathcal{R}_A(A) = \emptyset$, we can easily transform our optimization to the same objective function

in [Boots et al., 2007]. Furthermore, we can apply the following theorem to change eq.(3.25) into the standard quadratic program formulation.

Theorem 5. *Minimizing A from eq.(3.25) with $\mathcal{R}_A(A) = \emptyset$ is equivalent to minimizing the following problem:*

$$\min_a a^\top Ba - 2q^\top a$$

where $a = \text{vec}(A^\top)$, $B = I_d \otimes (\mathbf{Z}_-\mathbf{Z}_-^\top)$, $q = (I_d \otimes \mathbf{Z}_-\mathbf{Z}_+^\top)\,\text{vec}(I_d)$.

Proof. The proof of this theorem appears in Appendix F. □

After the quadratic program transformation, we can apply the same constraints generation techniques described in [Boots et al., 2007] to optimize the transition matrix and guarantee its stability. Details can be found in [Boots et al., 2007].

3.2.4 The Ridge Model (gLDS-ridge)

Ridge regularization, a.k.a, Tikhonov regularization or ℓ_2 regularization [Hoerl, 1962], is widely used to prevent overfitting since it encourages the sum of the squares of the fitted parameters to be small. Furthermore, it alleviates the ill-posed problems in numerical methods. In our gLDS framework, we achieve the ridge model (gLDS-ridge) by setting $\mathcal{R}_C(C)$, $\mathcal{R}_A(A)$, and $\mathcal{R}_\mathbf{Z}(\mathbf{Z})$ to the square of Frobenius norm, i.e., $\|\cdot\|_F^2$.

Due to the differentiability of ridge regularization, we can take the partial derivatives of eqs.(3.25 - 3.30), set them to zero and solve. The results are shown as follows:

$$\begin{aligned}
\hat{A} &= (\mathbf{Z}_+\mathbf{Z}_-^\top)(\mathbf{Z}_-\mathbf{Z}_-^\top + \gamma/\lambda I_d)^{-1} \\
\hat{C} &= (\mathbf{Y}\mathbf{Z}^\top)(\mathbf{Z}\mathbf{Z}^\top + \alpha I_d)^{-1} \\
\hat{\mathbf{z}}_1^l &= (G + \lambda A^\top A)^{-1}(C^\top \mathbf{y}_1^l + \lambda A^\top \mathbf{z}_2^l) \\
\hat{\mathbf{z}}_i^l &= (G + \lambda A^\top A + \lambda I_d)^{-1}(F_i^l + \lambda A^\top \mathbf{z}_{i+1}^l) \\
\hat{\mathbf{z}}_{T_l}^l &= (G + \lambda I_d)^{-1} F_{T_l}^l
\end{aligned}$$

where $G = C^\top C + \beta I_d$ and $F_i^l = C^\top \mathbf{y}_i^l + \lambda A \mathbf{z}_{i-1}^l$.

3.2.5 The Smooth Model (gLDS-smooth)

In this section, we propose a novel temporal smoothing regularization (Section 3.2.5.1), which penalizes the difference of predictive results from the learned model during the learning phase, to achieve smooth forecasts from the learned LDS models. In Section 3.2.5.2, we show how to incorporate the temporal smoothing regularization into gLDS and describe the corresponding learning procedure.

3.2.5.1 Temporal Smoothing Regularization To incorporate temporal smoothness property in LDS models for MTS modeling and forecasting, we propose a temporal smoothing regularization term $\mathcal{R}_{\mathcal{T}}^{l}$ for each MTS sequence l, which penalizes the difference of two consecutive predictive values:

$$\mathcal{R}_{\mathcal{T}}^{l} = \sum_{i=2}^{T_l} w_{i-1,i}^{l} \|\hat{\mathbf{y}}_i^l - \hat{\mathbf{y}}_{i-1}^l\|_2^2 \tag{3.38}$$

where $\hat{\mathbf{y}}_i^l$ is the model forecast at time stamp i and $w_{i-1,i}^l$ is the smoothing coefficient balancing the difference between predictions $\hat{\mathbf{y}}_{i-1}^l$ and $\hat{\mathbf{y}}_i^l$.

Briefly, the regularization term penalizes the predictions at each temporally consecutive time stamps if they disagree with other prediction made within the same MTS sequence. The amount of penalty is controlled by a smoothing coefficient $w_{i-1,i}^l$. Furthermore, we can express eq.(3.38) into a more general form as follows:

$$\mathcal{R}_{\mathcal{T}}^{l} = \sum_{p=1}^{T_l-1} \sum_{q=p+1}^{T_l} w_{p,q}^{l} \|\hat{\mathbf{y}}_p^l - \hat{\mathbf{y}}_q^l\|_2^2 \tag{3.39}$$

where

$$w_{p,q}^{l} = \begin{cases} 1 & \text{if } |p-q| = 1, \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, after some algebraic manipulations, the regularization term eq.(3.39) can be rewritten as,

$$
\begin{aligned}
\mathcal{R}_{\mathcal{T}}^{l} =& \sum_{p=1}^{T_l-1} \sum_{q=p+1}^{T_l} w_{p,q}^{l} \|\hat{\mathbf{y}}_{p}^{l} - \hat{\mathbf{y}}_{q}^{l}\|_2^2 \\
=& \frac{1}{2} \sum_{p=1}^{T_l} \sum_{q=1}^{T_l} w_{p,q}^{l} \|\hat{\mathbf{y}}_{p}^{l} - \hat{\mathbf{y}}_{q}^{l}\|_2^2 \\
=& \sum_{p=1}^{T_l} \sum_{q=1}^{T_l} \sum_{j=1}^{n} w_{p,q}^{l} (\hat{y}_{j,p}^{l})^2 - w_{p,q}^{l} \hat{y}_{j,p}^{l} \hat{y}_{j,q}^{l} \\
=& \sum_{j=1}^{n} \hat{\mathbf{Y}}^{l}(j,:)(D^{l} - W^{l}) \hat{\mathbf{Y}}^{l}(j,:)^{\top} \\
=& Tr[C\mathbf{Z}^{l} L^{l} (\mathbf{Z}^{l})^{\top} C^{\top}]
\end{aligned}
\tag{3.40}
$$

where $\hat{\mathbf{Y}}^{l}(j,:)$ represents the jth row of matrix $\hat{\mathbf{Y}}^{l}$. L^{l} is the $T_l \times T_l$ Laplacian matrix for the lth MTS sequence, $L^{l} = D^{l} - W^{l}$. D^{l} is a diagonal matrix with the pth diagonal element $D_{p,p}^{l} = \sum_{q=1}^{T_l} w_{p,q}^{l}$. W^{l} is the smoothing coefficient matrix among T_l observations and $w_{p,q}^{l}$ represents the (p,q)th element in W^{l}.

We apply the temporal smooth regularization to each MTS sequence in the training data, which leads to the following compact form of regularization:

$$
\mathcal{R}_{\mathcal{T}} = \sum_{l=1}^{N} \mathcal{R}_{\mathcal{T}}^{l} = Tr[C\mathbf{Z}P\mathbf{Z}^{\top}C^{\top}] \tag{3.41}
$$

where P is the $T \times T$ block diagonal matrix with N blocks and the lth block component is the Laplacian matrix L^{l} for lth MTS sequence.

3.2.5.2 Learning We incorporate the temporal smoothing regularization (eq.(3.41)) into the objective function (eq.(3.24)). Here similarly to gLDS-ridge approach, we set $\mathcal{R}_C(C)$, $\mathcal{R}_A(A)$, and $\mathcal{R}_{\mathbf{Z}}(\mathbf{Z})$ to the ridge regularizations (square of Frobenius norm), which leads to the following new learning objective function:

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\|C\|_F^2 + \beta\|\mathbf{Z}\|_F^2 + \gamma\|A\|_F^2 + \delta Tr[C\mathbf{Z}P\mathbf{Z}^\top C^\top] \quad (3.42)$$

We follow the gLDS model learning algorithm (Algorithm 4) and we optimize eq.(3.42) in a coordinate descent fashion. Since the temporal smoothing regularization only involves C and $\mathbf{Z}$, the update rules for A, R, Q, $\boldsymbol{\xi}$ and $\boldsymbol{\Psi}$ remain the same. The update rules for C and $\mathbf{Z}$ are shown as follows:

$$\hat{C} = (\mathbf{Y}\mathbf{Z}^\top)(\mathbf{Z}\mathbf{Z}^\top + \delta\mathbf{Z}P\mathbf{Z}^\top + \alpha I_d)^{-1} \quad (3.43)$$

$$\hat{\mathbf{z}}_1^l = \left(\Gamma_1^l + \lambda A^\top A\right)^{-1}\left(\Phi_1^l + \lambda A^\top \mathbf{z}_2^l\right) \quad (3.44)$$

$$\hat{\mathbf{z}}_i^l = \left(\Gamma_i^l + \lambda A^\top A + \lambda I_d\right)^{-1}\left(\Phi_i^l + \lambda A^\top \mathbf{z}_{i+1}^l + \lambda A\mathbf{z}_{i-1}^l\right) \quad (3.45)$$

$$\hat{\mathbf{z}}_{T_l}^l = \left(\Gamma_{T_l}^l + \lambda I_d\right)^{-1}\left(\Phi_{T_l}^l + \lambda A\mathbf{z}_{T_l-1}^l\right) \quad (3.46)$$

where

$$\Gamma_i^l = (1 + \delta L_{i,i} - \delta W_{i,i})C^\top C + \beta I_d \quad (3.47)$$

$$\Phi_i^l = C^\top \mathbf{y}_i^l + \delta C^\top C \sum_{j\neq i} W_{i,j}\mathbf{z}_j^l \quad (3.48)$$

3.2.6 Experiments

In this experiments, we (1) qualitatively illustrate the prediction results for gLDS-smooth model; (2) quantitatively show superior predictive performance of our models (gLDS-ridge and gLDS-smooth) compared with traditional LDS learning algorithms (EM and spectral algorithms); (3) stability and sparsification effects achieved by our models. Experiments are conducted on four real-world data sets across different domains. The hyper parameters (α, β, λ and γ) used in our methods are selected (in all experiments) by the internal cross validation approach while optimizing models' predictive performances.

3.2.6.1 Data

Flour price data (*flourprice*). It is a monthly flour price indices data from [Reinsel, 2003], which contains the flour price series in Buffalo, Minneapolis and Kansas City, from August 1972 to November 1980.

Evap data (*evap*). The evaporation data contains the daily amounts of water evaporated, temperature, and barometric pressure from 10/11/1692 to 09/11/1693 [Halley, 1694].

H2O evap data (*h2o_evap*). It contains six MTS variables: the amount of evaporation, total global radiation, estimated net radiation, saturation deficit at max temperature, mean daily wind speed and saturation deficit at mean temperature [Krishnan and Kushwaha, 1973].

Clinical data (*clinical*). The same data as used in Section 3.1.

In order to get comprehensive evaluations of the proposed methods, we vary both the training sizes and the number of hidden states. For *flourprice*, *evap* and *h2o_evap* datasets, we conduct the experiments on both 80% and 90% data for training and use both 5 and 10 as the hidden state size. For *clinical* data, we have randomly selected 100 patients out of 500 as a test set and used the remaining 400 patients for training the models and vary the hidden states from 10 to 30 with a step increase of 5.

3.2.6.2 Results

Qualitative Prediction Analysis We qualitatively show the prediction effectiveness

of our gLDS-smooth model. Figure 14 shows the predictions results for the flour price series in Buffalo. 80% of the MTS is used for training and 20% is used for testing. As we can see from Figure 14, the gLDS-smooth is able to well capture the ups and downs of the time series and make the accurate predictions. Prediction results for Minneapolis and Kansas City are listed in Appendix G.

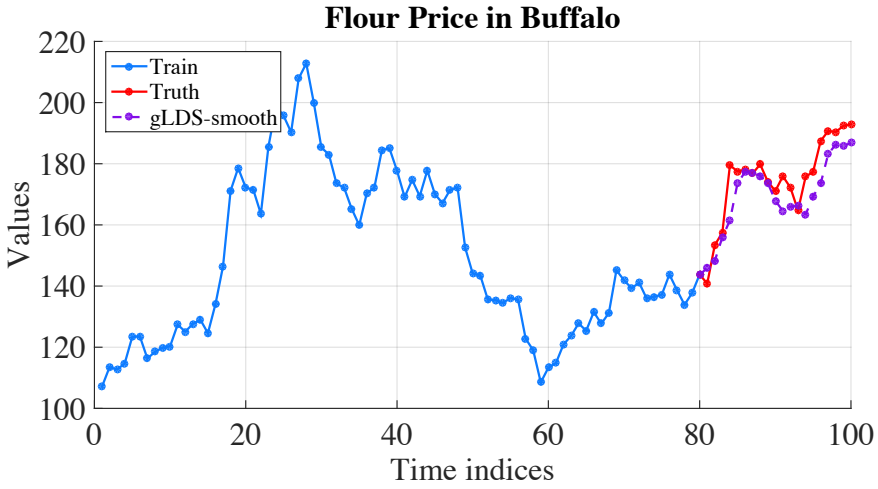


Figure 14: Predictions for flour price series in Buffalo by using gLDS-smooth.

Quantitative Prediction Analysis We quantitatively compute and compare the prediction accuracy of the proposed methods (gLDS-ridge and gLDS-smooth) with the standard LDS learning approaches: EM and spectral algorithms on four real-world data sets. The results are shown in Tables 5 - 8. As we can see, the gLDS-ridge and gLDS-smooth methods perform significantly better than all the other methods. Furthermore, due to the smooth effect of the temporal smooth regularization, gLDS-smooth supports better predictions than gLDS-ridge, which translates to the best predictive performance.

Stability Effects of gLDS-stable Similarly to [Boots et al., 2007], we show the stability effects of the gLDS-stable model learned using our framework by generating the simulated sequences in the future. The long sequence simulation results of *evap* are shown in Figure 15. We can see in Figure 15 that the LDS models learned from EM and spectral algorithms fail to guarantee the system stability and the generated values of simulated long-term sequences may diverge. In contrast to this, the sequences generated by our gLDS framework with

Table 5: Average-MAPE results on *flourprice* dataset.

	Training: 80%		**Training: 90%**	
# of states	5	10	5	10
Spectral	6.25	5.86	6.61	5.93
EM	3.62	4.15	3.63	3.94
gLDS-ridge	3.37	3.14	3.29	2.82
gLDS-smooth	**3.24**	**2.71**	**2.86**	**2.50**

Table 6: Average-MAPE results on *evap* dataset.

	Training: 80%		**Training: 90%**	
# of states	5	10	5	10
Spectral	24.62	24.85	25.08	26.28
EM	17.68	14.45	16.32	17.35
gLDS-ridge	10.58	10.35	13.60	14.05
gLDS-smooth	**10.35**	**10.27**	**13.39**	**13.68**

Table 7: Average-MAPE results on *h2o_evap* dataset.

	Training: 80%		**Training: 90%**	
# of states	5	10	5	10
Spectral	36.26	32.20	13.73	15.88
EM	39.53	68.68	17.33	17.46
gLDS-ridge	27.97	28.53	16.12	14.42
gLDS-smooth	**26.38**	**26.46**	**14.01**	**14.08**

Table 8: Average-MAPE results on *clinical* dataset.

# of states	10	15	20	25	30
Spectral	6.29	6.24	6.32	6.04	6.00
EM	3.97	3.54	3.54	3.53	3.53
gLDS-ridge	3.22	3.21	3.21	3.21	3.21
gLDS-smooth	**3.21**	**3.20**	**3.20**	**3.19**	**3.19**

incorporated stability constraints are stabilized. Stability effects of gLDS-stable model for *fourprice*, *h2o_evap* and *clinical* data sets are shown in Appendix H.

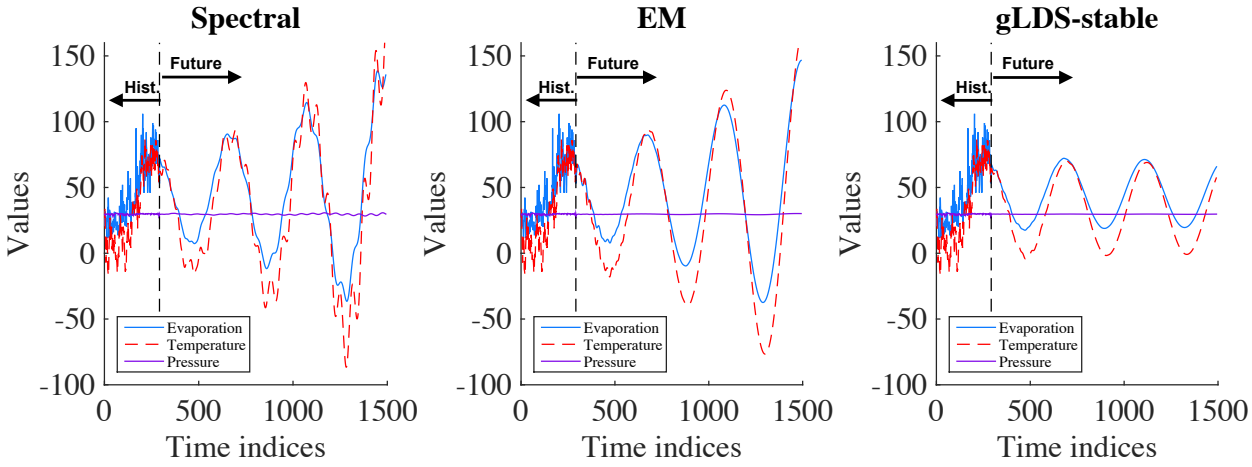


Figure 15: Training data and simulated sequences from gLDS-stable model in *evap* data. "Hist." represents the historical observations and "Future" represents the sequence generated by LDS.

Sparsification Effects of gLDS-low-rank Figure 16 shows the sparsification effects of the gLDS-low-rank model learned using our framework. As we can see, similar to the experimental results in Section 3.1.3.4, gLDS-low-rank model is able to shut down unnecessary dimensionality and find the intrinsic dimension of the hidden state space. Additional results

on the low-rank model are shown in Appendix I.

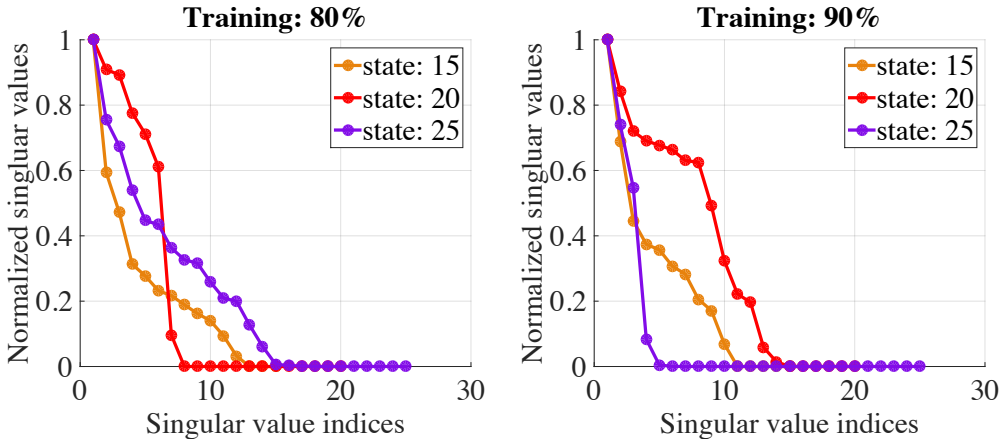


Figure 16: Intrinsic dimensionality recovery in *flourprice* data.

3.3 SUMMARY

In this chapter, we built forecasting models from regularly sampled multivariate time series data. Our work focused on the refinements of LDS, which is a popular model for real-valued MTS analysis. We did this in two ways. First, we introduced a novel regularized LDS learning framework for short-span MTS modeling, which automatically seeks the intrinsic state dimensionality and is robust in preventing model overfitting even for a small amount of MTS data. To learn the regularized LDS from data we incorporated proximal gradient descent methods into the MAP framework and used EM to obtain a low-rank transition matrix of the LDS model. We proposed three priors for modeling the matrix which lead to three instances of our rLDS. Experimentally, we demonstrated that rLDS is able to find the intrinsic dimensionality on a synthetic data set and prefers lower hidden state space on a real-world clinical time series data set even with a large initial state space. We showed that rLDS outperforms other state-of-the-art LDS learning approaches in terms of MAPE.

To further achieve expected properties of dynamic models when learning the models from an MTS collection, we presented a generalized LDS learning framework based on matrix fac-

torization, which is different from traditional EM learning and spectral learning algorithms. In gLDS, each MTS sequence is factorized as a product of a shared emission matrix and a sequence-specific state dynamics, where an individual hidden state sequence is represented with the help of a shared transition matrix. Compared to the traditional LDS learning algorithms, the advantages of our gLDS framework are: (1) the LDS models can be learned efficiently from multiple MTS sequences; (2) constraints on both the hidden states and the parameters can be easily incorporated into the learning process; (3) it is able to support accurate MTS prediction. Furthermore, we proposed a novel temporal smoothing regularization for learning the LDS models to stabilize the model learning and its predictions. In our experimental evaluation on four real-world data sets, we showed that (1) gLDS is able to achieve better time series predictive performance when compared to other LDS learning algorithms; (2) the proposed temporal smoothing regularization encourages more stable and accurate predictions; and (3) constraints can be directly integrated in the learning process and special designed system properties such as stability, low-rankness can be easily achieved.

4.0 LEARNING HIERARCHICAL DYNAMICAL SYSTEMS FROM IRREGULARLY SAMPLED UNIVARIATE TIME SERIES

In this chapter, we tackle the problem of modeling irregularly sampled univariate time series. While LDS and GP models we reviewed in Chapter 2 can be adapted to model irregularly sampled clinical time series data, they come with drawbacks that may limit their performance. More specifically, discrete time models are not able to represent well sequences of values in real time because values need to be re-estimated from quantities with a discrete time step. On the other hand, a continuous time GP model with a constant mean function is too restrictive and cannot model the different modes of dynamics or different subpopulations of patients well. On the positive side, discrete time models, especially LDS, are good at modeling changes in both the dynamics and different modes in time series behavior, while GP models are good at modeling time series in real time. Considering the respective advantages and limitations of the two frameworks, a combination of the two appears as the best solution to offset their limitations. To follow the above intuition, we propose a new hierarchical dynamical system model that splits the process into a sequence of dependent local GPs that are combined with LDS to better capture higher-level changes in the time series dynamics. The local GPs' dependencies naturally account for the transitions of mean functions and irregular samples are handled by the local GPs themselves. We note that the material presented in this chapter was originally published as [Liu et al., 2013, Liu and Hauskrecht, 2013, Liu and Hauskrecht, 2015b].

4.1 THE HIERARCHICAL DYNAMICAL FRAMEWORK

In this section, we develop a two-layer hierarchical dynamical model that lets us represent the irregularly sampled time series information in a more flexible manner. The key structure of the model is shown in Figure 17. Briefly, the model consists of two hierarchically related processes: the GP and the LDS. The GP is restricted to a time window of a finite duration and it is used to represent time series and its changes for shorter time spans. Longer-term process changes are modeled and controlled by the LDS. As we can see from Figure 17, in the lower layer, which is shown in a dashed line box, we transform the entire irregular time series data into M windows using a predefined window size and a predefined overlap size. Each window s in Figure 17 relates observations $\mathbf{Y}(s) = \{y_1(s), y_2(s), \ldots, y_{N_s}(s)\}$ using the same window-specific GP and N_s is the number of observations in the sth window, $s = 1, \cdots, M$. $\mathbf{Y}(s)$ represents all the observations fall in the sth window and $y_i(s)$ is the ith observation in the sth window. Hence, instead of using a single GP, we capture a time series by using many different window-specific local GPs and model global changes in dynamics using the upper level LDS that controls the means of the window specific GPs. That is, the LDS represents the dynamics and changes of summary statistics $\{\gamma_s\}_1^M$ defining individual $\{GP_s\}_1^M$. The upper level LDS is defined as:

$$\mathbf{z}_s = A\mathbf{z}_{s-1} + \boldsymbol{\epsilon}_s \tag{4.1}$$

$$\gamma_s = C\mathbf{z}_s + \boldsymbol{\zeta}_s \tag{4.2}$$

where summary statistics γ_s acts like observations $\mathbf{y}_i$ in eq.(2.2) and $A, C, Q, R, \mathbf{z}_s, \boldsymbol{\epsilon}_s, \boldsymbol{\zeta}_s$ are similarly to regular LDS introduced in Section 2.2.1.

As mentioned in Section 2.2.2, the covariance function measures the similarity of two function values $f(t)$ and $f(t')$ based on their input time t and t'. In general, the covariance function should reflect the properties of the modeled time series, such as its smoothness or periodicity which leads to the important question: *how do we pick the covariance functions that work well with clinical time series data?* In order to model covariances of clinical time series for numerical labs we can make the following assumption: the readings made at times

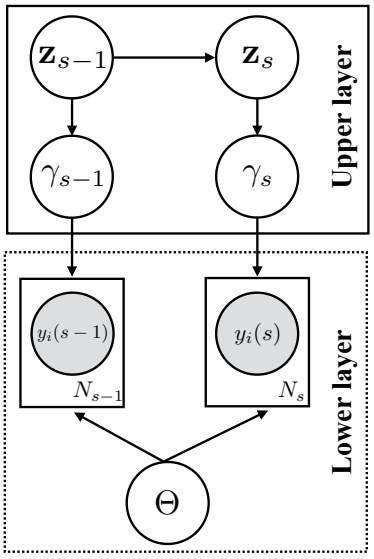


Figure 17: The graphical illustration of the hierarchical dynamical model combining the GP and the LDS. The shaded nodes denote irregular observations. The γ node is the window representative we extract from the corresponding window and the z node is the hidden state we introduce in LDS to model the change of γs. The Θ node represents the shared covariance function parameters for all the GPs.

t and t' which are close are likely to have similar reading values $f(t)$ and $f(t')$. Examples of covariance functions that represent this assumption are the Gaussian kernel eq.(4.3) and the mean reverting kernel eq.(4.4):

$$K(t,t') = \sigma_1 \exp\left(\beta_1 (t-t')^2\right) \tag{4.3}$$

$$K(t,t') = \sigma_2 \exp\left(\beta_2 |t-t'|\right) \tag{4.4}$$

The Gaussian kernel is the most frequently used kernel in literature [Brahim-Belhouari and Bermak, 2004, Girard et al., 2003, Wang et al., 2005] that promotes smoothness and pushes two different readings closer when they are close in time. The second kernel represents the mean reverting process and while it forces the two readings close in time to be similar, it also permits more abrupt changes in their observed values [Rasmussen and Williams, 2006, Chapter 4]. To approximate the clinical time series in this work we use a linear combination of eq.(4.3) and eq.(4.4) together with the observational noise component $\epsilon \sim \mathcal{N}(0, \sigma^2)$ (see Section 2.2.2) as our covariance function:

$$K(t,t') = \sigma_1 \exp\left(\beta_1 (t-t')^2\right) + \sigma_2 \exp\left(\beta_2 |t-t'|\right) + \sigma^2 \delta_{t,t'} \tag{4.5}$$

In this model, $\Theta = \{\sigma_1, \beta_1, \sigma_2, \beta_2, \sigma\}$ are parameters of the covariance function that can be learned directly from data. $\delta_{t,t'}$ is a Kronecker delta which is one iff $t = t'$ and zero otherwise.

4.1.1 Learning

We learn the parameters of our hierarchical dynamical model by devising solutions to two estimation/learning problems: (1) learning of the parameters Θ of the covariance function defining the lower level GPs, and (2) learning of the parameters of the upper level LDS.

4.1.1.1 Estimation of The Covariance Function

Since all window-specific $\{GP_s\}_1^M$ share the same covariance function, we set Θ by maximizing the likelihood using the partial derivative of the likelihood with respect to each parameter θ_i in Θ as defined in eq.(2.11).

4.1.1.2 Estimation of The LDS Parameters

The LDS controls the means of individual window-specific GPs. We learn its parameters as follows:

Step 1. Use window-based segmentation (WbS) (Section 2.2.1.3) approach to estimate each summary statistics γ_s from observations in windows s respectively. The γ_s represents the mean of window-specific GP_s. In general, there are many different ways to estimate γ_s. Let h denote a function used for estimating the mean of the GP from observations $\mathbf{Y}(s)$ in the sth window. Examples of h can be $max, mean$ or $last$ functions that return the maximum, the mean, or last observed value in the window. In this work, we use the mean function as the estimator of window-specific GP means.

Step 2. Use sequences of $\{\gamma_s\}_1^M$ statistics as observations of the upper level LDS in our hierarchical dynamical system. To learn the parameters of the LDS, we use the EM learning algorithm to iteratively re-estimate the parameters $\Lambda = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$ defining the LDS [Ghahramani and Hinton, 1996], similarly to standard LDS learning.

4.1.2 Prediction

Once the hierarchical dynamical system is learned from the training data we would like to use it to support predictions on future time series. Given the initial observations $\mathbf{Y}_o$ and an arbitrary future time t^*, the value $\mathbf{y}_{t*}$ is predicted as follows:

Step 1. Split $\mathbf{Y}_o$ into windows and continue splitting time after $\mathbf{Y}_o$ into windows until one contains t^*. The index of the window containing t^* is λ and the index of the window containing the last observation in $\mathbf{Y}_o$ is τ.

Step 2. Estimate summary statistics γ_ss for all windows up to window τ using $\mathbf{Y}_o$ using the WbS approach. After that use these statistics to predict γ_λ with the upper level LDS system.

Step 3. Compute the value $\hat{\mathbf{y}}_{t*}$ at future time t^* using the posterior mean of the GP with the mean function γ_λ, covariance parameters Θ and past observations $\mathbf{Y}_o$.

4.2 EXPERIMENT

4.2.1 Baselines

We compare our two-layer hierarchical dynamical system approach with LDS and GP layers (HDSGL) to six baseline methods:

1. First-order autoregressive (AR) process [Box and Pierce, 1970, Makridakis and Hibon, 1997] trained on the entire time series using DVI approach. We applied linear interpolation directly to fill the missing values.

2. Linear dynamical system (LDS) trained on the entire time series using DVI approach. We applied linear interpolation directly to fill the missing values.

3. Standard Gaussian process regression (GP) with constant mean function. The choice of covariance function is the linear combination of eq.(4.3) and eq.(4.4).

4. Window-based AR (WAR). Irregular sampled time series is handled by WbS, as described in Section 2.2.1.3. It splits the time series first into windows and, after that, it trains an AR over the windows' summary statistics.

5. Window-based LDS (WLDS). Irregularly sampled time series is handled by WbS, as described in Section 2.2.1.3. It splits the time series first into windows and, after that, it trains an LDS over the windows' summary statistics.

6. Hierarchical dynamical system combined with GP and AR process (HDSGA). HDSGA is similar to HDSGL, but we change the upper layer LDS to the first order AR process.

We set the summary statistics estimation function h that is used to calculate the summary statistics γ_s for each window s for all WbS approaches (WAR, WLDS, HDSGA and HDSGL) to the mean of the values observed in that window. Also, we use the combination of the Gaussian and the mean reverting kernels as the covariance function for all GP related methods (GP, HDSGA, HDSGL).

4.2.2 Evaluation Metrics

Our objective is to test the predictive performance of our approach by its ability to predict the future value of an observation for a patient for some future time given a sequence of patient's past observations. We judged the quality of the prediction using the Mean Absolute Error (MAE) on multiple test data predictions. Instead of Root Mean Square Error (RMSE), which gives a relatively high weight to large errors (the errors are squared before they are averaged), MAE is the average over the absolute values of the differences between predictions and the corresponding observations. The MAE is a linear score which means that all the individual differences are weighted equally in the average. More specifically, the MAE is defined as follows:

$$MAE = \left[N^{-1} \sum_{i=1}^{N} |y_i - \hat{y}_i| \right] \tag{4.6}$$

where N is the total number of predictions. y_i and $\hat{y}_i$ are the true and predicted values.

4.2.3 Data

We have tested our new approach on time series data obtained from electronic health records of 4,486 post-surgical cardiac patients stored in PCP database [Hauskrecht et al., 2010a, Valko and Hauskrecht, 2010, Nguyen et al., 2014]. We used ten tests from the CBC panel to learn ten different time series models, and evaluated them on the time series prediction task. The ten tests, their means and standard deviations, are listed in Table 9. These time series data are noisy; their signals fluctuate in time, and the time periods between observations vary. Figure 18 illustrates such a time series for one of the patients. The X-axis is the time index aligned by hour and the Y-axis are normalized values/observations for each test.

To test the performance of our prediction model, we have randomly selected 1000 patients that had at least 10 CBC tests ordered during their hospitalization. Among the 1000 patients selection, we randomly divided patients and their time series into the training and testing sets, such that data for 200 patients form the test data and time series data for 800 patients were used for training.

Table 9: Ten lab tests from the CBC panel.

Lab Test Name	Unit	Mean	Std	Average Length
White blood cell (WBC)	$10^9/L$	11.98	6.08	24.28
Hematocrit (HCT)	%	28.67	4.73	54.45
Hemoglobin (HGB)	g/dL	9.59	1.67	37.71
Mean corpuscular HGB concentration (MCHC)	g/dL	33.86	0.81	24.17
Mean corpuscular hemoglobin (MCH)	pg/cell	30.54	1.76	24.17
Mean corpuscular volume (MCV)	fL	90.17	4.55	24.18
Mean platelet volume (MPV)	fL	8.73	1.18	23.25
Platelet (PLT)	$10^9/L$	202.07	126.73	27.05
Red blood cell (RBC)	$10^{12}/L$	3.21	0.56	24.25
Red cell distribution width (RDW)	%	16.77	2.64	24.01

4.2.4 Results

In this section, we choose two lab prediction results (MCV and RBC) from Table 9 to highlight the main findings from the experimental evaluation in that work. The complete results are shown in Appendices J - L. To conduct the evaluation, we use the test dataset to generate various prediction tasks as follows. For each patient l and complete time series for that patient, we calculate the number of observations T_l in that time series. We use T_l to generate all different pairs of indices (ψ, ϕ) for that patient, such that $1 \leq \psi < \phi \leq T_l$, where ψ is the index of the last observation assumed to be seen, and ϕ is the index of the observation we would like to predict. By adding time stamp reading to each index, the two indices help us define all possible prediction tasks, we can formulate on that time series. Let Γ_l be the total number of different indices pairs (or Γ_l different prediction tasks) for patient l and $\sum_{l=1}^{200} \Gamma_l$ is total number of prediction tasks in our test data. For each method, we use the MAE on these tasks to judge the quality of test predictions and run the pairwise t-test on the $\sum_{l=1}^{200} \Gamma_l$ prediction tasks' results from our method and all the other baselines to check the statistical differences between them. In addition, we use the bootstrap approach [Felsenstein, 1985] to compute the 95% the confidence interval on MAE for each method. We would also like to note that the hyper parameters (window size $\mathcal{W}$ and overlap size $\mathcal{O}$) used in our

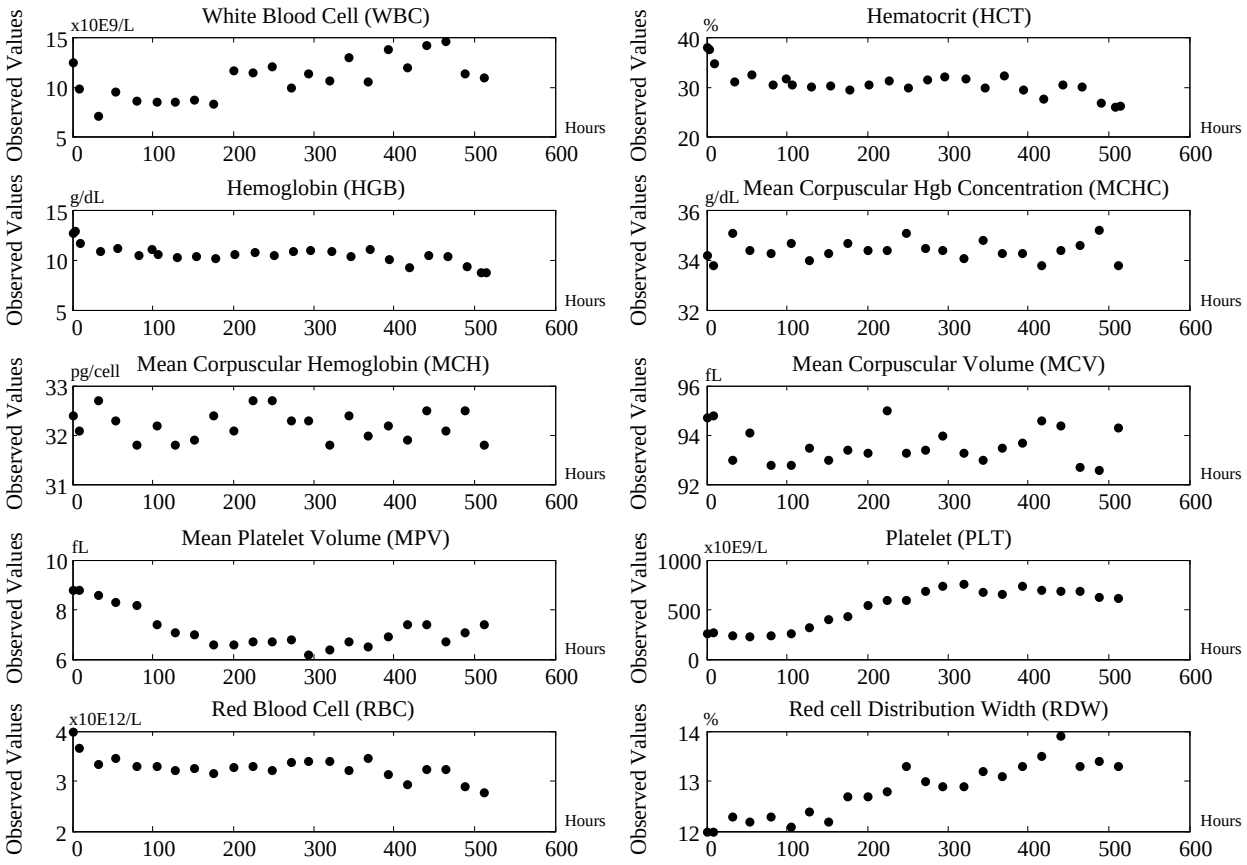
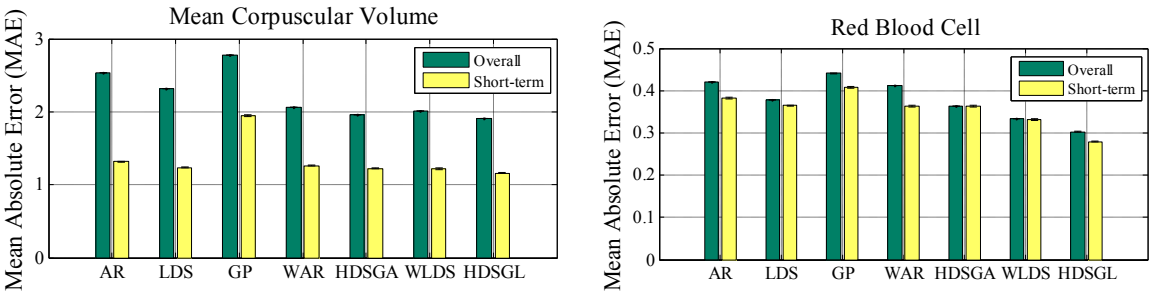


Figure 18: Time series for ten tests from the CBC panel for one of the patients.

methods are selected (in all experiments) by the internal cross validation approach while optimizing models' predictive performances.

4.2.4.1 Overall Prediction Performance In the overall prediction experiment, we follow the procedure described above to generate and randomly select different prediction tasks. These contains both short-term and long-term predictions depending on the difference in between the time at which we predict the value and the time of the last observation seen. Figure 19 shows the results of the prediction experiments on MCV and RBC for all methods. The results of our experiments (dark green bars in Figure 19) show that our hierarchical dynamical system (HDSGL) outperforms all other methods in terms of prediction error on the CBC test data. The results are statistically significantly different at 0.05 level for all labs. We determined the significance by running the pairwise t-test comparing the HDSGL to all other methods on all corresponding prediction tasks. We believe the main reason for the hierarchical approach outperforming all other methods is that it directly models and works with real time series (via lower level GP) and that it minimizes the effect of noisy observations by using window-based summary statistics. The hidden states of the upper layer LDS are able to capture the change of those summary statistics. The lower layer GP can adjust the prediction values based on the mean of the upper layer and the few observations we have, which gives us the lowest MAE.

4.2.4.2 Short-term Prediction Performance We expect that short-term predictions that are close to the last value observed should be better. To verify this expectation, we conduct a new experiment where observation indices for the prediction tasks involve (ψ, ϕ) pairs that satisfy $\phi = \psi + 1$, that is, we always try to predict the next lab reading. Figure 19 compares our method and the baselines in terms of their corresponding overall and short term prediction performances. As we can see from Figure 19, short-term predictions are much better than overall predictions (that include both short and long term prediction), which supports our intuition that the further we predict, the worse predictions we make. In addition, we see our method remains the best in all the lab tests for the short term prediction tasks.



(a) MAE on MCV test samples.

(b) MAE on RBC test samples.

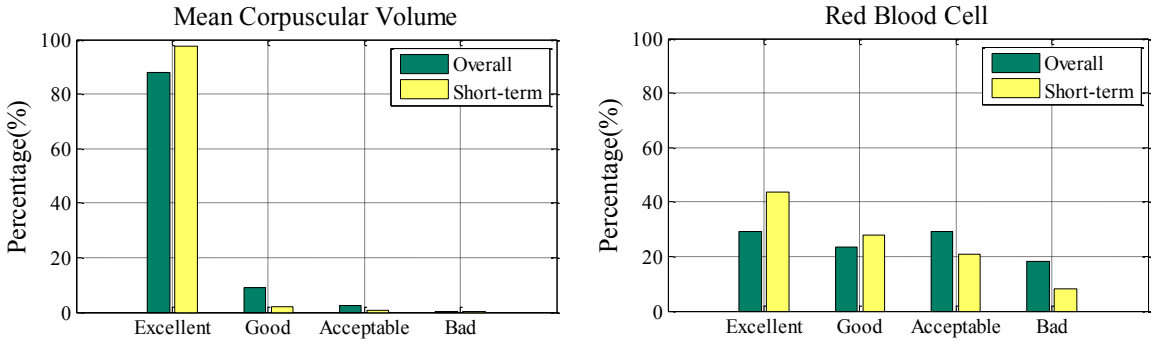
Figure 19: MAE on MCV and RBC test samples for random prediction tasks.

4.2.4.3 Clinical Expert Evaluation In this experiment, we examine whether the predictions made by our model (HDSGL) are clinically acceptable or not. In order to assess the clinical relevance of predictions, we consulted a clinical expert, and adopted his suggestion to judge the importance of prediction error by using Absolute Percentage Error (APE), which measures the prediction deviation relative to its true value and is defined as $APE = |y_i - \hat{y}_i|/y_i \times 100\%$. After calculating the APE for each prediction, we categorize its result into four qualitative categories suggested by the expert (shown in Table 10). These four categories tell us how well the model is able to predict the lab values in terms of their clinical acceptance. We use these four categories to calculate the distribution of predictions for each lab test in terms of both overall and short-term predictions. Figure 20 summarizes the distributions of these qualitative prediction categories for both MCV and RBC lab tests. In terms of clinical acceptance, we see that the results differ widely for the different labs. In particular, very good short and long term predictions are achieved for CBC lab components that are less sensitive to blood loss and drip infusions that are rather frequent during the management of post-surgical cardiac patients, such as MCV. On the other hand, some labs are volume based counts and hence are sensitive to the above events, such as RBC. Consequently the prediction quality of these models goes down. Overall, the results for these labs suggest the predictions based only on previous sequences of lab values alone may not be sufficient, and additional variables representing the different future events and/or possible

patient management steps should be included in the model to improve its prediction quality.

Table 10: Clinical acceptance categories.

Category	Excellent	Good	Acceptable	Bad
APE Range	$\leq 5\%$	5% - 10%	10% - 20%	$\geq 20\%$



(a) Clinical evaluations on MCV.

(b) Clinical evaluations on RBC.

Figure 20: Clinical evaluations of HDSGL on MCV and RBC for both overall prediction and short-term prediction quality distributions.

4.3 SUMMARY

In this chapter, we proposed and developed a novel hierarchical framework for modeling clinical time series data of varied length and with irregularly sampled observations. Our hierarchical dynamical system framework for modeling clinical time series combines advantages of the two temporal modeling approaches: the LDS and the GP. We modeled the irregularly sampled clinical time series by using multiple GP sequences in the lower level of our hierarchical framework and captured the transitions between GP by utilizing the LDS. Compared to traditional LDSs and modern GP regression, the new system adapts better to irregular sampling and it is more accurate when making predictions for different future times.

Experimentally, we tested our framework on a complete blood count (CBC) panel data of 1000 post-surgical cardiac patients during their hospitalization. We first learned the time series model from data for the patients in the training set, and then used it to predict future time series values for the patients in the test set. We showed that our model outperforms multiple existing models in terms of its predictive accuracy. Our method achieved a 3.13% average prediction accuracy improvement on ten CBC lab time series when it was compared against the best performing baseline. A 5.25% average accuracy improvement was observed when only short-term predictions were considered. Thus, our new hierarchical dynamical system framework is able to let us model irregularly sampled time series data and it is a promising new direction for modeling clinical time series and for improving their predictive performance.

5.0 LEARNING PERSONALIZED PREDICTIVE MODELS FROM IRREGULARLY SAMPLED MULTIVARIATE TIME SERIES

Building of an accurate predictive model of clinical time series for a patient is rather challenging due to the characteristics of clinical MTS and the computational and modeling trade-offs arising from them. Briefly, when the time series of past observations for the patient are short, it may be hard to learn a patient specific model from the patient's own data, and the population based model may be a better option. On the other hand, when the observed data for the target patient are sufficiently long, a patient specific time series model learned from patient's own data may better reflect the future behavior. Moreover, short-term variability and deviations from typical behaviors may prefer models that can adapt quickly to just a few recent observations. Overall, the prediction model should provide flexible and customized predictions for each new patient given his or her current health condition, and should benefit from what is known about other patients when the patient specific model is not available. The majority of existing approaches proposed for clinical MTS prediction in the literature are not able to cover all necessary model behaviors. They either build a population based model or a patient specific model ignoring what is known about the population.

In this chapter, we study and develop methods to address the above problems by building personalized clinical time series prediction models that better mimic patient specific temporal behaviors and variations. More specifically we develop two frameworks that can predict future values of real-valued MTS for a patient given his or her past observations, as well as, time series data for any past patient. This breaks down this chapter into two main sections. In Section 5.1, we build a personalized prediction model via model adaptation in which we first learn the population trend from clinical MTS sequences from many different patients and then we model patient specific residuals (or differences in between predictions made by

the population model and actually observed values) individually. In Section 5.2, we develop an adaptive forecasting framework via adaptive model selection approach. At any point in time, it selects the most promising time series model out of the pool of many possible models, and consequently, combines advantages of the population, patient specific and short-term individualized predictive models. Both proposed forecasting models are evaluated on a real-world clinical time series data set. The results demonstrate that our approaches are superior on the prediction tasks for multivariate, irregularly sampled clinical time series, and they are able to outperform predictions based on pure population and patient specific models, as well as, other patient specific model adaptation strategies. We note that the material of our model adaptation based personalized prediction model presented in Section 5.1 was originally published as [Liu and Hauskrecht, 2016a].

5.1 PERSONALIZED PREDICTION VIA MODEL ADAPTATION

We develop a new approach to support adaptive prediction for clinical time series by using model adaptation methods. Model adaptation methods try to bridge a possible gap in between population based models and the target patient by adjusting the population based model to fit better the specific patient. In general, model adaptation methods can be realized in different ways, as reviewed in Section 2.3.2.

In this work, we conduct the model adaptation on clinical MTS data by building a two-stage adaptive forecasting model. Our approach involves two stages: it first learns a population based model from collection of time series data of varying lengths and then builds patient specific models from the patient specific residuals. Residuals are the difference between the patient observations and the predictions from the population based model. In such a way, our method benefits from the population trend extracted from past data collection and at the same time adapt to patient specific data, thus allowing one to make more accurate MTS predictions.

5.1.1 Learning

In this section, we describe the learning procedure for our two-stage adaptive forecasting model that (1) is learned from a collection of time series data of varying lengths; (2) captures the patient specific short-term multivariate interactions.

5.1.1.1 Stage 1: Learning A Population Model In the first stage, we would like to learn a population model from all available data sequences to represent the trend of the entire population. We choose the regularized LDS model (Section 3.1) to model the population trend, which is able to choose the optimal number of hidden states and prevent the overfitting problem and support accurate MTS forecasting.

In spite of the advantages of LDS based models, they are restricted to discrete time domain where observations are regularly sampled. In order to apply the discrete time LDS model over our irregularly sampled clinical data, we follow [Adorf, 1995, Dezhbakhsh and Levy, 1994, Åström, 1969, Bellazzi et al., 1995, Kreindler and Lumsden, 2006, Rehfeld et al., 2011, Liu and Hauskrecht, 2015a] and apply the DVI technique (Section 2.2.1.3) to discretize each irregularly sampled clinical sequence and that replaces it with a regularly sampled time series data.

The DVI approach assumes that all observations are collected regularly with a pre-specified sampling frequency r. However, instead of actual readings, the values at these time points are estimated from readings at time points closest to them using various interpolation techniques. The interpolated (regular) time series, i.e., $\tilde{\mathbf{y}}_i^l$, is then used to train a discrete-time LDS model. We put a tilde sign ($\tilde{\cdot}$) over $\mathbf{Y}^l$ and $\mathbf{y}_i^l$ to indicate the discretized observations. $\tilde{T}_l$ is the length of discretized sequence for patient l.

A possible limitation of the DVI data transformation is possible information loss: as we can see from Figure 5, some observations in individual time series are discarded during this discretization process. However, given that LDS is building a coarse level population model over the entire collection of data (many patients), this loss is less important. We also note that patient specific observations are not discarded in the second stage of our approach that captures fine grained patient specific multivariate interactions by MTGP.

Once we obtain the entire discretized clinical sequences $\{\hat{\mathbf{Y}}^l\}_{l=1}^N$, we apply the standard LDS EM learning algorithm to learn the unified population based model.

5.1.1.2 Stage 2: Learning Multivariate Interaction Models A population model built from a collection of clinical data for multiple patients is crucial since each individual sequence is usually very short. The learned model from the entire population is more robust and stable. However, the prediction task is performed patient by patient and the forecasting model should also reflect and take into account the variations specific to the current patient. To address this problem, we model the patient specific multivariate interactions by using an MTGP (Section 2.2.3). More specifically, instead of simply modeling the clinical time series trends (the mean function of MTGP) by using constants or simple known parametric forms (e.g., linear functions) [Ghassemi et al., 2015, Durichen et al., 2015], we use the population model (learned in Stage 1) to reflect the time series tendency and build an MTGP on a residual signal that reflects the deviations of patients' true observations and the predictions made by the population LDS model. We define the *multivariate residual time series* as follows:

Definition 1. (Multivariate Residual Time Series) For each patient l, given time series $\mathbf{Y}^l$ and its corresponding predictions $\hat{\mathbf{Y}}^l$ from model Ω, a multivariate residual time series $\mathbf{R}^l$ represents the deviations from $\mathbf{Y}^l$ to $\hat{\mathbf{Y}}^l$, i.e., $\mathbf{R}^l = \mathbf{Y}^l - \hat{\mathbf{Y}}^l$.

Notice that each residual time series $\mathbf{R}^l$ is computed by using the true observations $\mathbf{Y}^l$ (not the discretized sequence $\tilde{\mathbf{Y}}^l$), there is no information loss for each patient under the prediction task and $\mathbf{R}^l$ is irregularly sampled.

The multivariate residual time series reflect each patient's unique variations from the general population and they are distinguished patient by patient. Furthermore, clinical events usually only affect a handful of measurements within a small time window. Hence, for each patient l, we model these transient deviations nonparametrically using an MTGP. The MTGP has mean $\mathbf{0}$ and a squared exponential covariance function, which is the most frequently-used example in literature [Rasmussen and Williams, 2006]. In eq.(2.12), K^G is defined as follows:

$$K^G(t,t') = \alpha \exp\Big(-\frac{(t-t')^2}{2\beta^2}\Big)$$

The complete parameter set Λ in the MTGP model is $\Lambda = \{\alpha, \beta, \delta_i, K^C\}$ where $i = 1, \cdots, n$. In this work, we adopt the Cholesky decomposition and the "free-form" parameterization techniques ($K^C = LL^\top$) to learn the parameter set Λ by minimizing the negative log marginal likelihood via gradient descent [Rasmussen and Williams, 2006, Ghassemi et al., 2015].

Usually the MTGP model has the computation limitation that it has $\mathcal{O}(n^3T^3)$ compared with $n \times \mathcal{O}(T^3)$ for standard GP models (T is the length of the time series). However, this limitation is not as relevant in our application setting, given that the number of clinical observations is very limited and clinical time series are usually short span.

5.1.2 Prediction

In the real clinical setting, a successful forecasting model needs to be *adaptive*, that is, when newly observed values are obtained, the model should efficiently adapt to the new change and utilize new values to make better predictions. In this work, we develop a new adaptive prediction algorithm based on the Kalman filtering algorithm [Kalman, 1960] that utilizes our two-stage forecasting model.

Let u denote the current patient we consider in our prediction task. $\mathbf{Y}^u$ is an $n \times T_u$ matrix which denotes the current observed values for patient u. Given an arbitrary future time stamp $t*$ ($t* > T_u$), the value $\hat{\mathbf{y}}^u_{t*}$ is predicted as follows:

Step 1. Compute the discretized observations $\tilde{\mathbf{Y}}^u$ by using DVI on $\mathbf{Y}^u$.

Step 2. Infer patient specific hidden dynamics by using population based model Ω and $\tilde{\mathbf{Y}}^u$. This step *adaptively* computes the patient specific hidden state $\mathbf{Z}^u$ using patient's latest observations. Details are provided in Appendix A.

Step 3. Make predictions by using the population model Ω and $\mathbf{Z}^u$. Note that we need to predict the value at time points closest to the target time $t*$, and after that, apply the interpolation approach to estimate the target value. The prediction made by the population model is $\hat{\mathbf{y}}^u_{t*}(\Omega)$

Step 4. Use the population model to predict patient u's known observations ($\mathbf{Y}^u$) adaptively, denoting as $\hat{\mathbf{Y}}^u$. Compute the residual time series for patient u, i.e., $\mathbf{R}^u = \mathbf{Y}^u - \hat{\mathbf{Y}}^u$.

Step 5. Learn the MTGP model Λ^u from $\mathbf{R}^u$ to capture the patient specific short-term variability.

Step 6. Predict patient specific short-term variability $\hat{\mathbf{y}}^u_{t*}(\Lambda^u)$ by using Λ^u at the target time $t*$.

Step 7. Compute the final prediction $\hat{\mathbf{y}}^u_{t*}$ by combining $\hat{\mathbf{y}}^u_{t*}(\Omega)$ and $\hat{\mathbf{y}}^u_{t*}(\Lambda^u)$, i.e., $\hat{\mathbf{y}}^u_{t*} = \hat{\mathbf{y}}^u_{t*}(\Omega) + \hat{\mathbf{y}}^u_{t*}(\Lambda^u)$.

5.1.3 Model Learning and Prediction Summary

Algorithm 5 summarizes our two-stage adaptive forecasting model and its learning and prediction parts.

Algorithm 5 Learning and Prediction Procedures

INPUT:

- Train data collection $\mathcal{D} = \{< \mathbf{Y}^l, \mathbf{x}^l >\}$, where $l = 1, \cdots, N$.
- DVI sampling frequency r.
- Number of hidden states in LDS d.
- Current observations $\mathbf{Y}^u$ for patient u who is being predicted.
- An arbitrary future time stamp $t*$ ($t* > T_u$).

PROCEDURE:

// **Stage1**: Learning the population model.
$\{\tilde{\mathbf{Y}}^l\} = DVI(\{\mathbf{Y}^l\}, \{\mathbf{x}^l\}, r)$.
$\Omega = LearnRegularizedLDS(\{\tilde{\mathbf{Y}}^l\})$.
// **Stage2**: Learning the multivariate interaction model.
Compute residual time series $\mathbf{R}^u$.
$\Lambda^u = LearnMTGP(\mathbf{R}^u)$.
// **Adaptive Prediction**: Predicting $\hat{\mathbf{y}}^u_{t*}$ by Ω and Λ^u.
Trend prediction: $\hat{\mathbf{y}}^u_{t*}(\Omega) = PredictLDS(\Omega, t*)$.
Variability prediction: $\hat{\mathbf{y}}^u_{t*}(\Lambda^u) = PredictMTGP(\Lambda^u, t*)$.
$\hat{\mathbf{y}}^u_{t*} = \hat{\mathbf{y}}^u_{t*}(\Omega) + \hat{\mathbf{y}}^u_{t*}(\Lambda^u)$.

OUTPUT: Prediction at time stamp t_*: $\hat{\mathbf{y}}^u_{t*}$.

5.1.4 Experiment

In this section we evaluate our approach on a real-world clinical data set obtained from EHRs of post-surgical cardiac patients in PCP database [Hauskrecht et al., 2010b,Hauskrecht

et al., 2013]. We demonstrate the benefits our adaptive approach both (1) qualitatively by visualizing time series predictions made for one of the patients, and (2) quantitatively by comparing the prediction accuracy of our two-stage adaptive forecasting model to alternative approaches. We would also like to note that the hyper parameters (e.g., DVI sampling frequency r, number of hidden states in LDS d) used in our methods are selected (in all experiments) by the internal cross validation approach while optimizing models' predictive performance. We evaluate and compare the performance of the different methods by calculating the average Mean Absolute Percentage Error (Average-MAPE) of models' predictions (eq.(3.21)). In the following experiments, we have randomly selected 100 patients out of 500 as a test set and used the remaining 400 patients for training the models.

5.1.4.1 Baselines We compare our proposed approach (rLDS+reMTGP) to the following methods. Some of these are widely used in both clinical pharmacology and machine learning communities:

1. Mean of the entire population (P_Mean).
2. Mean of each individual patient (I_Mean).
3. GP model learned from the entire population with a squared exponential covariance function (eq.(5.1.1.2)) (P_GP). [Rasmussen and Williams, 2006].
4. GP model learned from each individual time series with a squared exponential covariance function (I_GP).
5. Multi-task GP model learned from the entire MTS population with a squared exponential covariance function (eq.(2.12)) (P_MTGP). [Ghassemi et al., 2015, Durichen et al., 2015]
6. Multi-task GP model learned from each individual MTS sequence with a squared exponential covariance function (I_MTGP).
7. Regularized LDS based population model (rLDS).
8. Regularized LDS based population model combined with the Gaussian process regression model for each individual residual time series (rLDS+reGP). It is a special (simpler) version of our model.

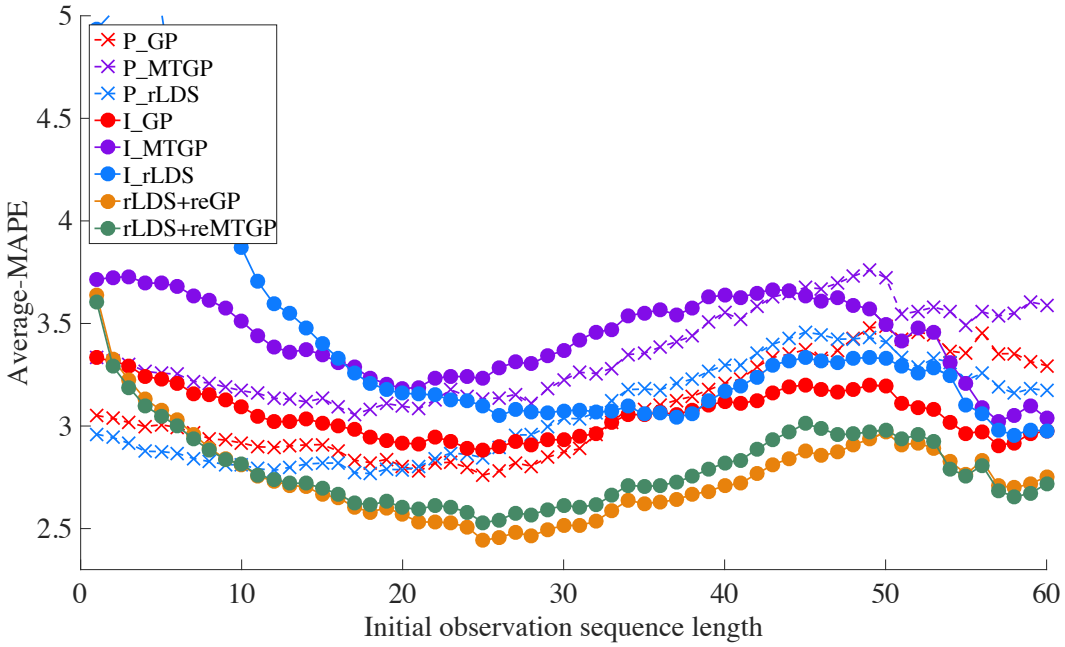


Figure 21: Average-MAPE results with different initial observation lengths.

5.1.4.2 Results Figure 21 compares our new method rLDS+reMTGP in terms of Average-MAPE to various state-of-the-art approaches listed in *Baselines* subsection. Figure 21 shows the Average-MAPE performance of all methods when they start to predict with a fixed delay corresponding to the different number of initial observations (initial observation sequence length). For example, when the initial observation sequence length is set to 4 the Average-MAPE reflects the errors of all one-step-ahead predictions the method makes when starting from four initial observations for the target patient (that is, when all predictions the model can make for sequences of $0, 1, 2, 3$ initial observations are ignored). To evaluate the statistical significance of performance difference, we apply paired t-tests at 0.05 significance level. The best methods are shown in bold in Table 15 in Appendix M. Due to the poor performance of the P_Mean and I_Mean methods, we don't visualize them in Figure 21; however, all numerical results and the corresponding standard errors are included and listed in Table 15.

The results show that the population based LDS model (P_rLDS) is the best performer

when very little is known about the target patient and when patient's observation sequences are short. In general, patients who have longer hospitalizations tend to deviate more from the population and the corresponding dynamics are not well described by the population based models. As we can see from Figure 21, the Average-MAPE of all population based models (P_GP, P_MTGP and P_rLDS) gradually increase when the initial observation length gets longer. Since our two-stage approach has to fit the parameters of the GP or MTGP models of the residual time series it may experience some initial period in which it is not stable and may lead to suboptimal predictions. However, rLDS+reGP and rLDS+reMTGP methods outperform the P_rLDS and other population based models rather quickly and become superior when more than ten initial observations for the target patient become available and are considered. Our residual based models utilize the GP based models to capture the patient specific deviations from the population rLDS model and can quickly adapt to sudden changes short-term variability appeared in each individual patient. On the other hand, pure patient specific models (I_GP, I_MTGP and I_rLDS) that ignore any population data adapt very slowly and do not reach the performance of LDS or our methods even for the initial observation sequence of length 60. Finally, a simple population based method (P_Mean) and a simple patient specific method (I_Mean) lag behind (see Appendix M for the results) and perform much worse than more advanced time series prediction models.

5.2 PERSONALIZED PREDICTION VIA ADAPTIVE MODEL SELECTION

In this section we propose and develop an adaptive clinical time series prediction framework that is different from the approach in Section 5.1 and that reflects the fact that predictions at different times may be driven by the different types of prediction models. In general, this type of problem is tackled in the machine learning literature by adaptive model selection methods. Briefly, these methods assume a pool of candidate prediction models and each of them is associated with an optimized weight that reflects how much they contribute to the prediction solution. The adaptive model selection framework we propose and develop uses

the online switching approach [Littlestone and Warmuth, 1994, Freund and Schapire, 1997] that uses a mix of population based and patient specific prediction models. The switching is driven by the weighted sum of prediction errors (or deviations) of each model on past patient's data. The weights are set so that more recent errors are more important. The method which makes fewer errors recently is more likely to be selected. We test the different error criteria on laboratory time series data and show that due to the short-term variability a model switching strategy penalizing more recent prediction errors is the best online model selection strategy in such clinical environment.

The quality of the adaptive model switching framework ultimately depends on the quality of prediction models included in the pool of time series models and their variety assuring the coverage of many different modes and behaviors. In general one can choose and put any arbitrary model into the pool. However, in this work we narrow our focus to study the trade-offs related to population based and patient specific models. This is reflected by the choices of our models. Briefly in addition to simple population and patient specific baselines we also include and consider more advanced population based LDS, patient specific LDS, as well as, population and patient specific versions of two GP models: one that relies on a set of independent univariate GP models (a time series of each clinical variable is modeled by a GP) [Rasmussen and Williams, 2006] and an MTGP where entire MTS and interactions among variables are modeled together [Bonilla et al., 2007].

5.2.1 Time Series Models

Our framework works by combining multiple different time series models and their strength to improve the prediction. Various time series models with the different assumptions may be considered [Hamilton, 1994]. In this work we power our model switching framework with two widely used time series models - LDS and GP models (Section 2.2.1 and Section 2.2.2), and develop robust population based and patient specific versions of these models and algorithms for learning them from data. The robustness assures the models can applied to cases when the number of time series examples is small or the length of the individual time series is short-span.

5.2.1.1 Population based and Patient Specific LDS

In general, the LDS model can be learned either from a collection of MTS sequences or from an individual sequence, which leads to either population based models or patient specific models.

For the LDS based population models, they are learned from all available data sequences of patients and aim to summarize the dynamics of all patients in the population. As we can see from eq.(3.1), due to the probabilistic interpretation of the LDS based models and the i.i.d assumption between each patient sequence, each patient contributes its own part to the objective function and the population model is optimized to fit the entire MTS collection to its most extent. On the other hand, we can also solely use currently available observations from the target patient and train the patient specific LDS models to capture the changes of individual patients.

The population based models in general are more robust and insensitive to outliers compared to individual specific models since they try to seek a unified model which fits all data sequences. The effects of abnormal observations will be minimized and corrected by the majority of normal observations. Furthermore, a population model is especially useful in the early stage of clinical predictions because at the beginning, observations of clinical variables for individual patient are often short and insufficient to learn a high quality model solely based on patient's own data.

However, population models usually fail to capture patient's variability due to the fact that population models are trained to have good forecasting performance on average on all the patients' sequences. Since the prediction task is performed patient by patient, an ideal forecasting model should reflect and take into account the variations specific to the current patient. Furthermore, a patient may exhibit short-term variability reflecting the different events affecting the care and patient state [Schulam et al., 2015]. Since the individual specific model is trained on each sequence, the model is better at capturing the short-term variability and providing customized predictions than population models.

We note that LDS based models belong to discrete time models which require that the time intervals between any two consecutive observations are same. When dealing with irregularly sampled time series, time series discretization techniques can be used as a data preprocessing step before learning the models. In this work, similar to Section 5.1, we

discretize the irregularly sampled MTS by using DVI approach.

5.2.1.2 Population based and Patient Specific GP and MTGP

Similarly to learning the LDS based models, GP based models can be also learned from the population sequence collection or each patient specific sequence. Learning GP based patient specific model from each sequence is straightforward. For each target patient, patient specific models are learned solely from the patient's past q observation-time pairs $(\mathbf{y}_i, t_i)_{i=1}^{q}$. More specifically, we treat each clinical time series in $(\mathbf{y}_i, t_i)_{i=1}^{q}$ independently and learn a patient specific GP model for each clinical variable. Also we take into account of the correlation and interaction between clinical variables and learn a patient specific MTGP model from $(\mathbf{y}_i, t_i)_{i=1}^{q}$. Both the GP and MTGP models has zero mean and a squared exponential covariance function (eq.(5.1.1.2)), which is the most frequently-used example in literature [Rasmussen and Williams, 2006].

Similar to Section 5.1.1.2, we adopt the Cholesky decomposition and the "free-form" parameterization techniques ($K^C = LL^\top$) to learn the parameter set Λ by minimizing the negative log marginal likelihood via gradient descent [Rasmussen and Williams, 2006, Ghassemi et al., 2015].

To learn the GP based models from a collection of MTS sequences, we learn the GP based models from each sequence in the training collection and use the average of all the learned parameters as our estimates of the population based models. While it is always possible to concatenate multiple MTS sequences into one large sequence, this brute-force concatenation process will let the covariance function learn the similarities between observations across different patients, which leads to inaccurate estimations.

Both GP and MTGP are used in clinical time series domain to capture the short-term and long-term variability [Marlin et al., 2012, Clifton et al., 2013, Lasko et al., 2013, Liu and Hauskrecht, 2015a, Schulam et al., 2015, Ghassemi et al., 2015, Durichen et al., 2015]. In [Marlin et al., 2012, Clifton et al., 2013, Lasko et al., 2013, Liu and Hauskrecht, 2015a, Schulam et al., 2015], each clinical time series is modeled by a single GP separately which does not allow one to represent dependences among the different time series. [Ghassemi et al., 2015, Durichen et al., 2015] try to capture MTS and dependences among its time series by

applying MTGP to clinical MTS modeling and forecasting. Since all above applications focus on individual-specific sequence, they tend to support more accurate and personalized time series predictions for each patient compared to population based models. However, those models usually require long enough sequences to optimize the models' parameters. This becomes unrealistic and inapplicable when a new patient comes in and very few observations are known for that patient.

5.2.2 Online Model Switching

Due to the rapid changes in the clinical time series, it is difficult to develop a single model that consistently performs well over the time for each individual. Therefore, in this work, we make the prediction for patient p at time t^* from a pool of candidate models, which contains both the population model (LDS based and GP based) and patient specific models (LDS based and GP based). Our objective is to develop a framework that is able to pick the best model from the pool to timely support accurate and personalized clinical predictions for each patient at every time stamp.

Although numerous ensemble and online methods exist, the majority of the methods require error feedback over longer periods of time to achieve any statistical guarantee of total errors made by the algorithms. However, in the real-world clinical setting, patients' time series are usually too short to obtain effective weights for both the ensemble and online algorithms. Furthermore, weight updating rules are often based on the overall performance of each model on all previously observed data and hence the recent errors are smoothed out by the errors made in the early stage of the process. Since clinical MTS may contain short-term variability (caused, for example, by acute infections, bleeding, surgeries, etc) standard weight updating rules are not able to respond to these changes quickly enough.

In this work, we propose and develop a novel online model switching strategy, i.e., "weighted Follow-the-Leader" (wFTL), to address the above problem. Different from traditional online learning algorithms that treat each past errors equally, we put more penalties on recent errors. The intuition is in that the predictive models that do not perform well initially can catch up in their performance rapidly and they may need be selected. More

precisely, for each model $\mathcal{M}_m$, all its past errors can be computed (up to current time stamp t_q) as $\mathbf{e}_m = [e_1^m, \cdots e_i^m, \cdots, e_q^m]$. The model being picked at time t^* is

$$\mathcal{M}_* = \arg\min_m \sum_{i=1}^{q} w_i * e_i^m \tag{5.1}$$

where w_i is the error weight at time t_i.

In order to capture the recency effect, we compute the error weight by using the kernel functions that take time stamps as inputs. The idea is that the errors made far away should be less penalized compared to the most recent errors. We experiment with two standard kernel functions: the square exponential kernel (eq.(5.2)) and the mean reverting kernel (eq.(5.3)) to penalize the errors with respect to time elapsed.

$$K_{se}(t_i, t^*) = \exp\Big(-\frac{(t_i - t^*)^2}{\gamma}\Big) \tag{5.2}$$

$$K_{mr}(t_i, t^*) = \exp\Big(-\frac{|t_i - t^*|}{\gamma}\Big) \tag{5.3}$$

where t^* is the time stamp of the target prediction. t_i is the all the past time stamps, $i = 1, 2, \cdots q$ and γ is the bandwidth parameter.

As we can see from eq.(5.1), the proposed approach downgrades to the "Follow-the-Leader" (FTL) strategy when all the weights (w_is) become 1. [Shalev-Shwartz, 2011]. The FTL strategy simply selects the best prediction model by integrating the loss across past t steps and neglects the recency effect. While wFTL always selects the prediction model with the minimum weighed loss over time. As a result, it is more sensitive to the recent observations that reflect the most current trend and change of the state of the target patient. By evaluating the candidate models' predictions and focusing on the recent performance, the proposed strategy is able to discover sudden changes and quickly switch to the best model. Compared with eq.(5.2) and eq.(5.3), the square exponential kernel squares the time difference which vanishes the past errors much quicker than mean reverting kernel. The hyper parameter γ controls the magnitude of the recency effect. wFTL with either eq.(5.2) or eq.(5.3) becomes FTL when γ goes to infinity.

5.2.3 Experiment

In this section, we evaluate our approach on the same clinical MTS data set we used in Section 5.1. We conduct a series of experiments to explore and demonstrate the benefits of our adaptive model switching framework. First, we study the quality of population based versus patient specific models for observations histories of the different length. Second, we focus on the MTS forecasting and the evaluation of the proposed model switching approach to other models. We evaluate and compare the performance of the different methods by calculating the average Mean Absolute Percentage Error (Average-MAPE) of models' predictions (eq.(3.21)).

During our evaluations we consider a variety of time series prediction models used commonly in both clinical pharmacology and machine learning and their population based and patient specific versions. All these can be put into the pool of candidate models our framework uses. For the population based models, we choose (1) P_Mean: mean of the entire population; (2) P_rLDS: a regularized LDS learned from other patient data $\mathcal{D}$; (3) P_GP: a population GP model learned from $\mathcal{D}$; and (4) P_MTGP: a population MTGP model learned from $\mathcal{D}$. For patient specific models, we choose (1) I_Mean: Mean value for the individual patient up to the current time stamp; (2) I_rLDS: learning a rLDS model from the MTS sequence of the target patient; (3) I_GP: Gaussian process regression model for each individual time series of the target patient; and (4) I_MTGP: multi-task Gaussian process model for the MTS sequence of the target patient.

5.2.3.1 Baselines In the following experiments, we denote the wFTL with the square exponential kernel (eq.(5.2)) as wFTL_se and denote the wFTL with the mean reverting kernel (eq.(5.3)) as wFTL_mr. We compare our wFTL_se and wFTL_mr model switching strategies to other approaches one can use for personalized predictive modeling.

- *Sub*: represents a subpopulation approach. For each patient at each time stamp, top k similar patients are selected and are used to train the rLDS model. The similarity is defined by the Euclidean distance between the sample means of clinical variables of the target patients and the sample means of available training patients. In this experiments,

we vary k to 50, 100, 200 and ALL where ALL means all the training examples.

- *rLDS+reGP*: is a model adaptation approach. In rLDS+reGP, a population rLDS model is trained first and the time series of past observations for the target patient is expressed in terms of residuals (or differences in between predictions made by the population based model and actually observed values). Then each of the residual time series is modeled by a GP. Details are described in Section 5.1.
- *rLDS+reMTGP*: is another model adaptation method that is similar to rLDS+reGP but the all residual time series are modeled by an MTGP. Details are described in Section 5.1.
- *En_Avg*: is a simple averaging method in which the prediction is made by uniformly averaging the results from all the models in the pool.
- *En_Err*: is the inverse-error weighted average method. Assuming M be the number of models in the pool. Let e_m be the sum of prediction errors of the model m over the past t time steps (rounds) and w_m be the mixture weight corresponding to model m. In En_Err, w_m is computed as $w_m = \frac{1}{e_m S}$ where $S = \sum_{m=1}^{M} \frac{1}{e_m}$.
- *OL_FTL*: Follow-the-Leader method that selects the best model based on the loss integrated over past t time stamps.
- *OL_MW*: Multiplicative weights algorithm [Cesa-Bianchi et al., 2007] that at each round t, makes the selection is based on the probability distribution $\mathbf{p} = \{w_1/\Phi, \cdots, w_M/\Phi\}$, where $\Phi = \sum_{m=1}^{M} w_m$. w_m is updated by penalizing the costly predictions, i.e., $w_m^+ = w_m(1 - \eta e_m)$ where $\eta, \eta \leq 0.5$ is the discounting factor.
- *OL_Hedge*: Hedge algorithm [Freund and Schapire, 1997] that is similar to OL_MW but uses an exponential factor instead of a linear cost $(1 - \eta e_m)$. The weight update is $w_m^+ = w_m \exp(-\eta e_m)$.

5.2.3.2 Results

Population based versus Patient specific Models We first explore the prediction performance of each model in the prediction model pool individually. Instead of averaging all the prediction results, we compute the Average-MAPE results of population based, patient specific methods and our proposed wFTL model switching approaches (wFTL_se and

wFTL_mr) when they start to predict with a delay corresponding to the different number of initial observations (initial observation sequence length). For example, when the initial observation sequence length is set to 4 the Average-MAPE reflects the errors of all one-step-ahead predictions the method makes when starting from four or more initial observations for the target patient (that is, the model starts to make predictions from the 5th time stamp). The Average-MAPE results with different initial observation lengths are shown in Figure 22. To evaluate the statistical significance of performance difference, we apply paired t-tests at 0.05 significance level. The best methods are shown in bold in Table 16 in Appendix N. Due to the poor performance of the P_Mean and I_Mean methods, we don't visualize them in Figure 22; however, all numerical results and the corresponding standard errors are included and listed in Table 16.

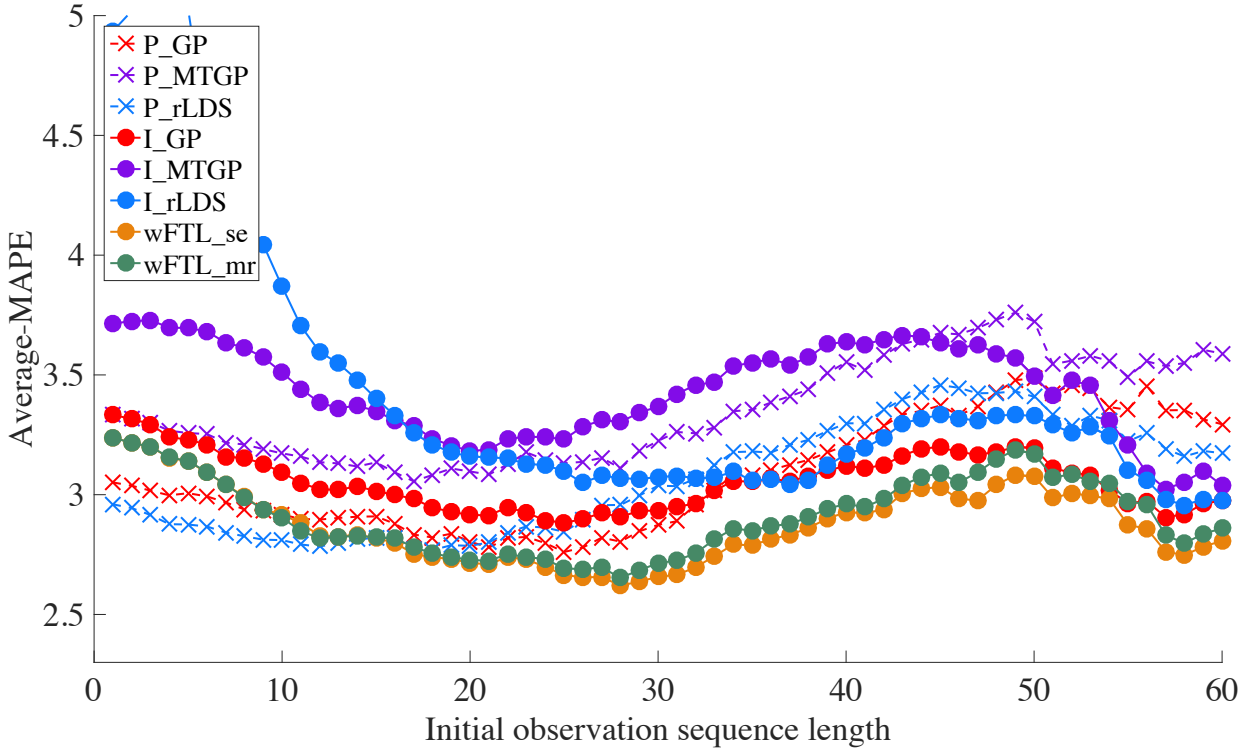


Figure 22: Average-MAPE results of all models in the pool and two wFTL methods for the different initial observation lengths.

First, Figure 22 shows the trade-off between the population based (P_rLDS, P_GP and

P_MTGP) and the patient specific (I_rLDS, I_GP and I_MTGP) models. Briefly, the performance of patient specific models built from patient's own past observations tends to gradually improve and eventually outperforms the population based models that are the best initially when little is known about the target patient. More specifically we observe that, P_rLDS model built on the population of past patients starts strong but deteriorates when more values are observed. We explain this deterioration by the fact that longer the patients stay in the hospital the more likely they deviate from the population based models. This is also reflected by the deterioration of the population based GP models (P_GP and P_MTGP) for longer observation sequences. On the other hand, we observe that patient specific models can adapt to the specifics of the patient but they also take a longer time (number of observations) to learn, especially when the model is more complex. While I_GP is relatively fast to adapt to the specifics and short-variability of the target patient, I_MTGP is slower because of increased model complexity and more parameters it needs to learn. In addition, from Figure 22, we can see that different models have various prediction performance when the number of observations change, which confirms the motivation of dynamically switching to the most appropriate model during the prediction. By using the different kernel functions (eq.(5.2) and eq.(5.3)), our wFTL strategies penalize the most recent errors made by each candidate model. As shown in Figure 22, the proposed wFTL approaches are slightly worse compared to P_rLDS initially. But they catch up the performance of P_rLDS rapidly (when initial observation length reaches 10 shown in Table 16) and consistently have the best performance among all the population based and patient specific models when enough initial observations are obtained.

Prediction Accuracy In this experiment, we compute and compare one-step-ahead prediction accuracy of wFTL to various state-of-the-art personalization approaches. We present the prediction results against baselines in different categories separately to make the differences clear. The results are shown in Figures 23 - 25. To evaluate the statistical significance of performance difference, we apply paired t-tests at 0.05 significance level. All numerical results, the corresponding standard errors and significant test results are listed in Tables 17 - 19 in Appendix O.

As we can see from Figure 23, when initial observation sequence length is short (less

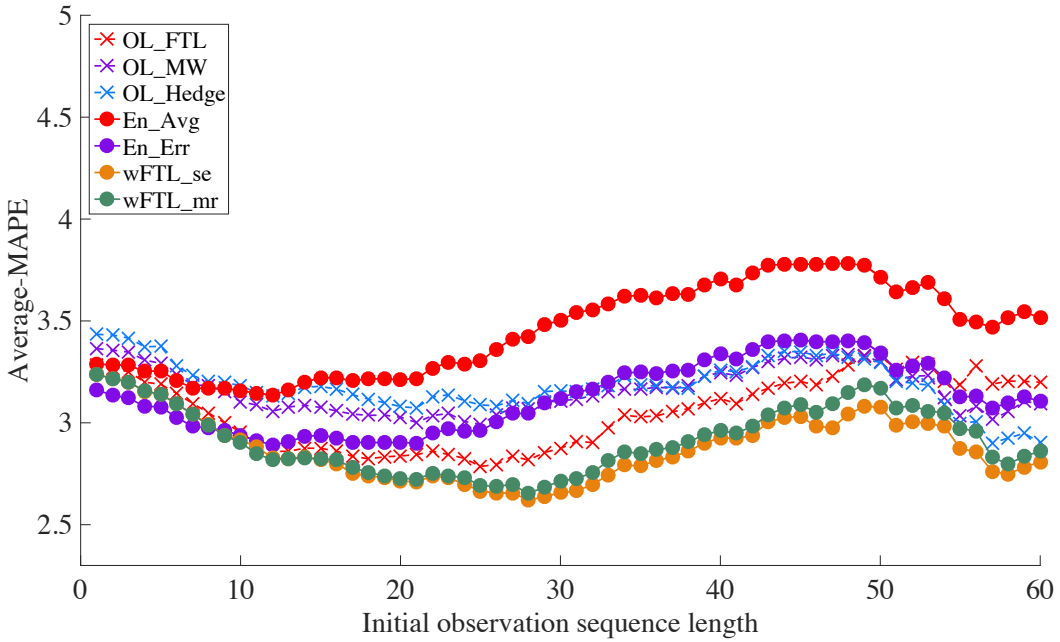


Figure 23: Average-MAPE results of the proposed wFTL approaches compared to the ensemble and online methods.

than 9), our wFTL strategies perform slightly worse than the inverse-error weighted average method (En_Err). But in the long run, our wFTL strategies have the best performance among all the other adaptive model selection based baselines. Clinical time series contain lots of short-term variability due to different causes [Schulam et al., 2015]. For example, the blood tests may be affected by events like infection, bleeding, transfusion, or a particular medication treatment. Patient specific models can adapt better to this variability while population based models tend to average the variability out (treat them as a noise) so they likely do not perform well when these "exceptions" occur. Since wFTL strategies not only consider the past errors but also focus on the most recent performance of each predictor, they are able to quickly adapt to the short-term variability and rapid changes. On the contrary, the standard adaptive model selection approaches (ensemble methods and online algorithms) are all based on weighting schemas extracted from the entire history. These historical observations are too long and may prevent us from adapting to these short-term

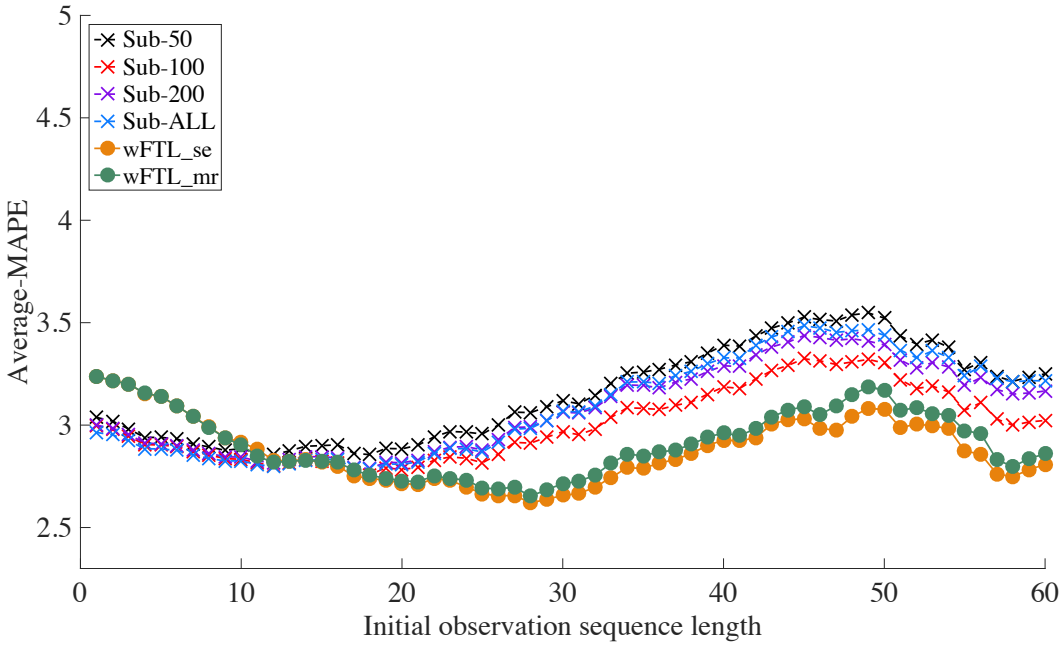


Figure 24: Average-MAPE results of the proposed wFTL approaches compared to the subpopulation methods.

variability (shown in Figure 23). Furthermore, in order to change the prediction behavior of these methods, the weights must be changed. Since there are many different weights it may take a long time for them to be adapted. This is also reflected by the improvement of the online learning approaches (OL_MW and OL_Hedge) for longer observation sequences. As we can see from the statistical significance test results in Table 17, both OL_MW and OL_Hedge have the comparable performance to our wFTL strategies only when the initial observation length is large than 49.

Figure 24 compares the prediction performance of our wFTL strategies and subpopulation methods. Similarly to the prediction results in Figure 23, the subpopulation methods achieve better performance when initial observation sequence length is less than 12. This is because patients start to differentiate and exhibit their unique symptoms as their hospitalizations go by. For subpopulation methods, it is difficult to accurately find and represent the target patient's short-term changes by solely using the static examples from the training set.

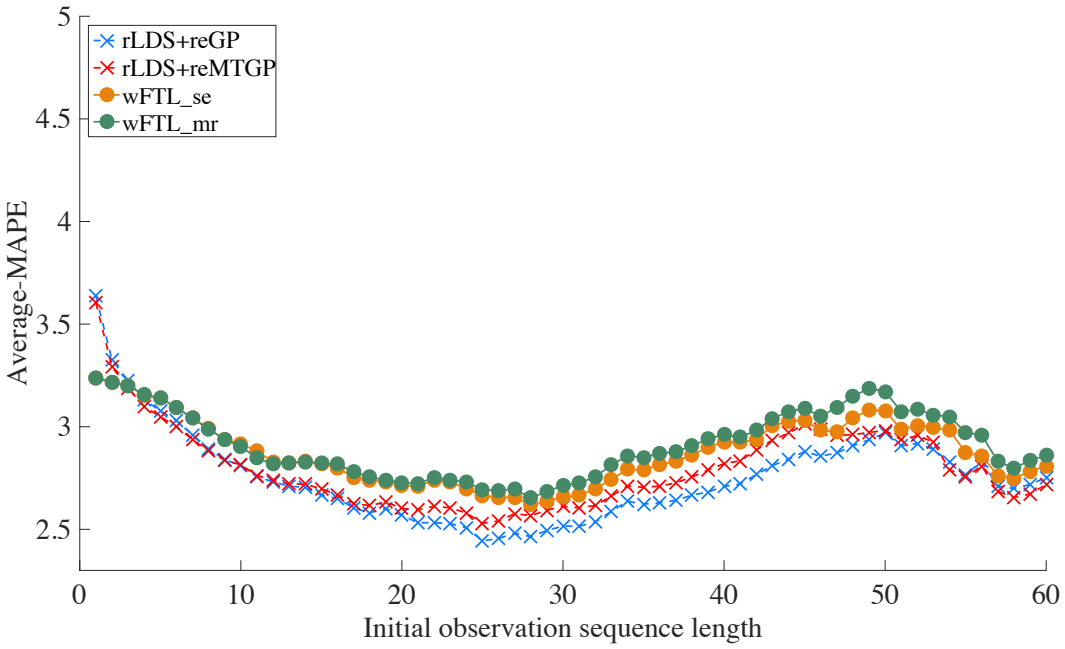


Figure 25: Average-MAPE results of the proposed wFTL approaches compared to the model adaptation based methods.

The top k similar subset might not be able to reflect the most recent temporal behavior of the target patient. Moreover, from Figure 24 and the statistical significance test results in Table 18, we can see that the performance of subpopulation methods vary with different values of k. Choosing the optimal value of k is an challenging issue. In subpopulation methods, the top k subset is specific to each patient and it is re-constructed when every new observation is obtained for that specific patient, repeatedly searching for the best subset. As a result, the training of the subpopulation model becomes very time consuming. It is not practical to apply such methods in a large scale EHR data set.

We also compare our online model switching strategies (wFTL_mr and wFTL_se) with the residual based model adaptation techniques (rLDS+reGP and rLDS+reMTGP) and the results are shown in Figure 25. As we can see, our wFTL_mr and wFTL_se switching strategies have comparable performance to model adaptation techniques although they are slightly worse numerically than rLDS+reGP and rLDS+reMTGP. We run the pairwise t-test

for each possible pair from these two categories of methods and none of them are statistically significantly different at 0.05 level (shown in Table 19). Please note that even though the two approaches have similar performance, they are different by nature: the wFTL strategies keep selecting the best predictor from a pool of candidate models based on the weighted average of past errors while the residual based model adaptation techniques rely on learning from patient specific residuals to capture the short-term variability in patient dynamics. Both rLDS+reGP and rLDS+reMTGP models have worse performance at the beginning is because they require enough residuals to fit the parameters of the GP or MTGP models.

We also note that models from the subpopulation methods and model adaptation approaches are complementary and they can be combined in the prediction process. For example, the model adaptation techniques can be applied to both population based models and subpopulation models. Moreover, both subpopulation models and adaptive models can be candidate models in the pool, which can be used by our online model switching strategies.

5.3 SUMMARY

In this chapter, we focused on the task of building an accurate predictive model of irregularly sampled clinical MTS for a patient, which is critical for understanding of the patient condition, its dynamics, and optimal patient management. We proposed and developed two forecasting frameworks to address the following two problems simultaneously: (1) patient specific variations are typically large and population models derived or learned from many different patients are often unable to support accurate predictions for each individual patient; and (2) time series observed for one patient at any point in time may be too short and insufficient to learn a high-quality patient specific model just from the patient's own data.

First, we built a personalized predictive model via model adaptation. We proposed and developed an adaptive two-stage forecasting approach which (1) learns the population trend from a collection of time series for past patients; (2) captures individual-specific short-term multivariate variability; and (3) adapts by automatically adjusting its predictions based on new observations. In contrast to the traditional time series forecasting models, our

model learns from both the population data (time series for other patients) and the target patient data (time series of past observations for the target patient). Our experimental results demonstrated that our model can outperform after a short adaptation period other prediction models and approaches.

Second, we tackled the above challenges by using adaptive model selection. We proposed, developed and experimented with a new adaptive forecasting framework for building multivariate clinical time series models for a patient and for supporting patient specific predictions. The framework relies on the adaptive model switching approach that at any point in time selects the most promising time series model out of the pool of many possible models, and consequently, combines advantages of the population, patient specific and short-term individualized predictive models. We demonstrated that the adaptive model switching framework is very promising approach to support personalized time series prediction, and that it is able to outperform predictions based on pure population and patient specific models, as well as, other patient specific model adaptation strategies.

6.0 CONCLUSION

The focus of this dissertation was on the development of multivariate clinical time series models that are able to support accurate time series predictions. We identified three important characteristics of clinical MTS not fully researched in the existing time series literature and developed new solutions and models to fill these gaps. The main contributions of this dissertation are summarized in Section 6.1 and some related open questions and research opportunities are outlined in Section 6.2.

6.1 CONTRIBUTIONS

- We presented a probabilistic framework for learning regularized LDS models from a limited number of regularly sampled MTS sequences. The framework builds upon the probabilistic formulation of the LDS model, and casts the optimization of its parameters as an MAP problem, where the choice of parameter priors biases the model towards a low-rank solution. We showed the regularized LDS models are able to recover the intrinsic dimensionality of the MTS data and consequently prevent overfitting problems of ordinary LDS models. We also showed that regularized LDS models can greatly improve the forecasting performance on a real-world clinical MTS data.
- We developed a new generalized LDS framework, gLDS, for learning constrained LDS models from a collection of MTS data. The framework treats both LDS parameters and the hidden states as unknown variables and applies alternating minimization to learn them from data. We evaluated our gLDS framework experimentally on four real-world MTS data sets. We showed that (1) ordinary LDS models learned from gLDS are able

to achieve better time series predictive performance than other LDS learning algorithms; (2) constraints can be flexibly integrated into the learning process to achieve desired properties of the dynamical models such as stability, low-rankness; and (3) the proposed temporal smoothing regularization encourages more stable and accurate predictions.

- We built a new hierarchical dynamical system to address the forecasting problem for irregularly sampled univariate time series. Our model is built by combining the advantages of the LDS and the GP models. Experimentally, we demonstrated that our model outperforms multiple existing models in terms of its predictive accuracy. Our method achieved a 3.13% average prediction accuracy improvement on ten CBC laboratory time series when it was compared against the best performing baseline. A 5.25% average accuracy improvement was observed when only short-term predictions were considered.
- We presented a two-stage adaptive forecasting model to provide patient specific predictions, which is learned from irregularly sampled multivariate clinical data. First, we learned a population based LDS model from many different patients. Then, we used a multi-task Gaussian process to model the patient specific residuals, which are the differences in between predictions made by the population based model and actually patient specific observations. We demonstrated the benefits of our approach on the prediction tasks for irregularly sampled multivariate clinical data, and showed that it can outperform both the population based and patient specific predictive models in terms of prediction accuracy.
- We studied a framework to build the personalized predictive models via adaptive model selection. Our framework addressed the problems of building accurate forecasting models from irregularly sampled multivariate clinical data and at the same time providing the patient specific predictions. Our framework selects the best model for each patient at every time stamp based on the weighted sum of prediction errors (or deviations) of each model on past patient specific data. The weights are set so that more recent errors are more important. The method which makes fewer errors recently is more likely to be selected. We tested the different error criteria on laboratory time series data and showed that due to the short-term variability a model switching strategy penalizing more recent prediction errors is the best online model selection strategy in such clinical environment.

6.2 OPEN QUESTIONS

We have proposed frameworks to learn better forecasting models for irregularly sampled multivariate clinical data while supporting patient specific predictions. Our approaches show superior performance compared to existing approaches. However, there are still many challenges and open questions that prompt to further investigations.

- **Incorporating effect of future events and actions into models**. Clinical time series contain lots of short-term variability. Many of them are caused by clinical events such as transfusion, taking a medication treatment, etc. These clinical events have direct influence on the values of clinical MTS and hence, cause the short-term variability. Explicitly incorporating or modeling the effects of external clinical events and actions on future laboratory values may potentially boost the performance of the existing work. An important research problem is to extend our method to event-specific models, which are specialized for patient specific short-term variability. More specifically, a series of clinical events can be viewed as a series of time dependent exogenous inputs and events transit from one to the others based on the patients' current status and the observed values of laboratory tests. Basically, we can either use a different transition matrix for each clinical action/event or supply a value of an external factor into the transition matrix of the hidden states (A). Also, partially observable Markov decision process (POMDP) could potentially be used to model the real-world sequential clinical decision making processes.
- **Regularized MTGP learning**. As observed before (in Figure 22), instance-specific MTGP models exhibit good performance when enough observations are obtained and perform poorly initially. This is due to the fact that the MTGP models are more complex and have more parameters they need to learn. More specifically, the number of parameters in K^C in eq.(2.12) (a matrix measuring the similarities between time series) are quadratic in the number of time series. Given the very limited number of available data at the beginning, the MTGP models may run into the overfitting problem which yield to low accuracy. Therefore, it is worthwhile to investigate how we can learn the MTGP models from a small amount of data and prevent the overfitting problems. A

further direction would be reducing the complexity of K^C by enforcing or constraining the structure of it and consequently, guarantee the stable learning process and avoid the overfitting problems.

- **Similarity of length-varying irregularly sampled MTS**. Personalized predictive methods discussed in Section 2.3 from different categories can be combined to support more accurate predictions. Therefore, we can apply our model selection based adaptive prediction framework (Section 5.2) to a pool of candidate models that include models learned from subpopulation. This opens a new research question of identifying the best possible time series subpopulation. Besides identifying similar MTS sequences by first picking an appropriate probabilistic model and then using the likelihood as an abstraction to represent the sequence, like the work by Huang et al. [Huang et al., 2014], can we direct compute the similarity between two length-varying irregularly sampled MTS? One potential solution is to discretize the irregularly sampled sequence first and then use dynamic time warping to compute the similarities between each sequence.
- **Exploration of non-linear dynamical models**. Non-linearity increases the time series models' expressive ability. Therefore, we can explore the opportunities of incorporating advanced variants of both LDS and GP models into the currently developed framework. Possible directions like replacing LDS models with unscented Kalman filter [Wan and Van Der Merwe, 2000], or using GP models as the non-linear transformation operators, such as in the state-space model with transition. The complexity of non-linear models may give rise to overfitting and computation issues. Another possible direction would be starting with linear models when the observations are few and switching to non-linear models when sufficient number of observations is collected.

APPENDIX A

KALMAN FILTER ALGORITHM FOR LDS

For the sake of notational brevity, we omit the explicit sample index ("l"). Let denote $\hat{\mathbf{z}}_{i|T} \equiv \mathbb{E}[\mathbf{z}_i|\mathbf{Y}]$, $M_{i|T} \equiv \mathbb{E}[\mathbf{z}_i\mathbf{z}_i^\top|\mathbf{Y}]$, $M_{i,i-1|T} \equiv \mathbb{E}[\mathbf{z}_i\mathbf{z}_{i-1}^\top|\mathbf{Y}]$, $P_{i|T} = \mathbb{VAR}[\mathbf{z}_i|\mathbf{Y}]$, and $P_{i,i-1|T} = \mathbb{VAR}[\mathbf{z}_i\mathbf{z}_{i-1}^\top|\mathbf{Y}]$. Let $\hat{\mathbf{z}}_{i|i-1}$ be the *priori* estimation of $\mathbb{E}[\mathbf{z}_i|\mathbf{Y}_{1:i-1}]$, $\hat{\mathbf{z}}_{i-1|i-1}$ be the *posteriori* estimation of $\mathbb{E}[\mathbf{z}_{i-1}|\mathbf{Y}_{1:i-1}]$, $P_{i|i-1}$ be the *priori* estimate error covariance of $\mathbb{E}[(\mathbf{z}_i - \hat{\mathbf{z}}_{i|i-1})(\mathbf{z}_i - \hat{\mathbf{z}}_{i|i-1})^\top]$ and $P_{i-1|i-1}$ be the *posteriori* estimate error covariance of $\mathbb{E}[(\mathbf{z}_{i-1} - \hat{\mathbf{z}}_{i-1|i-1})(\mathbf{z}_{i-1} - \hat{\mathbf{z}}_{i-1|i-1})^\top]$.

Algorithm 6 Kalman filter algorithm for LDS

INPUT:

- MTS data $\mathbf{Y}$.
- Current step LDS parameters: $\Omega = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$.

PROCEDURE:

// Initialize the recrusion
$\hat{\mathbf{z}}_{1|1} = \boldsymbol{\xi}$ and $P_{1|1} = \Psi$.
// Start the recrusion
for i = 2 → T **do**
  // Time Update:
  $\hat{\mathbf{z}}_{i|i-1} = A\hat{\mathbf{z}}_{i-1|i-1}$
  $P_{i|i-1} = AP_{i-1|i-1}A^\top + Q$
  // Measure Update:
  $K_i = P_{i|i-1}C^\top(CP_{i|i-1}C^\top + R)^{-1}$
  $\hat{\mathbf{z}}_{i|i} = \hat{\mathbf{z}}_{i|i-1} + K_i(\mathbf{y}_i - C\hat{\mathbf{z}}_{i|i-1})$
  $P_{i|i} = P_{i|i-1} - K_iCP_{i|i-1}$
end for

OUTPUT: $\{\hat{\mathbf{z}}_{i|i-1}\}_{i=2}^T$, $\{\hat{\mathbf{z}}_{i|i}\}_{i=1}^T$, $\{P_{i|i}\}_{i=1}^T$, $\{P_{i|i-1}\}_{i=2}^T$ and $\{K_i\}_{i=1}^T$.

APPENDIX B

E-STEP BACKWARD ALGORITHM FOR LDS

Algorithm 7 EM: E-step backward algorithm for LDS

INPUT:

- Output from Kalman filter algorithm: $\{\hat{\mathbf{z}}_{i|i-1}\}_{i=2}^{T}$, $\{\hat{\mathbf{z}}_{i|i}\}_{i=1}^{T}$, $\{P_{i|i}\}_{i=1}^{T}$, $\{P_{i|i-1}\}_{i=2}^{T}$ and $\{K_i\}_{i=1}^{T}$. Kalman filter algorithm is presented in Algorithm 6 in Appendix A.
- Current step LDS parameters: $\Omega = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$.

PROCEDURE:

// Initialize the recrusion
$M_{T|T} = P_{T|T} + \hat{\mathbf{z}}_{T|T}\hat{\mathbf{z}}_{T|T}^{\top}$
$J_{T-1} = P_{T-1|T-1}A^{\top}(P_{T|T-1})^{-1}$
$P_{T-1|T} = P_{T-1|T-1} + J_{T-1}(P_{T|T} - P_{T|T-1})J_{T-1}^{\top}$
$\hat{\mathbf{z}}_{T-1|T} = \hat{\mathbf{z}}_{T-1|T-1} + J_{T-1}(\hat{\mathbf{z}}_{T|T} - A\hat{\mathbf{z}}_{T-1|T-1})$
$P_{T,T-1|T} = (I - K_T C)AP_{T-1|T-1}$
$M_{T,T-1|T} = P_{T,T-1|T} + \hat{\mathbf{z}}_{T|T}\hat{\mathbf{z}}_{T-1|T}^{\top}$
// Start the recrusion
for i = T-1 $\rightarrow$ 1 **do**
    $M_{i|T} = P_{i|T} + \hat{\mathbf{z}}_{i|T}\hat{\mathbf{z}}_{i|T}^{\top}$
    $J_{i-1} = P_{i-1|i-1}A^{\top}(P_{i|i-1})^{-1}$
    $P_{i,i-1|T} = P_{i|i}J_{i-1}^{\top} + J_i(P_{i+1,t|T} - AP_{i|i})J_{i-1}^{\top}$
    $M_{i,i-1|T} = P_{i,i-1|T} + \hat{\mathbf{z}}_{i|T}\hat{\mathbf{z}}_{i-1|T}^{\top}$
    $\hat{\mathbf{z}}_{i-1|T} = \hat{\mathbf{z}}_{i-1|i-1} + J_{i-1}(\hat{\mathbf{z}}_{i|T} - A\hat{\mathbf{z}}_{i-1|i-1})$
    $P_{i-1|T} = P_{i-1|i-1} + J_{i-1}(P_{i|T} - P_{i|i-1})J_{i-1}^{\top}$
end for

OUTPUT: $\{\hat{\mathbf{z}}_{i-1|T}\}_{i=1}^{T}$, $\{M_{i|T}\}_{i=1}^{T}$ and $\{M_{i,i-1|T}\}_{i=1}^{T}$.

APPENDIX C

PROOF OF THEOREM 1

Proof. Let denote $\Delta \equiv \sum_{l=1}^{N}\sum_{i=1}^{T_l-1}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_i^l)^\top|\mathbf{Y}^l]$. Given the gradient of h(A) (eq.(3.5)), we have

$$\begin{aligned}
&\| \bigtriangledown h(X) - \bigtriangledown h(Y)\|_F \\
=&\|Q^{-1}(X-Y)\Delta + \lambda_G(X-Y)\|_F \\
\leq&\|Q^{-1}\|_F \cdot \|\Delta\|_F \cdot \|X-Y\|_F + \lambda_G \cdot \|X-Y\|_F \\
=&(\|Q^{-1}\|_F \cdot \|\Delta\|_F + \lambda_G) \cdot \|X-Y\|_F
\end{aligned}$$

Since we have $\| \bigtriangledown h(X) - \bigtriangledown h(Y)\|_F \leq \mathcal{L} \cdot \|X-Y\|_F$, where $\mathcal{L} = \|Q^{-1}\|_F \cdot \|\Delta\|_F + \lambda_G$. $\bigtriangledown h(A)$ has Lipschitz continuous with constant $\mathcal{L}$. According to [Shor, 1968, Fornasier and Rauhut, 2008], we have

$$\left\|h(A^{(k)}) + \lambda_N\|A^{(k)}\|_* - h(A^{(*)}) - \lambda_N\|A^{(*)}\|_*\right\| \leq \left\|A^{(0)} - A^*\right\|_F^2/2tk$$

where $A^{(0)}$ is the initial value and A^* is the optimal value for A; k is the number of iterations.

□

APPENDIX D

PROOF OF THEOREM 2

Proof. From eq.(3.8), we can have the following equation transformation,

$$
\begin{aligned}
&arg\min_A g(A) + \frac{\lambda_G}{2}\|A\|_F^2 \Leftrightarrow arg\min_A \frac{1}{2}\sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}\left[(\mathbf{z}_i^l - A\mathbf{z}_{i-1}^l)^\top Q^{-1}(\mathbf{z}_i^l - A\mathbf{z}_{i-1}^l)\right] + \frac{\lambda_G}{2}\|A\|_F^2 \\
\Leftrightarrow &arg\min_A \frac{1}{2}\sum_{l=1}^{N}\sum_{i=2}^{T_l} Tr\left[A\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_{i-1}^l(\mathbf{z}_{i-1}^l)^\top]A^\top Q^{-1} - 2\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top]A^\top Q^{-1} + \lambda_G A^\top A\right] \\
\Leftrightarrow &arg\min_A \frac{1}{2} Tr\left[L^\top A\sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_{i-1}^l(\mathbf{z}_{i-1}^l)^\top]A^\top L - 2L^\top\sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top]A^\top L + \lambda_G A^\top A\right] \\
\Leftrightarrow &arg\min_A 0.5\,\mathrm{vec}(A^\top L)^\top(I_d \otimes \sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_{i-1}^l(\mathbf{z}_{i-1}^l)^\top])\,\mathrm{vec}(A^\top L) \\
&- \mathrm{vec}(L)^\top(I_d \otimes \sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top])\,\mathrm{vec}(A^\top L) + 0.5\lambda_G\,\mathrm{vec}(A^\top)^\top\,\mathrm{vec}(A^\top) \\
\Leftrightarrow &arg\min_A 0.5\,\mathrm{vec}(A^\top)^\top(Q^{-1} \otimes \sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_{i-1}^l(\mathbf{z}_{i-1}^l)^\top] + \lambda_G I_{d^2})\,\mathrm{vec}(A^\top) \\
&- \mathrm{vec}(L)^\top(L^\top \otimes \sum_{l=1}^{N}\sum_{i=2}^{T_l}\mathbb{E}_{\mathbf{Z}^l}[\mathbf{z}_i^l(\mathbf{z}_{i-1}^l)^\top])\,\mathrm{vec}(A^\top) \\
\Leftrightarrow &arg\min_a \frac{1}{2}a^\top H a - b^\top a
\end{aligned}
$$

where $Q^{-1} = LL^\top$, H and b are defined in eq.(3.10) and eq.(3.11).

□

APPENDIX E

PROOF OF THEOREM 4

Proof. $g(A)$ is differentiable with respect to A, and its gradient is

$$\bigtriangledown g(A) = 2(A\mathbf{Z}_{-}\mathbf{Z}_{-}^{\top} - \mathbf{Z}_{+}\mathbf{Z}_{-}^{\top} + \gamma/\lambda A)$$

Using simple algebraic manipulation we arrive at

$$\begin{aligned}
&|| \bigtriangledown g(X) - \bigtriangledown g(Y)||_F \\
=&2||(X-Y)(\mathbf{Z}_{-}\mathbf{Z}_{-}^{\top}) + \gamma/\lambda(X-Y)||_F \\
\leq&2\|\mathbf{Z}_{-}\mathbf{Z}_{-}^{\top}\|_F \cdot ||X-Y||_F + 2\gamma/\lambda \cdot ||X-Y||_F \\
=&2(\|\mathbf{Z}_{-}\mathbf{Z}_{-}^{\top}\|_F + \gamma/\lambda) \cdot ||X-Y||_F
\end{aligned}$$

The inequality holds because of the sub-multiplicative property of Frobenius norm. Since we know for eq.(3.35), $\min_A g(A) + \gamma_A \|A\|_*$, and $g(A)$ has Lipschitz continuous gradient with constant $2(\|\mathbf{Z}_{-}\mathbf{Z}_{-}^{\top}\|_F + \gamma/\lambda)$, according to [Fornasier and Rauhut, 2008, Shor, 1968] we have

$$\begin{aligned}
&\left\| g(A^{(k)}) + \gamma_A \|A^{(k)}\|_* - g(A^{(*)}) - \gamma_A \|A^{(*)}\|_* \right\| \\
\leq&\left\| A^{(0)} - A^* \right\|_F^2 / 2tk
\end{aligned}$$

where $A^{(0)}$ is the initial value and A^* is the optimal value for A; k is the number of iterations.

□

APPENDIX F

PROOF OF THEOREM 5

Proof. We will use the following equation to show the equivalence.

$$tr(A_{k\times l}B_{l\times m}C_{m\times n}) = \mathrm{vec}(A^\top)^\top (I_k \otimes B)\,\mathrm{vec}(C)$$

$$\begin{aligned}
&\min_A \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 \\
\Leftrightarrow &\min_A Tr[(\mathbf{Z}_+^\top - \mathbf{Z}_-^\top A^\top)(\mathbf{Z}_+ - A\mathbf{Z}_-)] \\
\Leftrightarrow &\min_A Tr[A\mathbf{Z}_-\mathbf{Z}_-^\top A^\top - 2I_d\mathbf{Z}_+\mathbf{Z}_-^\top A^\top] \\
\Leftrightarrow &\min_A \mathrm{vec}(A^\top)^\top (I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top)\,\mathrm{vec}(A^\top) - 2\,\mathrm{vec}(I_d)^\top (I_d \otimes \mathbf{Z}_+\mathbf{Z}_-^\top)\,\mathrm{vec}(A^\top) \\
\Leftrightarrow &\min_a a^\top (I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top)a - 2\,\mathrm{vec}(I_d)^\top (I_d \otimes \mathbf{Z}_+\mathbf{Z}_-^\top)a \\
\Leftrightarrow &\min_a a^\top (I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top)a - 2\Big((I_d \otimes \mathbf{Z}_-\mathbf{Z}_+^\top)\,\mathrm{vec}(I_d)\Big)^\top a \\
\Leftrightarrow &\min_a a^\top Ba - 2q^\top a
\end{aligned}$$

where $a = \mathrm{vec}(A^\top)$, $B = I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top$ and $q = (I_d \otimes \mathbf{Z}_-\mathbf{Z}_+^\top)\,\mathrm{vec}(I_d)$.

□

APPENDIX G

ADDITIONAL RESULTS ON QUALITATIVE PREDICTIONS

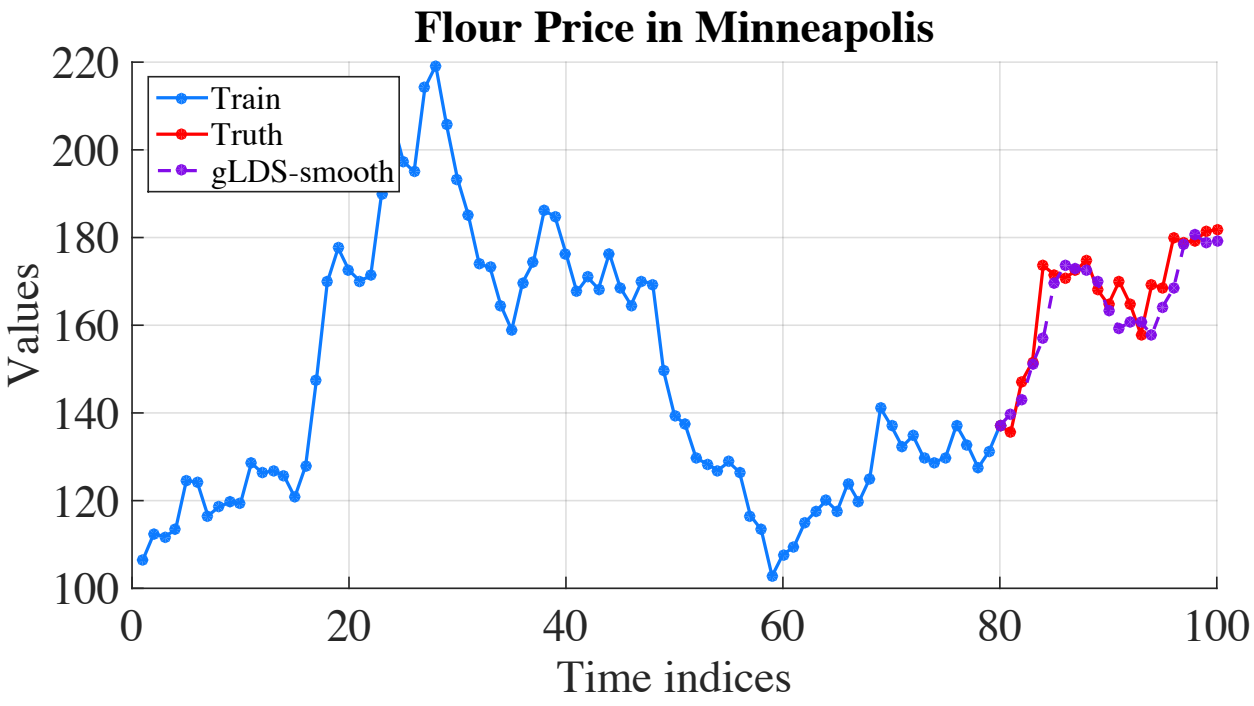


Figure 26: Predictions for flour price series in Minneapolis by using gLDS-smooth.

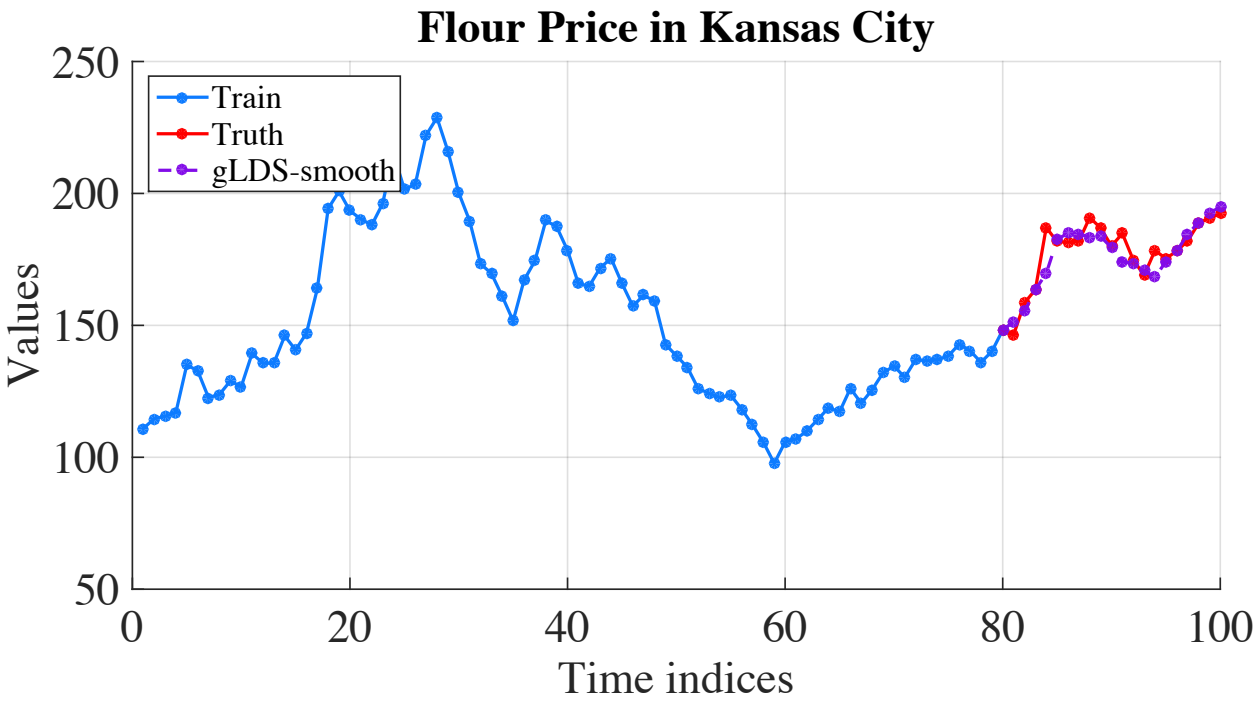


Figure 27: Predictions for flour price series in Kansas City by using gLDS-smooth.

APPENDIX H

ADDITIONAL RESULTS ON STABILITY EFFECTS

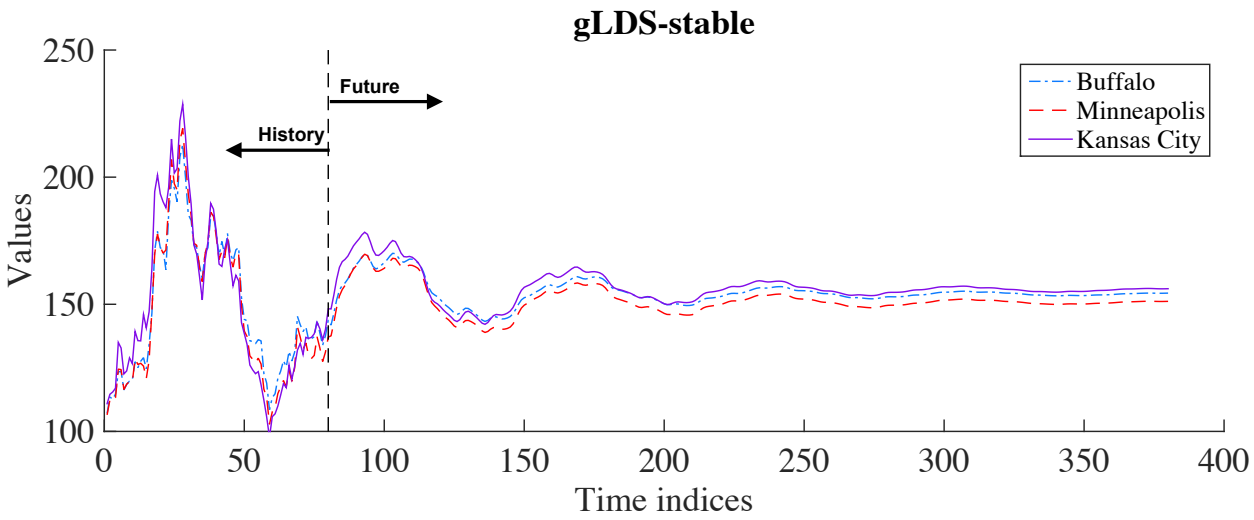


Figure 28: Training data and simulated sequences from gLDS-stable model in *fourprice* data.

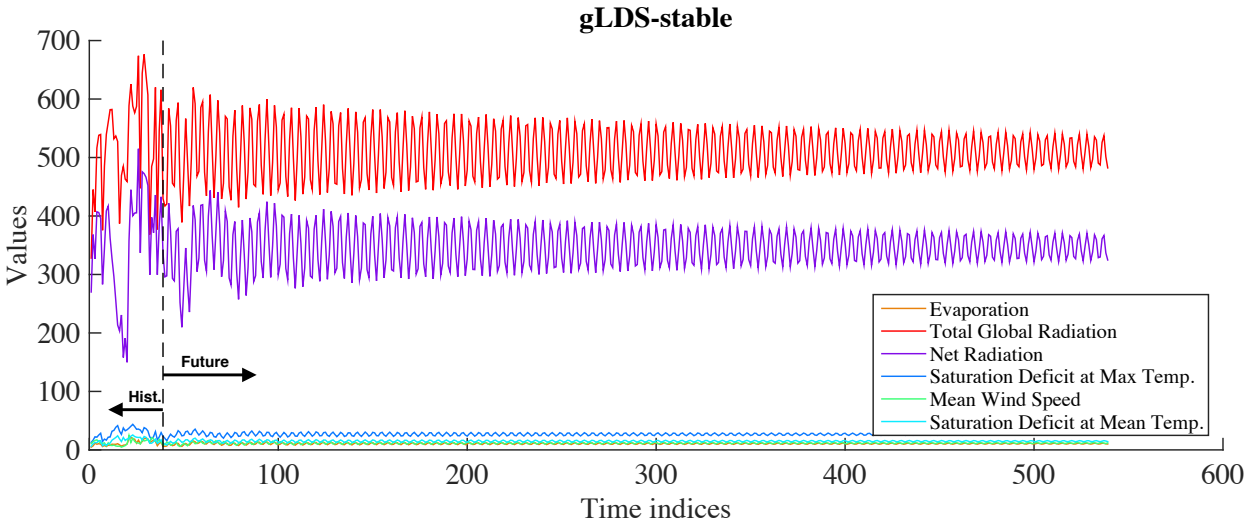


Figure 29: Training data and simulated sequences from gLDS-stable model in *h2o_evap* data.

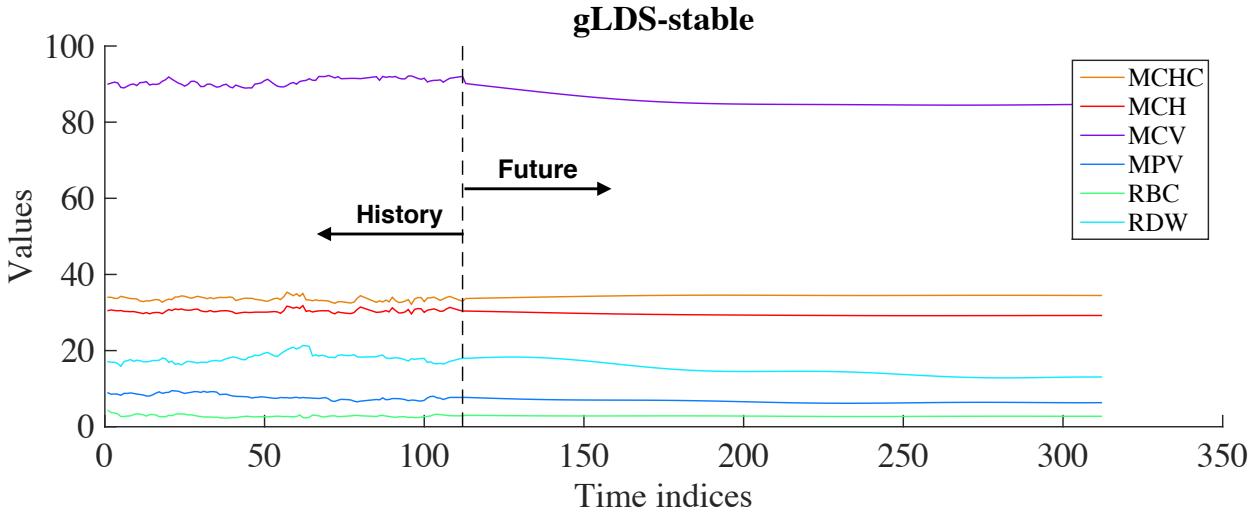


Figure 30: Training data and simulated sequences from gLDS-stable model in *clinical* data for one patient.

APPENDIX I

ADDITIONAL RESULTS ON SPARSIFICATION EFFECTS

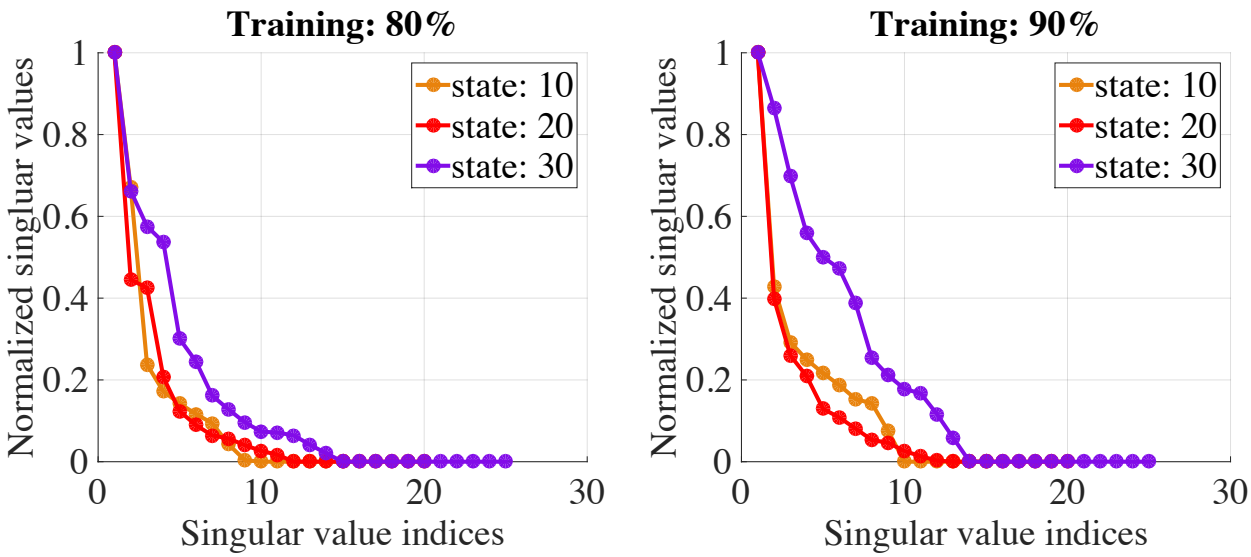


Figure 31: Intrinsic dimensionality recovery in *evap* data.

APPENDIX J

OVERALL PREDICTION PERFORMANCE

The experimental results of overall prediction performance on all ten lab tests are shown in Figure 32. Detailed numerical results are shown in Table 11.

Table 11: MAE on CBC test samples for overall prediction tasks.

Method	AR	LDS	GP	WAR	HDSGA	WLDS	HDSGL
WBC	4.7941 ± 0.0027	4.6805 ± 0.0026	5.0235 ± 0.0025	4.6400 ± 0.0026	4.5390 ± 0.0026	4.5720 ± 0.0027	4.4710 ± 0.0027
HCT	3.6893 ± 0.0007	3.3925 ± 0.0007	3.4253 ± 0.0007	3.5431 ± 0.0007	3.5315 ± 0.0007	3.3271 ± 0.0007	3.2177 ± 0.0007
HGB	1.3218 ± 0.0004	1.1755 ± 0.0004	1.1208 ± 0.0004	1.3198 ± 0.0004	1.3171 ± 0.0004	1.1577 ± 0.0004	1.1348 ± 0.0004
MCHC	0.6012 ± 0.0004	0.5959 ± 0.0004	0.6724 ± 0.0004	0.5963 ± 0.0004	0.5458 ± 0.0004	0.5701 ± 0.0004	0.5297 ± 0.0004
MCH	0.9941 ± 0.0006	0.9091 ± 0.0007	1.1154 ± 0.0007	0.8480 ± 0.0006	0.7975 ± 0.0006	0.8033 ± 0.0007	0.7831 ± 0.0007
MCV	2.5619 ± 0.0012	2.3410 ± 0.0014	2.8034 ± 0.0018	2.0814 ± 0.0012	1.9804 ± 0.0012	2.0294 ± 0.0013	1.9284 ± 0.0013
MPV	0.9412 ± 0.0005	0.9029 ± 0.0005	1.1392 ± 0.0005	0.9059 ± 0.0005	0.8554 ± 0.0005	0.8406 ± 0.0005	0.7901 ± 0.0005
PLT	102.5242 ± 0.0587	92.2360 ± 0.0561	120.4928 ± 0.0665	103.6683 ± 0.0587	100.6288 ± 0.0587	88.9408 ± 0.0555	85.8991 ± 0.0555
RBC	0.4242 ± 0.0002	0.3812 ± 0.0002	0.4453 ± 0.0002	0.4168 ± 0.0002	0.3663 ± 0.0002	0.3362 ± 0.0002	0.3059 ± 0.0002
RDW	1.4216 ± 0.0010	1.3860 ± 0.0010	1.8794 ± 0.0010	1.3427 ± 0.0010	1.2417 ± 0.0010	1.3185 ± 0.0010	1.2174 ± 0.0010

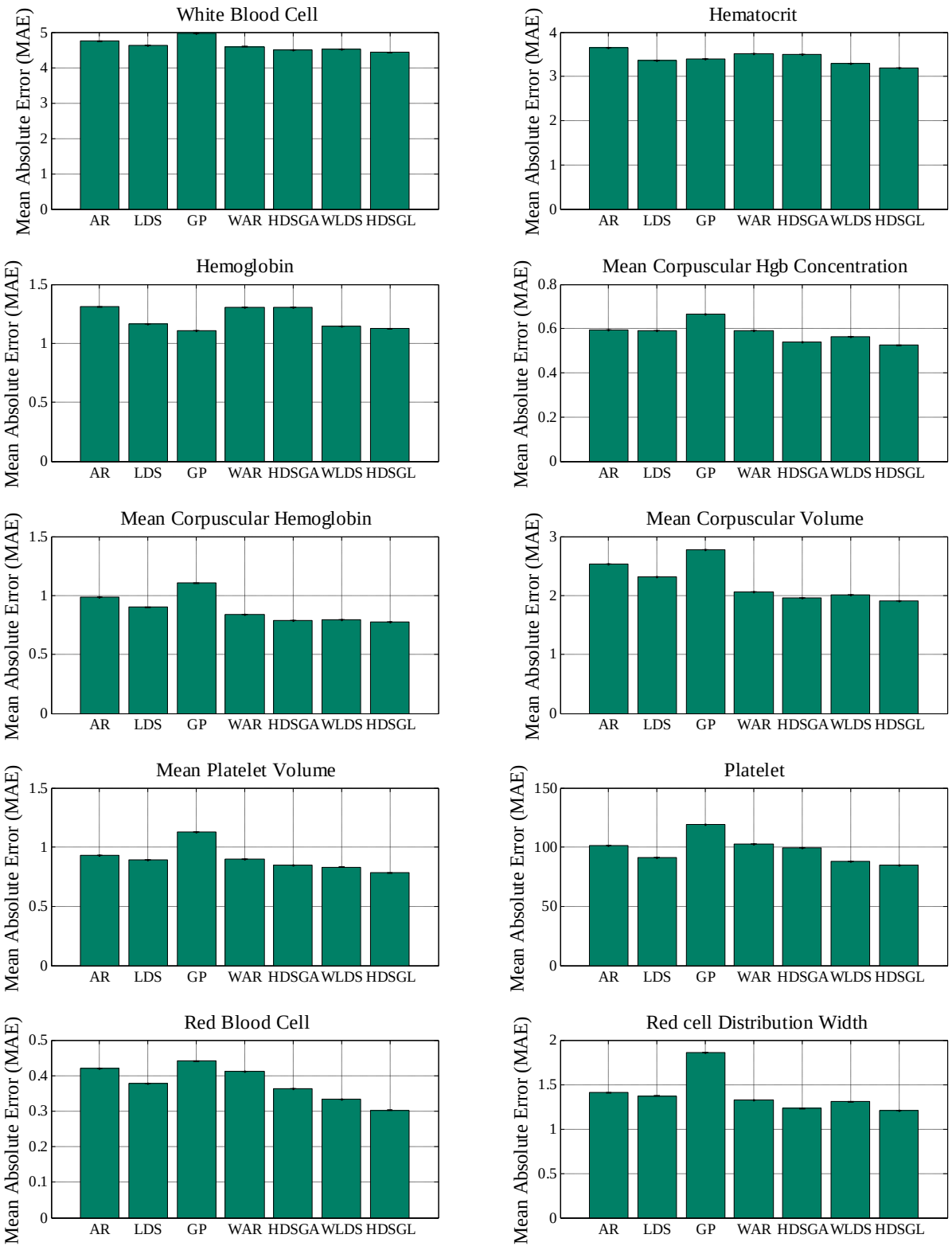


Figure 32: MAE on ten CBC lab tests for overall predictions.

APPENDIX K

SHORT-TERM PREDICTION PERFORMANCE

The experimental results of short-term prediction performance on all ten lab tests are shown in Figure 33. Detailed numerical results are shown in Table 12.

Table 12: MAE on CBC test samples for short-term prediction tasks.

Method	AR	LDS	GP	WAR	HDSGA	WLDS	HDSGL
WBC	3.5072 ± 0.0103	3.3998 ± 0.0103	3.9007 ± 0.0111	3.4993 ± 0.0108	3.2973 ± 0.0108	3.4756 ± 0.0116	3.2170 ± 0.0116
HCT	3.4376 ± 0.0096	2.9078 ± 0.0093	3.6121 ± 0.0110	2.9608 ± 0.0087	2.7588 ± 0.0087	3.0218 ± 0.0097	2.8859 ± 0.0100
HGB	0.9972 ± 0.0021	0.9528 ± 0.0023	1.2865 ± 0.0042	0.9627 ± 0.0026	0.8617 ± 0.0026	0.8810 ± 0.0033	0.9227 ± 0.0033
MCHC	0.4098 ± 0.0010	0.4019 ± 0.0011	0.5384 ± 0.0015	0.4091 ± 0.0011	0.3687 ± 0.0011	0.3739 ± 0.0014	0.3129 ± 0.0014
MCH	0.5439 ± 0.0021	0.4911 ± 0.0021	0.6975 ± 0.0045	0.4998 ± 0.0021	0.4594 ± 0.0021	0.5148 ± 0.0042	0.4522 ± 0.0042
MCV	1.3327 ± 0.0057	1.2458 ± 0.0058	1.9629 ± 0.0125	1.2734 ± 0.0059	1.2330 ± 0.0059	1.2288 ± 0.0115	1.1729 ± 0.0115
MPV	0.4628 ± 0.0014	0.4122 ± 0.0014	0.6472 ± 0.0019	0.4213 ± 0.0014	0.3708 ± 0.0014	0.4066 ± 0.0019	0.3055 ± 0.0019
PLT	49.6031 ± 0.1156	43.4901 ± 0.1191	71.5818 ± 0.1603	45.0584 ± 0.1208	45.0584 ± 0.1208	40.8308 ± 0.1669	40.2046 ± 0.1669
RBC	0.3862 ± 0.0011	0.3685 ± 0.0011	0.4120 ± 0.0015	0.3670 ± 0.0013	0.3670 ± 0.0013	0.3346 ± 0.0013	0.2820 ± 0.0013
RDW	0.5036 ± 0.0010	0.4019 ± 0.0012	1.2055 ± 0.0043	0.4476 ± 0.0012	0.3971 ± 0.0012	0.4136 ± 0.0044	0.3751 ± 0.0044

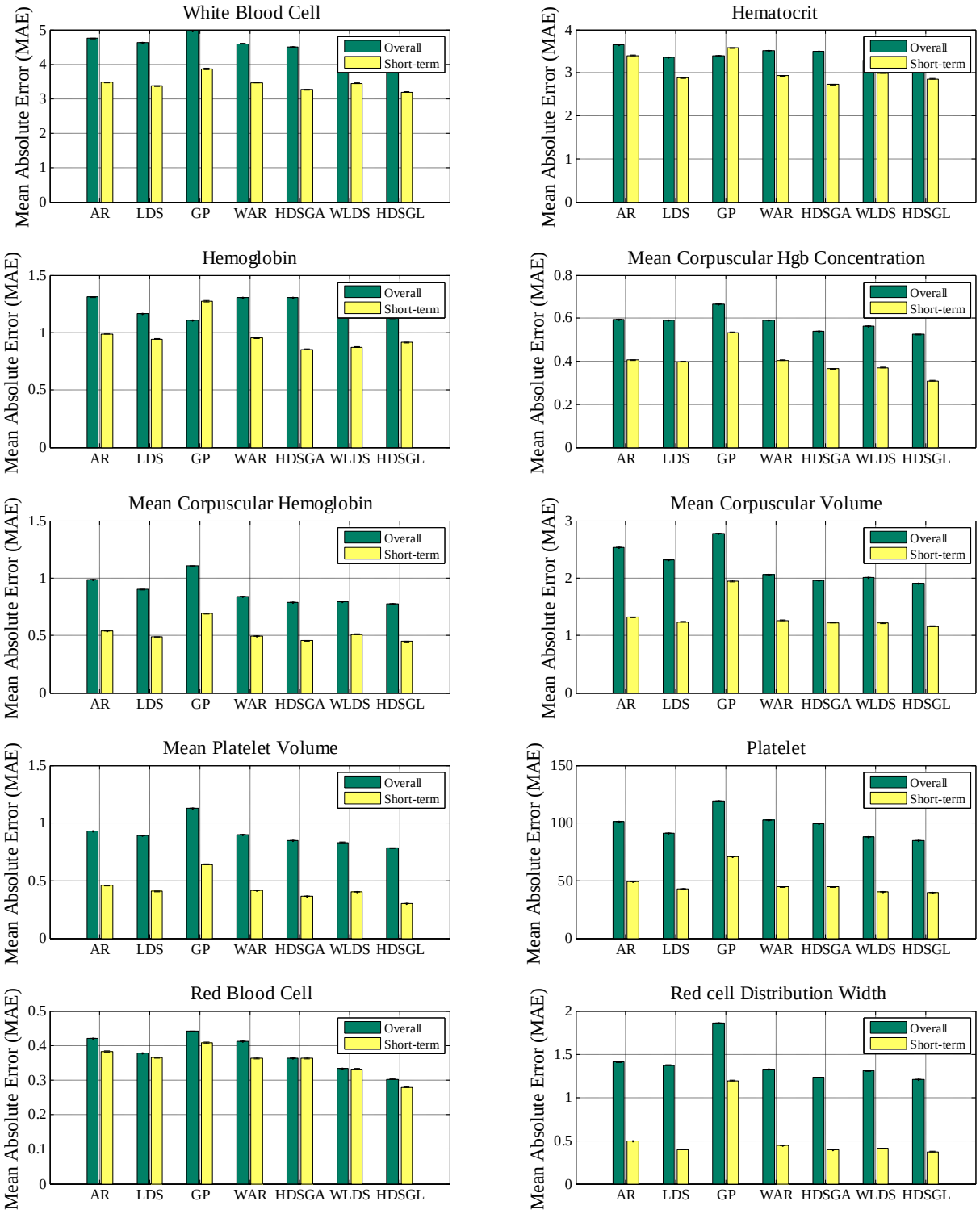


Figure 33: MAE on ten CBC lab tests for short-term predictions.

APPENDIX L

CLINICAL EXPERT EVALUATION

The clinical expert evaluation results of both overall and short-term prediction performance on all ten lab tests are shown in Figure 34. Detailed numerical results are shown in Table 13 and Table 14.

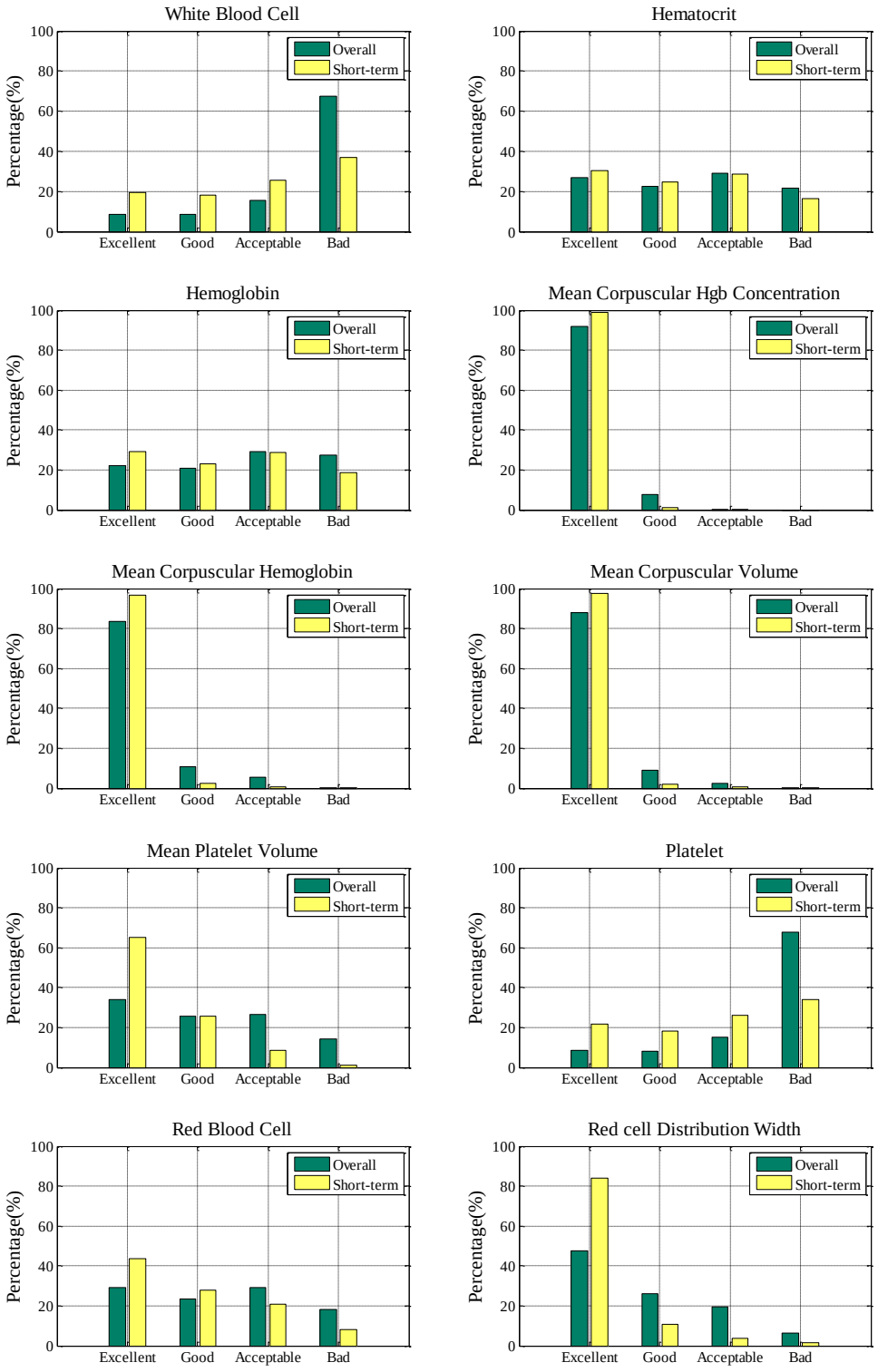


Figure 34: Clinical evaluations of HDSGL for both overall prediction and short-term prediction quality distributions.

Table 13: Clinical evaluation for overall prediction.

	Excellent	Good	Acceptable	Bad
WBC	0.0844	0.0842	0.1557	0.6757
HCT	0.2696	0.2228	0.2911	0.2165
HGB	0.2222	0.2083	0.2938	0.2757
MCHC	0.9205	0.0770	0.0024	0.0000
MCH	0.8343	0.1082	0.0539	0.0036
MCV	0.8815	0.0919	0.0263	0.0004
MPV	0.3411	0.2545	0.2638	0.1405
PLT	0.0866	0.0827	0.1514	0.6793
RBC	0.2919	0.2348	0.2932	0.1800
RDW	0.4754	0.2630	0.1968	0.0648

Table 14: Clinical evaluation for short-term prediction.

	Excellent	Good	Acceptable	Bad
WBC	0.1964	0.1798	0.2538	0.3699
HCT	0.3044	0.2469	0.2868	0.1619
HGB	0.2924	0.2320	0.2882	0.1873
MCHC	0.9903	0.0095	0.0002	0.0000
MCH	0.9683	0.0243	0.0063	0.0011
MCV	0.9749	0.0184	0.0057	0.0011
MPV	0.6497	0.2567	0.0839	0.0096
PLT	0.2164	0.1839	0.2611	0.3386
RBC	0.4355	0.2788	0.2068	0.0790
RDW	0.8400	0.1075	0.0389	0.0136

APPENDIX M

AVERAGE-MAPE RESULTS OF MODEL ADAPTATION APPROACHES

Table 15: Average-MAPE results (means and standard errors) for the different initial observation sequence lengths. reGP and reMTGP are short for rLDS+reGP and rLDS+reMTGP. The best performing method is shown in **bold**. Also in bold are the methods that are not statistically significantly different from the best method at 0.05 significance level.

Method	1	2	3	4	5	6	7	8	9
P_Mean	6.80±0.07	6.76±0.08	6.72±0.08	6.68±0.08	6.66±0.08	6.67±0.09	6.67±0.09	6.70±0.10	6.77±0.10
P_GP	3.05±0.04	3.04±0.04	3.02±0.05	3.00±0.05	3.01±0.05	2.99±0.05	2.97±0.05	2.94±0.05	2.93±0.05
P_MTGP	3.33±0.05	3.32±0.05	3.30±0.05	3.27±0.05	3.26±0.05	3.25±0.06	3.22±0.06	3.21±0.06	3.19±0.06
P_rLDS	**2.96±0.04**	**2.94±0.04**	**2.92±0.04**	**2.88±0.04**	**2.88±0.05**	**2.87±0.05**	**2.84±0.05**	**2.83±0.05**	**2.81±0.05**
I_Mean	4.28±0.06	4.32±0.06	4.37±0.07	4.38±0.07	4.42±0.07	4.46±0.07	4.47±0.08	4.51±0.08	4.55±0.08
I_GP	3.33±0.05	3.32±0.05	3.29±0.05	3.24±0.05	3.23±0.05	3.21±0.06	3.16±0.05	3.15±0.06	3.13±0.06
I_MTGP	3.72±0.06	3.72±0.06	3.73±0.06	3.70±0.06	3.70±0.06	3.68±0.07	3.63±0.07	3.61±0.07	3.57±0.07
I_rLDS	4.94±0.23	5.02±0.24	5.08±0.26	5.05±0.27	5.03±0.29	4.51±0.11	4.28±0.09	4.16±0.09	4.04±0.10
reGP	3.64±0.06	3.33±0.05	3.22±0.05	3.13±0.05	3.08±0.06	3.03±0.06	2.96±0.06	2.89±0.06	**2.84±0.06**
reMTGP	3.60±0.06	3.29±0.05	3.19±0.05	3.10±0.05	3.05±0.05	3.00±0.06	2.94±0.06	2.88±0.06	**2.84±0.06**
Method	10	11	12	13	14	15	16	17	18
P_Mean	6.82±0.11	6.91±0.12	6.97±0.13	7.11±0.14	7.23±0.15	7.39±0.16	7.50±0.16	7.58±0.17	7.70±0.18
P_GP	2.92±0.05	2.90±0.06	2.90±0.06	2.90±0.07	2.91±0.07	2.91±0.08	2.88±0.08	2.83±0.08	2.82±0.08
P_MTGP	3.17±0.06	3.16±0.07	3.14±0.07	3.13±0.08	3.12±0.08	3.13±0.09	3.09±0.08	3.06±0.08	3.08±0.09
P_rLDS	**2.81±0.05**	**2.79±0.06**	**2.78±0.06**	2.80±0.06	2.81±0.07	2.82±0.07	2.82±0.07	2.77±0.07	2.77±0.08
I_Mean	4.57±0.08	4.58±0.09	4.58±0.09	4.63±0.10	4.70±0.10	4.76±0.11	4.79±0.12	4.80±0.12	4.81±0.13
I_GP	3.09±0.06	3.05±0.06	3.02±0.06	3.02±0.06	3.04±0.07	3.01±0.07	3.00±0.07	2.98±0.08	2.95±0.08
I_MTGP	3.51±0.07	3.44±0.07	3.38±0.07	3.36±0.07	3.37±0.08	3.35±0.09	3.31±0.09	3.29±0.09	3.23±0.10
I_rLDS	3.87±0.09	3.70±0.08	3.60±0.08	3.55±0.09	3.48±0.09	3.40±0.09	3.33±0.09	3.26±0.09	3.21±0.10
reGP	**2.81±0.06**	**2.76±0.06**	**2.73±0.06**	**2.71±0.06**	**2.70±0.07**	**2.67±0.07**	**2.65±0.07**	**2.60±0.08**	**2.58±0.08**
reMTGP	**2.81±0.06**	**2.76±0.06**	**2.74±0.06**	**2.72±0.06**	**2.72±0.07**	**2.70±0.07**	**2.67±0.07**	**2.63±0.07**	**2.62±0.08**
Method	19	20	21	22	23	24	25	26	27
P_Mean	7.74±0.19	7.78±0.20	7.88±0.21	8.01±0.22	8.13±0.23	8.17±0.23	8.28±0.25	8.49±0.26	8.66±0.27
P_GP	2.84±0.09	2.80±0.09	2.78±0.09	2.82±0.10	2.82±0.10	2.80±0.10	2.76±0.10	2.78±0.11	2.82±0.11
P_MTGP	3.11±0.09	3.10±0.10	3.08±0.10	3.13±0.11	3.18±0.12	3.15±0.11	3.13±0.12	3.14±0.12	3.15±0.13
P_rLDS	2.79±0.08	2.79±0.08	2.80±0.09	2.84±0.09	2.87±0.10	2.86±0.10	2.85±0.10	2.89±0.10	2.96±0.10
I_Mean	4.80±0.13	4.79±0.14	4.80±0.14	4.85±0.15	4.90±0.15	4.89±0.15	4.93±0.16	5.03±0.16	5.12±0.17
I_GP	2.93±0.08	2.92±0.08	2.91±0.09	2.95±0.10	2.92±0.10	2.89±0.09	2.88±0.10	2.90±0.10	2.92±0.11
I_MTGP	3.20±0.10	3.18±0.10	3.19±0.11	3.23±0.11	3.24±0.12	3.24±0.12	3.23±0.13	3.28±0.13	3.31±0.14
I_rLDS	3.18±0.10	3.16±0.10	3.16±0.11	3.15±0.11	3.13±0.11	3.12±0.11	3.10±0.11	3.05±0.10	3.08±0.11
reGP	**2.60±0.08**	**2.57±0.09**	**2.53±0.09**	**2.53±0.09**	**2.53±0.10**	**2.51±0.09**	**2.44±0.09**	**2.46±0.09**	**2.48±0.10**
reMTGP	**2.63±0.08**	**2.60±0.08**	2.59±0.09	2.61±0.09	2.60±0.10	2.58±0.10	2.53±0.10	2.54±0.10	2.58±0.11

Method	28	29	30	31	32	33	34	35	36
P_Mean	8.79±0.28	8.92±0.29	9.00±0.30	9.11±0.31	9.16±0.33	9.22±0.34	9.22±0.36	9.23±0.37	9.21±0.38
P_GP	2.80±0.11	2.85±0.12	2.87±0.13	2.89±0.13	2.96±0.14	3.02±0.15	3.08±0.15	3.08±0.16	3.10±0.16
P_MTGP	3.11±0.13	3.18±0.14	3.22±0.14	3.26±0.15	3.26±0.16	3.28±0.17	3.35±0.17	3.35±0.18	3.38±0.19
P_rLDS	2.96±0.11	3.00±0.11	3.04±0.12	3.04±0.12	3.07±0.13	3.12±0.14	3.18±0.14	3.18±0.15	3.18±0.15
I_Mean	5.14±0.17	5.17±0.18	5.16±0.18	5.17±0.18	5.14±0.19	5.15±0.20	5.16±0.20	5.14±0.21	5.09±0.21
I_GP	2.91±0.11	2.93±0.11	2.93±0.12	2.95±0.12	2.96±0.13	3.02±0.13	3.06±0.14	3.05±0.14	3.07±0.14
I_MTGP	3.30±0.15	3.34±0.15	3.37±0.15	3.42±0.16	3.46±0.17	3.47±0.18	3.54±0.19	3.55±0.20	3.57±0.20
I_rLDS	3.07±0.11	3.07±0.12	3.07±0.12	3.08±0.12	3.07±0.13	3.07±0.13	3.10±0.14	3.06±0.14	3.07±0.14
reGP	**2.46±0.10**	**2.49±0.11**	**2.52±0.11**	**2.51±0.12**	**2.54±0.12**	**2.59±0.13**	**2.64±0.14**	**2.62±0.14**	**2.63±0.14**
reMTGP	2.57±0.11	2.59±0.12	2.61±0.12	**2.60±0.13**	**2.62±0.13**	**2.66±0.14**	**2.71±0.15**	**2.70±0.15**	**2.71±0.16**

Method	37	38	39	40	41	42	43	44	45
P_Mean	9.26±0.39	9.23±0.39	9.26±0.40	9.29±0.40	9.27±0.40	9.32±0.40	9.38±0.41	9.42±0.42	9.48±0.42
P_GP	3.13±0.17	3.14±0.17	3.18±0.18	3.21±0.18	3.23±0.19	3.29±0.20	3.33±0.20	3.35±0.20	3.37±0.20
P_MTGP	3.41±0.19	3.44±0.20	3.51±0.20	3.55±0.21	3.52±0.21	3.58±0.22	3.63±0.23	3.65±0.23	3.68±0.24
P_rLDS	3.21±0.15	3.23±0.15	3.27±0.16	3.30±0.16	3.30±0.17	3.35±0.17	3.40±0.17	3.43±0.18	3.46±0.18
I_Mean	5.07±0.21	5.02±0.22	5.04±0.21	5.05±0.21	5.00±0.21	5.03±0.21	5.06±0.21	5.06±0.22	5.07±0.22
I_GP	3.06±0.14	3.08±0.14	3.10±0.15	3.12±0.15	3.11±0.15	3.12±0.16	3.16±0.16	3.19±0.17	3.20±0.16
I_MTGP	3.54±0.21	3.57±0.21	3.63±0.22	3.64±0.23	3.63±0.23	3.65±0.24	3.67±0.24	3.66±0.25	3.63±0.25
I_rLDS	3.04±0.14	3.06±0.14	3.12±0.15	3.17±0.15	3.20±0.16	3.24±0.16	3.30±0.16	3.32±0.17	3.33±0.17
reGP	**2.64±0.14**	**2.67±0.15**	**2.68±0.15**	**2.71±0.16**	**2.72±0.16**	**2.77±0.17**	**2.81±0.17**	**2.84±0.17**	**2.88±0.18**
reMTGP	**2.73±0.16**	**2.75±0.17**	**2.79±0.17**	**2.82±0.18**	**2.83±0.18**	**2.89±0.19**	**2.93±0.19**	**2.97±0.20**	**3.01±0.20**

Method	46	47	48	49	50	51	52	53	54
P_Mean	9.58±0.42	9.67±0.43	9.72±0.44	9.75±0.45	9.72±0.46	9.76±0.47	9.80±0.48	9.98±0.48	10.00±0.47
P_GP	3.33±0.21	3.37±0.21	3.43±0.22	3.48±0.22	3.48±0.23	3.42±0.22	3.46±0.23	3.44±0.24	3.37±0.23
P_MTGP	3.67±0.25	3.70±0.25	3.73±0.26	3.76±0.27	3.72±0.27	3.54±0.25	3.56±0.25	3.58±0.26	3.56±0.26
P_rLDS	3.45±0.19	3.42±0.19	3.42±0.19	3.43±0.20	3.41±0.21	3.34±0.20	3.29±0.20	3.33±0.21	3.31±0.21
I_Mean	5.12±0.22	5.12±0.23	5.10±0.24	5.05±0.25	4.97±0.25	4.94±0.26	4.90±0.26	4.91±0.26	4.83±0.25
I_GP	3.18±0.17	**3.17±0.17**	**3.18±0.18**	**3.20±0.18**	**3.20±0.19**	**3.11±0.18**	**3.09±0.19**	**3.08±0.20**	**3.02±0.19**
I_MTGP	3.61±0.26	3.63±0.27	3.59±0.28	3.57±0.28	3.49±0.28	3.41±0.28	3.48±0.28	3.46±0.29	3.31±0.25
I_rLDS	3.32±0.18	3.31±0.17	3.33±0.17	3.33±0.18	3.33±0.19	3.29±0.19	3.26±0.19	3.28±0.20	3.25±0.20
reGP	**2.86±0.18**	**2.87±0.19**	**2.91±0.19**	**2.94±0.20**	**2.97±0.21**	**2.91±0.20**	**2.92±0.20**	**2.89±0.20**	**2.83±0.21**
reMTGP	**2.99±0.21**	**2.96±0.20**	**2.96±0.21**	**2.97±0.22**	**2.98±0.23**	**2.94±0.22**	**2.96±0.23**	**2.93±0.23**	**2.79±0.20**

Method	55	56	57	58	59	60
P_Mean	9.99±0.46	9.99±0.44	10.18±0.44	10.24±0.45	10.28±0.46	10.26±0.48
P_GP	3.35±0.23	3.45±0.23	3.35±0.20	3.35±0.21	3.31±0.21	3.29±0.22
P_MTGP	3.49±0.26	3.56±0.27	3.54±0.28	3.55±0.29	3.60±0.29	3.59±0.30
P_rLDS	3.22±0.20	3.26±0.19	3.19±0.17	3.16±0.17	3.18±0.18	3.17±0.18
I_Mean	4.75±0.24	4.66±0.22	4.67±0.23	4.68±0.24	4.71±0.24	4.71±0.25
I_GP	**2.96±0.19**	**2.97±0.19**	**2.91±0.18**	**2.92±0.18**	**2.96±0.18**	**2.98±0.19**
I_MTGP	3.21±0.24	**3.09±0.19**	**3.02±0.18**	3.05±0.18	3.10±0.18	**3.04±0.18**
I_rLDS	**3.10±0.18**	**3.06±0.19**	**2.98±0.19**	**2.96±0.19**	**2.98±0.19**	**2.97±0.20**
reGP	**2.77±0.20**	**2.83±0.21**	**2.71±0.16**	**2.70±0.17**	**2.72±0.17**	**2.75±0.18**
reMTGP	**2.76±0.21**	**2.81±0.22**	**2.68±0.17**	**2.65±0.18**	**2.67±0.18**	**2.72±0.18**

APPENDIX N

COMPARISON OF RESULTS FOR POPULATION BASED AND PATIENT SPECIFIC MODELS

Table 16: Average-MAPE results (means and standard errors) of all models in the pool and two wFTL methods for the different initial observation sequence lengths. The best performing method is shown in **bold**. Also in bold are the methods that are not statistically significantly different from the best method at 0.05 significance level.

Method	1	2	3	4	5	6	7	8	9
P_Mean	6.80±0.07	6.76±0.08	6.72±0.08	6.68±0.08	6.66±0.08	6.67±0.09	6.67±0.09	6.70±0.10	6.77±0.10
P_GP	3.05±0.04	3.04±0.04	3.02±0.05	3.00±0.05	3.01±0.05	2.99±0.05	2.97±0.05	2.94±0.05	2.93±0.05
P_MTGP	3.33±0.05	3.32±0.05	3.30±0.05	3.27±0.05	3.26±0.05	3.25±0.06	3.22±0.06	3.21±0.06	3.19±0.06
P_rLDS	**2.96±0.04**	**2.94±0.04**	**2.92±0.04**	**2.88±0.04**	**2.88±0.05**	**2.87±0.05**	**2.84±0.05**	**2.83±0.05**	**2.81±0.05**
I_Mean	4.28±0.06	4.32±0.06	4.37±0.07	4.38±0.07	4.42±0.07	4.46±0.07	4.47±0.08	4.51±0.08	4.55±0.08
I_GP	3.33±0.05	3.32±0.05	3.29±0.05	3.24±0.05	3.23±0.05	3.21±0.06	3.16±0.05	3.15±0.06	3.13±0.06
I_MTGP	3.72±0.06	3.72±0.06	3.73±0.06	3.70±0.06	3.70±0.06	3.68±0.07	3.63±0.07	3.61±0.07	3.57±0.07
I_rLDS	4.94±0.23	5.02±0.24	5.08±0.26	5.05±0.27	5.03±0.29	4.51±0.11	4.28±0.09	4.16±0.09	4.04±0.10
wFTL_se	3.24±0.05	3.22±0.05	3.20±0.05	3.15±0.06	3.14±0.06	3.09±0.06	3.04±0.05	2.99±0.06	2.94±0.05
wFTL_mr	3.24±0.05	3.22±0.05	3.20±0.05	3.16±0.06	3.14±0.06	3.09±0.05	3.04±0.05	2.99±0.05	2.94±0.05
Method	10	11	12	13	14	15	16	17	18
P_Mean	6.82±0.11	6.91±0.12	6.97±0.13	7.11±0.14	7.23±0.15	7.39±0.16	7.50±0.16	7.58±0.17	7.70±0.18
P_GP	2.92±0.05	2.90±0.06	2.90±0.06	2.90±0.07	2.91±0.07	**2.91±0.08**	**2.88±0.08**	**2.83±0.08**	**2.82±0.08**
P_MTGP	3.17±0.06	3.16±0.07	3.14±0.07	3.13±0.08	3.12±0.08	3.13±0.09	3.09±0.08	3.06±0.08	3.08±0.09
P_rLDS	**2.81±0.05**	**2.79±0.06**	**2.78±0.06**	**2.80±0.06**	**2.81±0.07**	**2.82±0.07**	**2.82±0.07**	**2.77±0.07**	**2.77±0.08**
I_Mean	4.57±0.08	4.58±0.09	4.58±0.09	4.63±0.10	4.70±0.10	4.76±0.11	4.79±0.12	4.80±0.12	4.81±0.13
I_GP	3.09±0.06	3.05±0.06	3.02±0.06	3.02±0.06	3.04±0.07	3.01±0.07	3.00±0.07	2.98±0.08	2.95±0.08
I_MTGP	3.51±0.07	3.44±0.07	3.38±0.07	3.36±0.07	3.37±0.08	3.35±0.09	3.31±0.09	3.29±0.09	3.23±0.10
I_rLDS	3.87±0.09	3.70±0.08	3.60±0.08	3.55±0.09	3.48±0.09	3.40±0.09	3.33±0.09	3.26±0.09	3.21±0.10
wFTL_se	2.92±0.06	2.88±0.06	**2.83±0.06**	**2.82±0.06**	**2.83±0.07**	**2.82±0.07**	**2.80±0.07**	**2.75±0.08**	**2.74±0.08**
wFTL_mr	2.90±0.05	**2.85±0.05**	**2.82±0.06**	**2.82±0.06**	**2.83±0.06**	**2.82±0.07**	**2.82±0.07**	**2.78±0.07**	**2.75±0.08**

Method	19	20	21	22	23	24	25	26	27
P_Mean	7.74±0.19	7.78±0.20	7.88±0.21	8.01±0.22	8.13±0.23	8.17±0.23	8.28±0.25	8.49±0.26	8.66±0.27
P_GP	**2.84±0.09**	**2.80±0.09**	**2.78±0.09**	**2.82±0.10**	**2.82±0.10**	**2.80±0.10**	**2.76±0.10**	**2.78±0.11**	2.82±0.11
P_MTGP	3.11±0.09	3.10±0.10	3.08±0.10	3.13±0.11	3.18±0.12	3.15±0.11	3.13±0.12	3.14±0.12	3.15±0.13
P_rLDS	**2.79±0.08**	**2.79±0.08**	**2.80±0.09**	**2.84±0.09**	2.87±0.10	2.86±0.10	2.85±0.10	2.89±0.10	2.96±0.10
I_Mean	4.80±0.13	4.79±0.14	4.80±0.14	4.85±0.15	4.90±0.15	4.89±0.15	4.93±0.16	5.03±0.16	5.12±0.17
I_GP	2.93±0.08	2.92±0.08	2.91±0.09	2.95±0.10	2.92±0.10	2.89±0.09	2.88±0.10	2.90±0.10	2.92±0.11
I_MTGP	3.20±0.10	3.18±0.10	3.19±0.11	3.23±0.11	3.24±0.12	3.24±0.12	3.23±0.13	3.28±0.13	3.31±0.14
I_rLDS	3.18±0.10	3.16±0.10	3.16±0.11	3.15±0.11	3.13±0.11	3.12±0.11	3.10±0.11	3.05±0.10	3.08±0.11
wFTL_se	**2.73±0.08**	**2.71±0.09**	**2.71±0.09**	**2.74±0.10**	**2.73±0.10**	**2.70±0.10**	**2.66±0.09**	**2.65±0.10**	**2.66±0.10**
wFTL_mr	**2.74±0.08**	**2.73±0.08**	**2.72±0.09**	**2.75±0.09**	**2.74±0.09**	**2.73±0.09**	**2.69±0.09**	**2.69±0.10**	**2.70±0.10**

Method	28	29	30	31	32	33	34	35	36
P_Mean	8.79±0.28	8.92±0.29	9.00±0.30	9.11±0.31	9.16±0.33	9.22±0.34	9.22±0.36	9.23±0.37	9.21±0.38
P_GP	2.80±0.11	2.85±0.12	2.87±0.13	2.89±0.13	2.96±0.14	3.02±0.15	3.08±0.15	3.08±0.16	3.10±0.16
P_MTGP	3.11±0.13	3.18±0.14	3.22±0.14	3.26±0.15	3.26±0.16	3.28±0.17	3.35±0.17	3.35±0.18	3.38±0.19
P_rLDS	2.96±0.11	3.00±0.11	3.04±0.12	3.04±0.12	3.07±0.13	3.12±0.14	3.18±0.14	3.18±0.15	3.18±0.15
I_Mean	5.14±0.17	5.17±0.18	5.16±0.18	5.17±0.18	5.14±0.19	5.15±0.20	5.16±0.20	5.14±0.21	5.09±0.21
I_GP	2.91±0.11	2.93±0.11	2.93±0.12	2.95±0.12	2.96±0.13	3.02±0.13	3.06±0.14	3.05±0.14	3.07±0.14
I_MTGP	3.30±0.15	3.34±0.15	3.37±0.15	3.42±0.16	3.46±0.17	3.47±0.18	3.54±0.19	3.55±0.20	3.57±0.20
I_rLDS	3.07±0.11	3.07±0.12	3.07±0.12	3.08±0.12	3.07±0.13	3.07±0.13	3.10±0.14	3.06±0.14	3.07±0.14
wFTL_se	**2.62±0.11**	**2.64±0.11**	**2.66±0.12**	**2.67±0.12**	**2.70±0.13**	**2.74±0.14**	**2.79±0.14**	**2.79±0.15**	**2.81±0.15**
wFTL_mr	**2.65±0.10**	**2.68±0.11**	**2.71±0.12**	**2.73±0.12**	**2.76±0.13**	**2.81±0.14**	**2.86±0.14**	**2.85±0.15**	**2.87±0.15**

Method	37	38	39	40	41	42	43	44	45
P_Mean	9.26±0.39	9.23±0.39	9.26±0.40	9.29±0.40	9.27±0.40	9.32±0.40	9.38±0.41	9.42±0.42	9.48±0.42
P_GP	3.13±0.17	3.14±0.17	**3.18±0.18**	**3.21±0.18**	3.23±0.19	3.29±0.20	3.33±0.20	3.35±0.20	3.37±0.20
P_MTGP	3.41±0.19	3.44±0.20	3.51±0.20	3.55±0.21	3.52±0.21	3.58±0.22	3.63±0.23	3.65±0.23	3.68±0.24
P_rLDS	3.21±0.15	3.23±0.15	3.27±0.16	3.30±0.16	3.30±0.17	3.35±0.17	3.40±0.17	3.43±0.18	3.46±0.18
I_Mean	5.07±0.21	5.02±0.22	5.04±0.21	5.05±0.21	5.00±0.21	5.03±0.21	5.06±0.21	5.06±0.22	5.07±0.22
I_GP	3.06±0.14	3.08±0.14	**3.10±0.15**	**3.12±0.15**	**3.11±0.15**	**3.12±0.16**	**3.16±0.16**	**3.19±0.17**	**3.20±0.16**
I_MTGP	3.54±0.21	3.57±0.21	3.63±0.22	3.64±0.23	3.63±0.23	3.65±0.24	3.67±0.24	3.66±0.25	3.63±0.25
I_rLDS	3.04±0.14	**3.06±0.14**	3.12±0.15	3.17±0.15	3.20±0.16	3.24±0.16	3.30±0.16	3.32±0.17	3.33±0.17
wFTL_se	**2.83±0.15**	**2.86±0.16**	**2.90±0.16**	**2.93±0.17**	**2.92±0.17**	**2.94±0.18**	**3.01±0.18**	**3.03±0.19**	**3.03±0.19**
wFTL_mr	**2.88±0.15**	**2.91±0.16**	**2.94±0.16**	**2.96±0.17**	**2.95±0.17**	**2.98±0.18**	**3.04±0.18**	**3.07±0.19**	**3.09±0.19**

Method	46	47	48	49	50	51	52	53	54
P_Mean	9.58±0.42	9.67±0.43	9.72±0.44	9.75±0.45	9.72±0.46	9.76±0.47	9.80±0.48	9.98±0.48	10.00±0.47
P_GP	3.33±0.21	3.37±0.21	3.43±0.22	3.48±0.22	3.48±0.23	3.42±0.22	3.46±0.23	3.44±0.24	**3.37±0.23**
P_MTGP	3.67±0.25	3.70±0.25	3.73±0.26	3.76±0.27	3.72±0.27	3.54±0.25	3.56±0.25	3.58±0.26	3.56±0.26
P_rLDS	3.45±0.19	3.42±0.19	3.42±0.19	3.43±0.20	3.41±0.21	3.34±0.20	**3.29±0.20**	3.33±0.21	3.31±0.21
I_Mean	5.12±0.22	5.12±0.23	5.10±0.24	5.05±0.25	4.97±0.25	4.94±0.26	4.90±0.26	4.91±0.26	4.83±0.25
I_GP	**3.18±0.17**	**3.17±0.17**	**3.18±0.18**	**3.20±0.18**	**3.20±0.19**	**3.11±0.18**	**3.09±0.19**	**3.08±0.20**	**3.02±0.19**
I_MTGP	3.61±0.26	3.63±0.27	3.59±0.28	3.57±0.28	3.49±0.28	3.41±0.28	3.48±0.28	3.46±0.29	**3.31±0.25**
I_rLDS	3.32±0.18	3.31±0.17	3.33±0.17	**3.33±0.18**	**3.33±0.19**	3.29±0.19	**3.26±0.19**	**3.28±0.20**	**3.25±0.20**
wFTL_se	**2.98±0.19**	**2.98±0.19**	**3.04±0.20**	**3.08±0.20**	**3.08±0.21**	**2.99±0.19**	**3.01±0.20**	**3.00±0.21**	**2.98±0.21**
wFTL_mr	**3.05±0.20**	3.09±0.20	**3.15±0.20**	**3.19±0.21**	**3.17±0.21**	**3.07±0.20**	**3.08±0.21**	**3.06±0.21**	**3.05±0.22**

Method	55	56	57	58	59	60
P_Mean	9.99±0.46	9.99±0.44	10.18±0.44	10.24±0.45	10.28±0.46	10.26±0.48
P_GP	3.35±0.23	3.45±0.23	3.35±0.20	3.35±0.21	3.31±0.21	3.29±0.22
P_MTGP	3.49±0.26	3.56±0.27	3.54±0.28	3.55±0.29	3.60±0.29	3.59±0.30
P_rLDS	3.22±0.20	3.26±0.19	3.19±0.17	3.16±0.17	3.18±0.18	**3.17±0.18**
I_Mean	4.75±0.24	4.66±0.22	4.67±0.23	4.68±0.24	4.71±0.24	4.71±0.25
I_GP	**2.96±0.19**	**2.97±0.19**	**2.91±0.18**	**2.92±0.18**	**2.96±0.18**	**2.98±0.19**
I_MTGP	**3.21±0.24**	**3.09±0.19**	**3.02±0.18**	**3.05±0.18**	**3.10±0.18**	**3.04±0.18**
I_rLDS	**3.10±0.18**	**3.06±0.19**	**2.98±0.19**	**2.96±0.19**	**2.98±0.19**	**2.97±0.20**
wFTL_se	**2.87±0.19**	**2.86±0.18**	**2.76±0.16**	**2.75±0.16**	**2.78±0.17**	**2.81±0.17**
wFTL_mr	**2.97±0.21**	**2.96±0.20**	**2.83±0.16**	**2.80±0.16**	**2.84±0.16**	**2.86±0.16**

APPENDIX O

COMPARISON OF RESULTS FOR ENSEMBLE METHODS, ONLINE LEARNING, SUBPOPULATION AND MODEL ADAPTATION APPROACHES

Table 17: Average-MAPE results (means and standard errors) of the proposed wFTL approaches compared to the ensemble and online methods for the different initial observation sequence lengths. The best performing method is shown in **bold**. Also in bold are the methods that are not statistically significantly different from the best method at 0.05 significance level.

Method	1	2	3	4	5	6	7	8	9
OL_FTL	3.28±0.05	3.26±0.05	3.25±0.06	3.20±0.06	3.19±0.06	3.14±0.06	3.10±0.06	3.05±0.06	3.00±0.06
OL_MW	3.36±0.06	3.36±0.06	3.35±0.06	3.31±0.06	3.29±0.06	3.24±0.05	3.19±0.05	3.18±0.06	3.15±0.05
OL_Hedge	3.43±0.06	3.43±0.07	3.42±0.07	3.37±0.07	3.38±0.08	3.28±0.05	3.23±0.05	3.20±0.05	3.20±0.05
En_Avg	3.29±0.05	3.29±0.05	3.28±0.05	3.25±0.05	3.25±0.06	3.21±0.05	3.17±0.05	3.17±0.05	3.17±0.05
En_Err	**3.16±0.05**	**3.13±0.05**	**3.12±0.05**	**3.08±0.06**	**3.08±0.06**	**3.03±0.05**	**2.98±0.05**	**2.97±0.05**	**2.96±0.05**
wFTL_se	3.24±0.05	3.22±0.05	3.20±0.05	3.15±0.06	**3.14±0.06**	3.09±0.06	**3.04±0.05**	**2.99±0.06**	**2.94±0.05**
wFTL_mr	3.24±0.05	3.22±0.05	3.20±0.05	3.16±0.06	**3.14±0.06**	3.09±0.05	**3.04±0.05**	**2.99±0.05**	**2.94±0.05**
Method	10	11	12	13	14	15	16	17	18
OL_FTL	2.96±0.05	**2.90±0.06**	**2.86±0.06**	**2.86±0.06**	**2.88±0.07**	**2.87±0.07**	**2.86±0.07**	**2.84±0.08**	**2.82±0.08**
OL_MW	3.10±0.05	3.09±0.06	3.06±0.06	3.08±0.06	3.09±0.07	3.08±0.07	3.06±0.07	3.04±0.07	3.04±0.08
OL_Hedge	3.18±0.06	3.15±0.06	3.13±0.06	3.14±0.07	3.18±0.07	3.18±0.08	3.17±0.08	3.14±0.08	3.12±0.08
En_Avg	3.16±0.05	3.15±0.06	3.13±0.06	3.16±0.06	3.20±0.07	3.22±0.07	3.22±0.08	3.21±0.08	3.22±0.08
En_Err	**2.94±0.05**	**2.91±0.05**	2.89±0.06	2.91±0.06	2.93±0.06	2.94±0.07	2.93±0.07	2.90±0.07	2.90±0.08
wFTL_se	**2.92±0.06**	**2.88±0.06**	**2.83±0.06**	**2.82±0.06**	**2.83±0.07**	**2.82±0.07**	**2.80±0.07**	**2.75±0.08**	**2.74±0.08**
wFTL_mr	**2.90±0.05**	**2.85±0.05**	**2.82±0.06**	**2.82±0.06**	**2.83±0.06**	**2.82±0.07**	**2.82±0.07**	**2.78±0.07**	**2.75±0.08**
Method	19	20	21	22	23	24	25	26	27
OL_FTL	**2.83±0.09**	2.84±0.09	2.84±0.09	**2.86±0.10**	**2.84±0.10**	**2.83±0.10**	**2.79±0.10**	**2.80±0.11**	2.84±0.11
OL_MW	3.04±0.08	3.03±0.09	3.00±0.09	3.04±0.10	3.05±0.10	3.01±0.09	2.99±0.10	3.04±0.10	3.07±0.10
OL_Hedge	3.10±0.09	3.08±0.09	3.07±0.10	3.13±0.10	3.14±0.11	3.11±0.11	3.09±0.11	3.08±0.11	3.11±0.12
En_Avg	3.22±0.09	3.21±0.09	3.22±0.09	3.27±0.10	3.30±0.10	3.29±0.10	3.30±0.10	3.36±0.11	3.41±0.11
En_Err	2.90±0.08	2.90±0.08	2.90±0.09	2.95±0.09	2.97±0.10	2.96±0.10	2.96±0.10	3.00±0.10	3.05±0.11
wFTL_se	**2.73±0.08**	**2.71±0.09**	**2.71±0.09**	**2.74±0.10**	**2.73±0.10**	**2.70±0.10**	**2.66±0.09**	**2.65±0.10**	**2.66±0.10**
wFTL_mr	**2.74±0.08**	**2.73±0.08**	**2.72±0.09**	**2.75±0.09**	**2.74±0.09**	**2.73±0.09**	**2.69±0.09**	**2.69±0.10**	**2.70±0.10**

Method	28	29	30	31	32	33	34	35	36
OL_FTL	2.82±0.12	2.85±0.12	2.87±0.13	2.91±0.14	2.90±0.14	2.98±0.14	3.04±0.15	3.03±0.15	**3.03±0.16**
OL_MW	3.07±0.11	3.10±0.11	3.11±0.12	3.12±0.12	3.13±0.13	3.15±0.14	3.17±0.14	3.16±0.15	3.17±0.15
OL_Hedge	3.10±0.12	3.15±0.12	3.16±0.13	3.15±0.13	3.16±0.14	3.18±0.14	3.23±0.15	3.19±0.15	3.18±0.15
En_Avg	3.42±0.12	3.48±0.12	3.50±0.13	3.54±0.13	3.55±0.14	3.58±0.15	3.62±0.16	3.63±0.16	3.61±0.16
En_Err	3.05±0.11	3.10±0.11	3.12±0.12	3.15±0.12	3.16±0.13	3.20±0.14	3.25±0.14	3.25±0.14	3.24±0.15
wFTL_se	**2.62±0.11**	**2.64±0.11**	**2.66±0.12**	**2.67±0.12**	**2.70±0.13**	**2.74±0.14**	**2.79±0.14**	**2.79±0.15**	**2.81±0.15**
wFTL_mr	**2.65±0.10**	**2.68±0.11**	**2.71±0.12**	**2.73±0.12**	**2.76±0.13**	**2.81±0.14**	**2.86±0.14**	**2.85±0.15**	**2.87±0.15**

Method	37	38	39	40	41	42	43	44	45
OL_FTL	**3.05±0.17**	**3.07±0.17**	**3.10±0.18**	**3.12±0.18**	**3.09±0.19**	**3.14±0.19**	**3.17±0.19**	**3.19±0.20**	**3.20±0.20**
OL_MW	3.17±0.15	3.18±0.16	3.23±0.16	3.25±0.16	3.23±0.17	3.27±0.17	3.30±0.18	3.32±0.18	3.32±0.18
OL_Hedge	3.17±0.16	3.17±0.16	3.23±0.16	3.26±0.16	3.25±0.17	3.28±0.17	3.33±0.18	3.35±0.18	3.35±0.18
En_Avg	3.63±0.16	3.63±0.17	3.68±0.17	3.71±0.18	3.68±0.18	3.73±0.18	3.77±0.18	3.78±0.19	3.78±0.19
En_Err	3.26±0.15	3.26±0.15	3.31±0.16	3.34±0.16	3.31±0.16	3.36±0.16	3.40±0.17	3.40±0.17	3.41±0.17
wFTL_se	**2.83±0.15**	**2.86±0.16**	**2.90±0.16**	**2.93±0.17**	**2.92±0.17**	**2.94±0.18**	**3.01±0.18**	**3.03±0.19**	**3.03±0.19**
wFTL_mr	**2.88±0.15**	**2.91±0.16**	**2.94±0.16**	**2.96±0.17**	**2.95±0.17**	**2.98±0.18**	**3.04±0.18**	**3.07±0.19**	**3.09±0.19**

Method	46	47	48	49	50	51	52	53	54
OL_FTL	**3.18±0.21**	**3.23±0.21**	**3.28±0.21**	**3.33±0.22**	**3.33±0.23**	**3.26±0.22**	**3.30±0.23**	**3.29±0.24**	**3.20±0.21**
OL_MW	3.31±0.19	3.33±0.19	3.33±0.20	**3.33±0.21**	**3.30±0.21**	**3.21±0.20**	**3.23±0.21**	**3.23±0.22**	**3.12±0.19**
OL_Hedge	3.33±0.19	3.34±0.20	3.32±0.20	**3.31±0.21**	**3.30±0.21**	**3.20±0.21**	**3.19±0.21**	**3.19±0.22**	**3.09±0.20**
En_Avg	3.78±0.19	3.78±0.20	3.78±0.21	3.77±0.21	3.72±0.22	3.64±0.22	3.66±0.23	3.69±0.23	3.61±0.21
En_Err	3.40±0.18	3.40±0.18	3.40±0.19	3.39±0.20	3.34±0.20	3.26±0.19	3.28±0.20	3.29±0.21	**3.22±0.19**
wFTL_se	**2.98±0.19**	**2.98±0.19**	**3.04±0.20**	**3.08±0.20**	**3.08±0.21**	**2.99±0.19**	**3.01±0.20**	**3.00±0.21**	**2.98±0.21**
wFTL_mr	**3.05±0.20**	3.09±0.20	**3.15±0.20**	**3.19±0.21**	**3.17±0.21**	**3.07±0.20**	**3.08±0.21**	**3.06±0.21**	**3.05±0.22**

Method	55	56	57	58	59	60
OL_FTL	**3.19±0.21**	3.28±0.21	3.19±0.20	3.20±0.20	3.20±0.21	3.20±0.21
OL_MW	**3.04±0.18**	**3.08±0.17**	**3.02±0.17**	**3.05±0.17**	**3.11±0.17**	**3.09±0.17**
OL_Hedge	**3.00±0.19**	**3.01±0.20**	**2.90±0.17**	**2.92±0.18**	**2.95±0.18**	**2.90±0.18**
En_Avg	3.51±0.20	3.49±0.18	3.47±0.19	3.51±0.19	3.54±0.19	3.51±0.20
En_Err	**3.13±0.18**	**3.13±0.17**	3.07±0.16	3.10±0.16	3.13±0.17	**3.11±0.17**
wFTL_se	**2.87±0.19**	**2.86±0.18**	**2.76±0.16**	**2.75±0.16**	**2.78±0.17**	**2.81±0.17**
wFTL_mr	**2.97±0.21**	**2.96±0.20**	**2.83±0.16**	**2.80±0.16**	**2.84±0.16**	**2.86±0.16**

Table 18: Average-MAPE results (means and standard errors) of the proposed wFTL approaches compared to the subpopulation methods for the different initial observation sequence lengths. The best performing method is shown in **bold**. Also in bold are the methods that are not statistically significantly different from the best method at 0.05 significance level.

Method	1	2	3	4	5	6	7	8	9
Sub-50	3.04±0.04	3.02±0.05	2.98±0.05	2.94±0.05	2.94±0.05	2.93±0.05	2.91±0.05	2.89±0.05	2.88±0.05
Sub-100	3.00±0.04	2.98±0.04	2.95±0.05	2.91±0.05	2.90±0.05	2.90±0.05	**2.87±0.05**	**2.85±0.05**	**2.84±0.05**
Sub-200	3.00±0.04	2.99±0.04	2.95±0.05	2.91±0.05	2.91±0.05	2.90±0.05	2.88±0.05	2.86±0.05	2.84±0.05
Sub-ALL	**2.96±0.04**	**2.95±0.04**	**2.92±0.04**	**2.88±0.04**	**2.88±0.05**	**2.88±0.05**	**2.85±0.05**	**2.84±0.05**	**2.82±0.05**
wFTL_se	3.24±0.05	3.22±0.05	3.20±0.05	3.15±0.06	3.14±0.06	3.09±0.06	3.04±0.05	2.99±0.06	2.94±0.05
wFTL_mr	3.24±0.05	3.22±0.05	3.20±0.05	3.16±0.06	3.14±0.06	3.09±0.05	3.04±0.05	2.99±0.05	2.94±0.05

Method	10	11	12	13	14	15	16	17	18
Sub-50	2.89±0.05	2.86±0.06	2.86±0.06	2.87±0.06	2.90±0.07	**2.90±0.07**	2.90±0.07	2.86±0.07	2.86±0.08
Sub-100	**2.83±0.05**	**2.81±0.06**	**2.80±0.06**	**2.82±0.06**	**2.83±0.07**	**2.82±0.07**	**2.82±0.07**	**2.77±0.07**	**2.76±0.08**
Sub-200	2.84±0.05	2.82±0.06	2.82±0.06	2.83±0.06	**2.85±0.07**	**2.85±0.07**	**2.84±0.07**	**2.79±0.07**	**2.79±0.08**
Sub-ALL	**2.83±0.05**	**2.81±0.06**	**2.80±0.06**	**2.81±0.06**	**2.83±0.07**	**2.83±0.07**	**2.83±0.07**	**2.79±0.07**	**2.78±0.08**
wFTL_se	2.92±0.06	2.88±0.06	**2.83±0.06**	**2.82±0.06**	**2.83±0.07**	**2.82±0.07**	**2.80±0.07**	**2.75±0.08**	**2.74±0.08**
wFTL_mr	2.90±0.05	**2.85±0.05**	**2.82±0.06**	**2.82±0.06**	**2.83±0.06**	**2.82±0.07**	**2.82±0.07**	**2.78±0.07**	**2.75±0.08**

Method	19	20	21	22	23	24	25	26	27
Sub-50	2.89±0.08	2.88±0.08	2.90±0.09	2.94±0.09	2.97±0.10	2.97±0.10	2.96±0.10	3.00±0.10	3.06±0.10
Sub-100	**2.79±0.08**	**2.78±0.08**	**2.79±0.09**	**2.83±0.09**	**2.84±0.10**	2.84±0.10	2.82±0.09	2.86±0.10	2.92±0.10
Sub-200	**2.82±0.08**	**2.82±0.08**	**2.83±0.09**	2.87±0.09	2.89±0.10	2.89±0.10	2.88±0.10	2.93±0.10	2.99±0.10
Sub-ALL	**2.81±0.08**	**2.81±0.08**	**2.82±0.09**	**2.86±0.09**	2.89±0.10	2.88±0.10	2.87±0.10	2.92±0.10	2.98±0.10
wFTL_se	**2.73±0.08**	**2.71±0.09**	**2.71±0.09**	**2.74±0.10**	**2.73±0.10**	**2.70±0.10**	**2.66±0.09**	**2.65±0.10**	**2.66±0.10**
wFTL_mr	**2.74±0.08**	**2.73±0.08**	**2.72±0.09**	**2.75±0.09**	**2.74±0.09**	**2.73±0.09**	**2.69±0.09**	**2.69±0.10**	**2.70±0.10**

Method	28	29	30	31	32	33	34	35	36
Sub-50	3.06±0.11	3.09±0.11	3.12±0.11	3.10±0.12	3.15±0.12	3.20±0.13	3.25±0.13	3.26±0.14	3.27±0.14
Sub-100	2.91±0.11	2.94±0.11	2.97±0.11	2.95±0.12	2.98±0.12	3.04±0.13	3.08±0.14	3.08±0.14	3.08±0.14
Sub-200	2.99±0.11	3.02±0.11	3.06±0.12	3.06±0.12	3.08±0.13	3.14±0.13	3.20±0.14	3.19±0.14	3.18±0.14
Sub-ALL	2.98±0.11	3.02±0.11	3.07±0.12	3.06±0.12	3.09±0.13	3.15±0.14	3.21±0.14	3.21±0.15	3.20±0.15
wFTL_se	**2.62±0.11**	**2.64±0.11**	**2.66±0.12**	**2.67±0.12**	**2.70±0.13**	**2.74±0.14**	**2.79±0.14**	**2.79±0.15**	**2.81±0.15**
wFTL_mr	**2.65±0.10**	**2.68±0.11**	**2.71±0.12**	**2.73±0.12**	**2.76±0.13**	**2.81±0.14**	**2.86±0.14**	**2.85±0.15**	**2.87±0.15**

Method	37	38	39	40	41	42	43	44	45
Sub-50	3.29±0.14	3.31±0.15	3.35±0.15	3.39±0.15	3.39±0.16	3.43±0.16	3.47±0.17	3.50±0.17	3.53±0.17
Sub-100	3.10±0.14	3.11±0.15	3.15±0.15	3.19±0.15	3.18±0.16	3.23±0.16	3.27±0.17	3.29±0.17	3.32±0.18
Sub-200	3.21±0.15	3.23±0.15	3.26±0.16	3.29±0.16	3.29±0.17	3.34±0.17	3.38±0.17	3.40±0.18	3.44±0.18
Sub-ALL	3.24±0.15	3.26±0.15	3.30±0.16	3.33±0.16	3.33±0.17	3.39±0.17	3.43±0.17	3.45±0.18	3.48±0.18
wFTL_se	**2.83±0.15**	**2.86±0.16**	**2.90±0.16**	**2.93±0.17**	**2.92±0.17**	**2.94±0.18**	**3.01±0.18**	**3.03±0.19**	**3.03±0.19**
wFTL_mr	**2.88±0.15**	**2.91±0.16**	**2.94±0.16**	**2.96±0.17**	**2.95±0.17**	**2.98±0.18**	**3.04±0.18**	**3.07±0.19**	**3.09±0.19**

Method	46	47	48	49	50	51	52	53	54
Sub-50	3.52±0.18	3.51±0.18	3.54±0.18	3.55±0.19	3.52±0.20	3.44±0.19	3.40±0.20	3.42±0.20	3.38±0.21
Sub-100	3.31±0.18	3.30±0.18	3.31±0.19	**3.32±0.20**	**3.30±0.20**	**3.22±0.20**	**3.18±0.20**	**3.19±0.21**	**3.16±0.21**
Sub-200	3.43±0.18	3.41±0.18	3.42±0.19	3.41±0.20	3.39±0.20	3.32±0.20	**3.28±0.20**	3.31±0.21	**3.28±0.21**
Sub-ALL	3.47±0.18	3.45±0.18	3.46±0.19	3.47±0.20	3.44±0.20	3.37±0.20	3.33±0.20	3.36±0.21	3.33±0.21
wFTL_se	**2.98±0.19**	**2.98±0.19**	**3.04±0.20**	**3.08±0.20**	**3.08±0.21**	**2.99±0.19**	**3.01±0.20**	**3.00±0.21**	**2.98±0.21**
wFTL_mr	**3.05±0.20**	3.09±0.20	**3.15±0.20**	**3.19±0.21**	**3.17±0.21**	**3.07±0.20**	**3.08±0.21**	**3.06±0.21**	**3.05±0.22**

Method	55	56	57	58	59	60
Sub-50	3.27±0.19	3.31±0.18	3.24±0.16	3.21±0.17	3.23±0.17	3.25±0.17
Sub-100	**3.07±0.19**	**3.11±0.19**	**3.03±0.16**	**3.00±0.16**	**3.02±0.17**	**3.02±0.17**
Sub-200	3.19±0.19	3.23±0.18	3.17±0.17	3.15±0.17	3.16±0.17	3.17±0.18
Sub-ALL	3.24±0.19	3.29±0.18	3.23±0.17	3.21±0.17	3.22±0.18	3.22±0.18
wFTL_se	**2.87±0.19**	**2.86±0.18**	**2.76±0.16**	**2.75±0.16**	**2.78±0.17**	**2.81±0.17**
wFTL_mr	**2.97±0.21**	**2.96±0.20**	**2.83±0.16**	**2.80±0.16**	**2.84±0.16**	**2.86±0.16**

Table 19: Average-MAPE results (means and standard errors) of the proposed wFTL approaches compared to the model adaptation based methods for the different initial observation sequence lengths. reGP and reMTGP are the abbreviations for rLDS+reGP and rLDS+reMTGP. The best performing method is shown in **bold**. Also in bold are the methods that are not statistically significantly different from the best method at 0.05 significance level.

Method	1	2	3	4	5	6	7	8	9
reGP	3.64±0.06	3.33±0.05	3.22±0.05	3.13±0.05	**3.08±0.06**	**3.03±0.06**	**2.96±0.06**	**2.89±0.06**	**2.84±0.06**
reMTGP	3.60±0.06	3.29±0.05	**3.19±0.05**	**3.10±0.05**	**3.05±0.05**	**3.00±0.06**	**2.94±0.06**	**2.88±0.06**	**2.84±0.06**
wFTL_se	**3.24±0.05**	**3.22±0.05**	**3.20±0.05**	**3.15±0.06**	3.14±0.06	3.09±0.06	3.04±0.05	2.99±0.06	2.94±0.05
wFTL_mr	**3.24±0.05**	**3.22±0.05**	**3.20±0.05**	**3.16±0.06**	3.14±0.06	3.09±0.05	3.04±0.05	2.99±0.05	2.94±0.05
Method	10	11	12	13	14	15	16	17	18
reGP	**2.81±0.06**	**2.76±0.06**	**2.73±0.06**	**2.71±0.06**	**2.70±0.07**	**2.67±0.07**	**2.65±0.07**	**2.60±0.08**	**2.58±0.08**
reMTGP	**2.81±0.06**	**2.76±0.06**	**2.74±0.06**	**2.72±0.06**	**2.72±0.07**	**2.70±0.07**	**2.67±0.07**	**2.63±0.07**	**2.62±0.08**
wFTL_se	2.92±0.06	2.88±0.06	2.83±0.06	2.82±0.06	2.83±0.07	2.82±0.07	2.80±0.07	2.75±0.08	2.74±0.08
wFTL_mr	2.90±0.05	2.85±0.05	2.82±0.06	2.82±0.06	2.83±0.06	2.82±0.07	2.82±0.07	2.78±0.07	2.75±0.08
Method	19	20	21	22	23	24	25	26	27
reGP	**2.60±0.08**	**2.57±0.09**	**2.53±0.09**	**2.53±0.09**	**2.53±0.10**	**2.51±0.09**	**2.44±0.09**	**2.46±0.09**	**2.48±0.10**
reMTGP	**2.63±0.08**	**2.60±0.08**	2.59±0.09	2.61±0.09	2.60±0.10	2.58±0.10	2.53±0.10	2.54±0.10	2.58±0.11
wFTL_se	2.73±0.08	2.71±0.09	2.71±0.09	2.74±0.10	2.73±0.10	2.70±0.10	2.66±0.09	2.65±0.10	2.66±0.10
wFTL_mr	2.74±0.08	2.73±0.08	2.72±0.09	2.75±0.09	2.74±0.09	2.73±0.09	2.69±0.09	2.69±0.10	2.70±0.10
Method	28	29	30	31	32	33	34	35	36
reGP	**2.46±0.10**	**2.49±0.11**	**2.52±0.11**	**2.51±0.12**	**2.54±0.12**	**2.59±0.13**	**2.64±0.14**	**2.62±0.14**	**2.63±0.14**
reMTGP	2.57±0.11	2.59±0.12	2.61±0.12	**2.60±0.13**	**2.62±0.13**	**2.66±0.14**	**2.71±0.15**	**2.70±0.15**	**2.71±0.16**
wFTL_se	2.62±0.11	**2.64±0.11**	**2.66±0.12**	**2.67±0.12**	**2.70±0.13**	**2.74±0.14**	**2.79±0.14**	**2.79±0.15**	**2.81±0.15**
wFTL_mr	2.65±0.10	2.68±0.11	2.71±0.12	2.73±0.12	2.76±0.13	2.81±0.14	2.86±0.14	2.85±0.15	2.87±0.15
Method	37	38	39	40	41	42	43	44	45
reGP	**2.64±0.14**	**2.67±0.15**	**2.68±0.15**	**2.71±0.16**	**2.72±0.16**	**2.77±0.17**	**2.81±0.17**	**2.84±0.17**	**2.88±0.18**
reMTGP	**2.73±0.16**	**2.75±0.17**	**2.79±0.17**	**2.82±0.18**	**2.83±0.18**	**2.89±0.19**	**2.93±0.19**	**2.97±0.20**	**3.01±0.20**
wFTL_se	**2.83±0.15**	**2.86±0.16**	**2.90±0.16**	**2.93±0.17**	**2.92±0.17**	**2.94±0.18**	**3.01±0.18**	**3.03±0.19**	**3.03±0.19**
wFTL_mr	2.88±0.15	2.91±0.16	2.94±0.16	2.96±0.17	2.95±0.17	**2.98±0.18**	**3.04±0.18**	**3.07±0.19**	**3.09±0.19**
Method	46	47	48	49	50	51	52	53	54
reGP	**2.86±0.18**	**2.87±0.19**	**2.91±0.19**	**2.94±0.20**	**2.97±0.21**	**2.91±0.20**	**2.92±0.20**	**2.89±0.20**	**2.83±0.21**
reMTGP	**2.99±0.21**	**2.96±0.20**	**2.96±0.21**	**2.97±0.22**	**2.98±0.23**	**2.94±0.22**	**2.96±0.23**	**2.93±0.23**	**2.79±0.20**
wFTL_se	**2.98±0.19**	**2.98±0.19**	**3.04±0.20**	**3.08±0.20**	**3.08±0.21**	**2.99±0.19**	**3.01±0.20**	**3.00±0.21**	**2.98±0.21**
wFTL_mr	**3.05±0.20**	**3.09±0.20**	**3.15±0.20**	**3.19±0.21**	**3.17±0.21**	**3.07±0.20**	**3.08±0.21**	**3.06±0.21**	**3.05±0.22**
Method	55	56	57	58	59	60			
reGP	**2.77±0.20**	**2.83±0.21**	**2.71±0.16**	**2.70±0.17**	**2.72±0.17**	**2.75±0.18**			
reMTGP	**2.76±0.21**	**2.81±0.22**	**2.68±0.17**	**2.65±0.18**	**2.67±0.18**	**2.72±0.18**			
wFTL_se	**2.87±0.19**	**2.86±0.18**	**2.76±0.16**	**2.75±0.16**	**2.78±0.17**	**2.81±0.17**			
wFTL_mr	**2.97±0.21**	**2.96±0.20**	**2.83±0.16**	**2.80±0.16**	**2.84±0.16**	**2.86±0.16**			

BIBLIOGRAPHY

[Aburto and Weber, 2007] Aburto, L. and Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, 7(1):136–144.

[Adorf, 1995] Adorf, H.-M. (1995). Interpolation of irregularly sampled data series – a survey. In *Astronomical Data Analysis Software and Systems IV*, volume 77, pages 460–463.

[Angelosante et al., 2009] Angelosante, D., Roumeliotis, S., and Giannakis, G. (2009). Lasso-Kalman smoother for tracking sparse signals. In *Asilomar Conference on Signals, Systems and Computers*, pages 181–185, Pacific Grove, CA, USA.

[Åström, 1969] Åström, K. J. (1969). On the choice of sampling rates in parametric identification of time series. *Information Sciences*, 1(3):273–278.

[Bach et al., 2011] Bach, F., Jenatton, R., Mairal, J., Obozinski, G., et al. (2011). Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53.

[Bahadori and Liu, 2013] Bahadori, M. T. and Liu, Y. (2013). An examination of practical Granger causality inference. In *SIAM Conference on Data Mining*, pages 467–475, Austin, TX, USA.

[Batal et al., 2012] Batal, I., Fradkin, D., Harrison, J., Moerchen, F., and Hauskrecht, M. (2012). Mining recent temporal patterns for event detection in multivariate time series data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 280–288, Beijing, China. ACM.

[Batal et al., 2011] Batal, I., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2011). A pattern mining approach for classifying multivariate temporal data. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 358–365, Atlanta, GA, USA. IEEE.

[Bellazzi et al., 2000] Bellazzi, R., Larizza, C., Magni, P., Montani, S., and Stefanelli, M. (2000). Intelligent analysis of clinical time series: an application in the diabetes mellitus domain. *Artificial Intelligence in Medicine*, 20(1):37–57.

[Bellazzi et al., 1995] Bellazzi, R., Siviero, C., Stefanelli, M., and De Nicolao, G. (1995). Adaptive controllers for intelligent monitoring. *Artificial Intelligence in Medicine*, 7(6):515–540.

[Berliner, 1996] Berliner, L. M. (1996). Hierarchical Bayesian time series models. In *Maximum entropy and Bayesian methods*, pages 15–22. Springer.

[Berndt and Clifford, 1994] Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA.

[Berzuini et al., 1992] Berzuini, C., Bellazzi, R., Quaglini, S., and Spiegelhalter, D. J. (1992). Bayesian networks for patient monitoring. *Artificial Intelligence in Medicine*, 4(3):243–260.

[Berzuini et al., 1991] Berzuini, C., Bellazzi, R., and Spiegelhalter, D. (1991). Bayesian networks applied to therapy monitoring. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 35–43, Los Angeles, CA, USA.

[Blum, 1998] Blum, A. (1998). *On-line Algorithms in Machine Learning*. Springer.

[Bonilla et al., 2007] Bonilla, E. V., Chai, K. M., and Williams, C. (2007). Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160, Whistler, B.C., Canada.

[Boots et al., 2007] Boots, B., Gordon, G. J., and Siddiqi, S. M. (2007). A constraint generation approach to learning stable linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 1329–1336, Vancouver, B.C., Canada.

[Box and Pierce, 1970] Box, G. E. and Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526.

[Brahim-Belhouari and Bermak, 2004] Brahim-Belhouari, S. and Bermak, A. (2004). Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, 47(4):705–712.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[Carmi et al., 2010] Carmi, A., Gurfil, P., and Kanevsky, D. (2010). Methods for sparse signal recovery using Kalman filtering with embedded pseudo-measurement norms and quasi-norms. *IEEE Transactions on Signal Processing*, 58(4):2405–2409.

[Caruana et al., 2004] Caruana, R., Niculescu-Mizil, A., Crew, G., and Ksikes, A. (2004). Ensemble selection from libraries of models. In *Proceedings of The Twenty-First International Conference on Machine Learning*, pages 18–25, Alberta, Canada. ACM.

[Cesa-Bianchi et al., 2007] Cesa-Bianchi, N., Mansour, Y., and Stoltz, G. (2007). Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352.

[Charles et al., 2011] Charles, A., Asif, M. S., Romberg, J., and Rozell, C. (2011). Sparsity penalties in dynamical system estimation. In *The 45th Annual Conference on Information Sciences and Systems*, pages 1–6, Baltimore, MD, USA. IEEE.

[Cheng et al., 2014] Cheng, D., Bahadori, M. T., and Liu, Y. (2014). FBLG: a simple and effective approach for temporal dependence discovery from time series data. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 382–391, New York, NY, USA. ACM.

[Chu, 1995] Chu, C.-S. J. (1995). Time series segmentation: a sliding window approach. *Information Sciences*, 85(1):147–173.

[Clifton et al., 2013] Clifton, L., Clifton, D. A., Pimentel, M., Watkinson, P. J., Tarassenko, L., et al. (2013). Gaussian processes for personalized e-health monitoring with wearable sensors. *IEEE Transactions on Biomedical Engineering*, 60(1):193–197.

[Cohen, 1995] Cohen, L. (1995). *Time-frequency Analysis*, volume 1406. Prentice Hall PTR Englewood Cliffs, NJ:.

[Crammer et al., 2006] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

[Das et al., 1998] Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. (1998). Rule discovery from time series. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, volume 98, pages 16–22, New York, NY, USA.

[Deisenroth et al., 2009] Deisenroth, M. P., Huber, M. F., and Hanebeck, U. D. (2009). Analytic moment-based Gaussian process filtering. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 225–232, Montreal, Canada. ACM.

[Dezhbakhsh and Levy, 1994] Dezhbakhsh, H. and Levy, D. (1994). Periodic properties of interpolated time series. *Economics Letters*, 44(3):221–228.

[Doretto et al., 2003] Doretto, G., Chiuso, A., Wu, Y. N., and Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109.

[Du Preez and Witt, 2003] Du Preez, J. and Witt, S. F. (2003). Univariate versus multivariate time series forecasting: an application to international tourism demand. *International Journal of Forecasting*, 19(3):435–451.

[Durichen et al., 2015] Durichen, R., Pimentel, M., Clifton, L., Schweikard, A., Clifton, D. A., et al. (2015). Multitask Gaussian processes for multivariate physiological time-series analysis. *IEEE Transactions on Biomedical Engineering*, 62(1):314–322.

[Felsenstein, 1985] Felsenstein, J. (1985). Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, 39(4):783–791.

[Fornasier and Rauhut, 2008] Fornasier, M. and Rauhut, H. (2008). Iterative thresholding algorithms. *Applied and Computational Harmonic Analysis*, 25(2):187–208.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

[Funk, 2003] Funk, N. (2003). A study of the Kalman filter applied to visual tracking. *University of Alberta, Project for CMPUT*, 652:6.

[Gardner, 1985] Gardner, E. S. (1985). Exponential smoothing: the state of the art. *Journal of Forecasting*, 4(1):1–28.

[Georgiadis et al., 2005] Georgiadis, S. D., Ranta-aho, P. O., Tarvainen, M. P., and Karjalainen, P. A. (2005). Single-trial dynamical estimation of event-related potentials: a Kalman filter-based approach. *IEEE Transactions on Biomedical Engineering*, 52(8):1397–1406.

[Georgiadis et al., 2007] Georgiadis, S. D., Ranta-aho, P. O., Tarvainen, M. P., and Karjalainen, P. A. (2007). A subspace method for dynamical estimation of evoked potentials. *Computational Intelligence and Neuroscience*, 2007:12.

[Ghahramani and Hinton, 1996] Ghahramani, Z. and Hinton, G. (1996). Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Totronto, Toronto, Canada.

[Ghahramani and Jordan, 1997] Ghahramani, Z. and Jordan, M. I. (1997). Factorial hidden Markov models. *Machine Learning*, 29(2-3):245–273.

[Ghassemi et al., 2015] Ghassemi, M., Pimentel, M. A., Naumann, T., Brennan, T., Clifton, D. A., Szolovits, P., and Feng, M. (2015). A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data. In *The Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 446–453, Austin, Texas, USA.

[Girard et al., 2003] Girard, A., Rasmussen, C. E., Candela, J. Q., and Murray-Smith, R. (2003). Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting. In *Proceedings of Advances in Neural Information Processing Systems*, pages 545–552, Vancouver, Whistler, Canada.

[Gneiting and Raftery, 2005] Gneiting, T. and Raftery, A. E. (2005). Weather forecasting with ensemble methods. *Science*, 310(5746):248–249.

[Gottrup et al., 2005] Gottrup, C., Thomsen, K., Locht, P., Wu, O., Sorensen, A. G., Koroshetz, W. J., and Østergaard, L. (2005). Applying instance-based techniques to prediction of final outcome in acute stroke. *Artificial Intelligence in Medicine*, 33(3):223–236.

[Halley, 1694] Halley, E. (1694). An account of the evaporation of water, as it was experimented in Gresham Colledge in the year 1693. with some observations thereon. by edm. halley. *Philosophical Transactions*, 18(207-214):183–190.

[Hamilton, 1994] Hamilton, J. D. (1994). *Time Series Analysis*, volume 2. Princeton University Press.

[Hauskrecht et al., 2013] Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G. F., and Clermont, G. (2013). Outlier detection for patient monitoring and alerting. *Journal of Biomedical Informatics*, 46(1):47–55.

[Hauskrecht et al., 2010a] Hauskrecht, M., Valko, M., Batal, I., Clermont, G., Visweswaran, S., and Cooper, G. (2010a). Conditional outlier detection for clinical alerting. In *AMIA annual symposium proceedings*, pages 286–290, Washington DC, USA.

[Hauskrecht et al., 2010b] Hauskrecht, M., Valko, M., Batal, I., Clermont, G., Visweswaran, S., and Cooper, G. (2010b). Conditional outlier detection for clinical alerting. In *AMIA Annual Symposium Proceedings*, pages 286–290, Washington DC, USA.

[Ho et al., 2003] Ho, T. B., Nguyen, T. D., Kawasaki, S., Le, S. Q., Nguyen, D. D., Yokoi, H., and Takabayashi, K. (2003). Mining hepatitis data with temporal abstraction. In *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 369–377, Washington, DC, USA. ACM.

[Hoerl, 1962] Hoerl, A. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58(3):54–59.

[Hou and Zhang, 2007] Hou, X. and Zhang, L. (2007). Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

[Huang et al., 2014] Huang, Z., Dong, W., Duan, H., and Li, H. (2014). Similarity measure between patient traces for clinical pathway analysis: problem, method, and applications. *IEEE Journal of Biomedical and Health Informatics*, 18(1):4–14.

[Huang et al., 2015] Huang, Z., Dong, W., Wang, F., and Duan, H. (2015). Medical inpatient journey modeling and clustering: A Bayesian hidden Markov model based approach. In *AMIA Annual Symposium Proceedings*, volume 2015, page 649. American Medical Informatics Association.

[Huang et al., 2013] Huang, Z., Juarez, J. M., Duan, H., and Li, H. (2013). Length of stay prediction for clinical treatment process using temporal similarity. *Expert Systems with Applications*, 40(16):6330–6339.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). Condensationconditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.

[Jiang et al., 2012] Jiang, X., Boxwala, A. A., El-Kareh, R., Kim, J., and Ohno-Machado, L. (2012). A patient-driven adaptive prediction technique to improve personalized risk estimation for clinical decision support. *Journal of the American Medical Informatics Association*, 19(e1):e137–e144.

[Jørgensen, 2009] Jørgensen, J. T. (2009). New era of personalized medicine: a 10-year anniversary. *The Oncologist*, 14(5):557–558.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45.

[Kalman, 1963] Kalman, R. E. (1963). Mathematical description of linear dynamical systems. *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, 1(2):152–192.

[Karkar et al., 2015] Karkar, R., Zia, J., Vilardaga, R., Mishra, S. R., Fogarty, J., Munson, S. A., and Kientz, J. A. (2015). A framework for self-experimentation in personalized health. *Journal of the American Medical Informatics Association*, page ocv150.

[Katayama, 2005] Katayama, T. (2005). *Subspace Methods for System Identification*. Springer, NY, USA.

[Kazemi et al., 2008] Kazemi, R., Farsi, A., Ghaed, M., and Karimi-Ghartemani, M. (2008). Detection and extraction of periodic noises in audio and biomedical signals using Kalman filter. *Signal Processing*, 88(8):2114–2121.

[Keogh et al., 2001] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286.

[Keogh and Pazzani, 2000] Keogh, E. J. and Pazzani, M. J. (2000). A simple dimensionality reduction technique for fast similarity search in large time series databases. In *Knowledge Discovery and Data Mining. Current Issues and New Applications*, pages 122–133. Springer.

[Khan and Dutt, 2007] Khan, M. E. and Dutt, D. N. (2007). An expectation-maximization algorithm based Kalman smoother approach for event-related desynchronization (ERD) estimation from EEG. *IEEE Transactions on Biomedical Engineering*, 54(7):1191–1198.

[Kim, 1994] Kim, C.-J. (1994). Dynamic linear models with Markov-switching. *Journal of Econometrics*, 60(1):1–22.

[Kling and Bessler, 1985] Kling, J. L. and Bessler, D. A. (1985). A comparison of multivariate forecasting procedures for economic time series. *International Journal of Forecasting*, 1(1):5–24.

[Ko and Fox, 2011] Ko, J. and Fox, D. (2011). Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23.

[Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8):30–37.

[Kreindler and Lumsden, 2006] Kreindler, D. and Lumsden, C. (2006). The effects of the irregular sample and missing data in time series analysis. *Nonlinear Dynamics, Psychology, and Life Sciences*, 10(2):187–214.

[Krishnan and Kushwaha, 1973] Krishnan, A. and Kushwaha, R. S. (1973). A multiple regression analysis of evaporation during the growing season of vegetation in the arid zone of india. *Agricultural Meteorology*, 12:297–307.

[Lasko et al., 2013] Lasko, T. A., Denny, J. C., and Levy, M. A. (2013). Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS One*, 8(6):e66341.

[Lee and Seung, 1999] Lee, D. and Seung, S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

[Lee et al., 1995] Lee, J. W., Kim, M. S., and Kweon, I. S. (1995). A Kalman filter based visual tracking algorithm for an object moving in 3D. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 342–347, Pittsburgh, USA. IEEE.

[Li et al., 2009] Li, L., McCann, J., Pollard, N. S., and Faloutsos, C. (2009). Dynammo: Mining and summarization of coevolving sequences with missing values. In *Proceedings of The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 507–516, Paris, France. ACM.

[Li et al., 2004] Li, P., Zhang, T., and Ma, B. (2004). Unscented Kalman filter for visual curve tracking. *Image and Vision Computing*, 22(2):157–164.

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

[Littlestone and Warmuth, 1994] Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, 108(2):212–261.

[Liu et al., 2009] Liu, J., Ji, S., and Ye, J. (2009). Multi-task feature learning via efficient $l_{2,1}$-norm minimization. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 339–348, Montreal, Canada.

[Liu and Ye, 2009] Liu, J. and Ye, J. (2009). Efficient Euclidean projections in linear time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 657–664, Montreal, Canada.

[Liu and Ye, 2010] Liu, J. and Ye, J. (2010). Moreau-Yosida regularization for grouped tree structure learning. In *Advances in Neural Information Processing Systems*, pages 1459–1467, Vancouver, B.C., Canada.

[Liu and Hauskrecht, 2013] Liu, Z. and Hauskrecht, M. (2013). Clinical time series prediction with a hierarchical dynamical system. In *Artificial Intelligence in Medicine*, pages 227–237. Murcia, Spain.

[Liu and Hauskrecht, 2015a] Liu, Z. and Hauskrecht, M. (2015a). Clinical time series prediction: Toward a hierarchical dynamical system framework. *Artificial Intelligence in Medicine*, 65(1):5–18.

[Liu and Hauskrecht, 2015b] Liu, Z. and Hauskrecht, M. (2015b). A regularized linear dynamical system framework for multivariate time series analysis. In *The Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1798–1804, Austin, Texas, USA.

[Liu and Hauskrecht, 2016a] Liu, Z. and Hauskrecht, M. (2016a). Learning adaptive forecasting models from irregularly sampled multivariate clinical data. In *The Thirtieth AAAI Conference on Artificial Intelligence*, pages 1273–1279, Phoenix, Arizona, USA.

[Liu and Hauskrecht, 2016b] Liu, Z. and Hauskrecht, M. (2016b). Learning linear dynamical systems from multivariate time series: A matrix factorization based framework. In *SIAM International Conference on Data Mining*, Miami, Florida, USA.

[Liu et al., 2013] Liu, Z., Wu, L., and Hauskrecht, M. (2013). Modeling clinical time series using Gaussian process sequences. In *SIAM International Conference on Data Mining*, pages 623–631, Austin, Texas, USA.

[Ljung and Glad, 1994] Ljung, L. and Glad, T. (1994). Modeling of dynamic systems.

[Lunze, 1994] Lunze, J. (1994). Qualitative modelling of linear dynamical systems with quantized state measurements. *Automatica*, 30(3):417–431.

[Ma et al., 2009] Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. (2009). Identifying suspicious URLs: an application of large-scale online learning. In *Proceedings of The 26th Annual International Conference on Machine Learning*, pages 681–688, Montreal, Canada. ACM.

[MacDonald and Zucchini, 1997] MacDonald, I. L. and Zucchini, W. (1997). *Hidden Markov and Other Models for Discrete-valued Time Series*, volume 110. CRC Press.

[Makridakis and Hibon, 1997] Makridakis, S. and Hibon, M. (1997). ARMA models and the Box–jenkins methodology. *Journal of Forecasting*, 16(3):147–163.

[Marlin et al., 2012] Marlin, B. M., Kale, D. C., Khemani, R. G., and Wetzel, R. C. (2012). Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 389–398, Miami, Florida, USA. ACM.

[Martens, 2010] Martens, J. (2010). Learning the linear dynamical system with ASOS. In *Proceedings of the 27th International Conference on Machine Learning*, pages 743–750.

[McCulloch and Tsay, 1994] McCulloch, R. E. and Tsay, R. S. (1994). Statistical analysis of economic time series via Markov switching models. *Journal of Time Series Analysis*, 15(5):523–539.

[Moon et al., 2007] Moon, H., Ahn, H., Kodell, R. L., Baek, S., Lin, C.-J., and Chen, J. J. (2007). Ensemble methods for classification of patients for personalized medicine with high-dimensional data. *Artificial Intelligence in Medicine*, 41(3):197–207.

[Müller, 2007] Müller, M. (2007). Dynamic time warping. *Information Retrieval for Music and Motion*, pages 69–84.

[Nguyen et al., 2014] Nguyen, Q., Valizadegan, H., and Hauskrecht, M. (2014). Learning classification models with soft-label information. *Journal of the American Medical Informatics Association*, 21(3):501–508.

[Nutt et al., 2003] Nutt, C. L., Mani, D., Betensky, R. A., Tamayo, P., Cairncross, J. G., Ladd, C., Pohl, U., Hartmann, C., McLaughlin, M. E., Batchelor, T. T., et al. (2003). Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Research*, 63(7):1602–1607.

[Osorio et al., 1998] Osorio, I., Frei, M. G., and Wilkinson, S. B. (1998). Real-time automated detection and quantitative analysis of seizures and short-term prediction of clinical onset. *Epilepsia*, 39(6):615–627.

[Qu and Gotman, 1997] Qu, H. and Gotman, J. (1997). A patient-specific algorithm for the detection of seizure onset in long-term EEG monitoring: possible use as a warning device. *IEEE Transactions on Biomedical Engineering*, 44(2):115–122.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Rasmussen and Williams, 2006] Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press Cambridge, MA, USA.

[Rehfeld et al., 2011] Rehfeld, K., Marwan, N., Heitzig, J., and Kurths, J. (2011). Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404.

[Reinsel, 2003] Reinsel, G. C. (2003). *Elements of Multivariate Time Series Analysis*. Springer.

[Richard et al., 2012] Richard, E., Savalle, P.-a., and Vayatis, N. (2012). Estimation of simultaneously sparse and low rank matrices. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1351–1358, Edinburgh, Scotland.

[Richman and Moorman, 2000] Richman, J. S. and Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049.

[Roberts et al., 2013] Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., and Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550.

[Rogers et al., 2013] Rogers, M., Li, L., and Russell, S. J. (2013). Multilinear dynamical systems for tensor time series. In *Advances in Neural Information Processing Systems*, pages 2634–2642, Lake Tahoe, Nevada, USA.

[Sacchi et al., 2007] Sacchi, L., Larizza, C., Combi, C., and Bellazzi, R. (2007). Data mining with temporal abstractions: learning rules from time series. *Data Mining and Knowledge Discovery*, 15(2):217–247.

[Sayadi and Shamsollahi, 2008] Sayadi, O. and Shamsollahi, M. B. (2008). ECG denoising and compression using a modified extended Kalman filter structure. *IEEE Transactions on Biomedical Engineering*, 55(9):2240–2248.

[Scargle, 1982] Scargle, J. D. (1982). Studies in astronomical time series analysis. ii-statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263:835–853.

[Schleidgen et al., 2013] Schleidgen, S., Klingler, C., Bertram, T., Rogowski, W. H., and Marckmann, G. (2013). What is personalized medicine: sharpening a vague term based on a systematic literature review. *BMC Medical Ethics*, 14(55):1.

[Scholz et al., 1973] Scholz, C. H., Sykes, L. R., and Aggarwal, Y. P. (1973). Earthquake prediction: a physical basis. *Science*, 181(4102):803–810.

[Schulam et al., 2015] Schulam, P., Wigley, F., and Saria, S. (2015). Clustering longitudinal clinical marker trajectories from electronic health data: applications to phenotyping and endotype discovery. In *The Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2956–2964, Austin, Texas, USA.

[Shalev-Shwartz, 2011] Shalev-Shwartz, S. (2011). Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194.

[Shor, 1968] Shor, N. (1968). The rate of convergence of the generalized gradient descent method. *Cybernetics and Systems Analysis*, 4(3):79–80.

[Shotton et al., 2013] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124.

[Smyth and Keogh, 1997] Smyth, P. and Keogh, E. (1997). Clustering and mode classification of engineering time series data. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 24–30, Newport Beach, CA, USA.

[Smyth and Wolpert, 1999] Smyth, P. and Wolpert, D. (1999). Linearly combining density estimators via stacking. *Machine Learning*, 36(1-2):59–83.

[Swan, 2009] Swan, M. (2009). Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking. *International Journal of Environmental Research and Public Health*, 6(2):492–525.

[Taylor, 2007] Taylor, S. J. (2007). Modelling financial time series.

[Theil, 1992] Theil, H. (1992). A rank-invariant method of linear and polynomial regression analysis. In *Henri Theils Contributions to Economics and Econometrics*, pages 345–381. Springer.

[Tranter and Reynolds, 2006] Tranter, S. E. and Reynolds, D. A. (2006). An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1557–1565.

[Turner et al., 2010] Turner, R. D., Deisenroth, M. P., and Rasmussen, C. E. (2010). State-space inference and learning with Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 868–875, Sardinia, Italy.

[Valko and Hauskrecht, 2010] Valko, M. and Hauskrecht, M. (2010). Feature importance analysis for patient management decisions. In *International Congress on Medical Informatics*, pages 861–865, Cape Town, South Africa.

[Van Overschee and De Moor, 1996] Van Overschee, P. and De Moor, B. (1996). *Subspace Identification for the Linear Systems: Theory - Implementation - Application*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[Victor and Alberto, 2011] Victor, M. M. and Alberto, P. (2011). *Kalman Filter Recent Advances and Applications*.

[Visweswaran and Cooper, 2004] Visweswaran, S. and Cooper, G. F. (2004). Instance-specific Bayesian model averaging for classification. In *Advances in Neural Information Processing Systems*, pages 1449–1456, Vancouver, Canada.

[Visweswaran et al., 2015] Visweswaran, S., Ferreira, A., Ribeiro, G. A., Oliveira, A. C., and Cooper, G. F. (2015). Personalized modeling for prediction with decision-path models. *PloS One*, 10(6):e0131022.

[Wan and Van Der Merwe, 2000] Wan, E. A. and Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, Alberta, Canada. IEEE.

[Wang et al., 2005] Wang, J. M., Fleet, D. J., and Hertzmann, A. (2005). Gaussian process dynamical models. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1441–1448, Vancouver & Whistler, BC, Canada.

[Wang et al., 2008] Wang, J. M., Fleet, D. J., and Hertzmann, A. (2008). Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298.

[Weng et al., 2006] Weng, S.-K., Kuo, C.-M., and Tu, S.-K. (2006). Video object tracking using adaptive Kalman filter. *Journal of Visual Communication and Image Representation*, 17(6):1190–1208.

[Wiley et al., 2016] Wiley, L. K., Tarczy-Hornoch, P., Denny, J. C., Freimuth, R. R., Overby, C. L., Shah, N., Martin, R. D., and Sarkar, I. N. (2016). Harnessing next-generation informatics for personalizing medicine: a report from AMIA's 2014 health policy invitational meeting. *Journal of the American Medical Informatics Association*, page ocv111.

[Xu et al., 2004] Xu, W., Guan, C., Siong, C. E., Ranganatha, S., Thulasidas, M., and Wu, J. (2004). High accuracy classification of EEG signal. In *International Conference on Pattern Recognition*, volume 2, pages 391–394, Cambridge, UK. IEEE.

[Yi and Faloutsos, 2000] Yi, B.-K. and Faloutsos, C. (2000). Fast time sequence indexing for arbitrary Lp norms. In *Proceedings of the International Conference on Very Large Data Bases*, pages 385–394, Cairo, Egypt.

[Zellner and Palm, 1974] Zellner, A. and Palm, F. (1974). Time series analysis and simultaneous equation econometric models. *Journal of Econometrics*, 2(1):17–54.

[Zhou et al., 2012] Zhou, J., Liu, J., Narayan, V. A., and Ye, J. (2012). Modeling disease progression via fused sparse group lasso. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1095–1103, Beijing, China.