NOVEL STATISTICAL LEARNING AND DATA MINING METHODS FOR SERVICE SYSTEMS IMPROVEMENT

A Thesis
Presented to
The Academic Faculty

by

Chitta Ranjan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2016

NOVEL STATISTICAL LEARNING AND DATA MINING METHODS FOR SERVICE SYSTEMS IMPROVEMENT

Approved by:

Professor Kamran Paynabar, Advisor
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Yajun Mei
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Jianjun (Jan) Shi
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Jonathan E Helm
Kelley School of Business
Indiana University

Professor Brani Vidakovic
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Date Approved: 06 September 2016

Dedicated to my parents Chinta and Dr. Radha Krishna Prasad for their sweet love and support in my life...

ACKNOWLEDGEMENTS

My heartfelt thanks goes to my advisor, Professor Kamran Paynabar, for his splendid guidance. His openness to hearing and encouraging new ideas helped me explore new fields and gain self-confidence in doing innovative research. Besides, his kind support in regular graduate life issues made my PhD journey easier.

I want to give my special thanks go to all other committee members, Professor Jan Shi, Professor Brani Vidakovic, Professor Yajun Mei, and Professor Jonathan E. Helm, for their invaluable comments, suggestions and guidance during the course of this dissertation.

Besides, I want to thank Professor Jonathan Helm again, for all I learnt while collaborating with him during the initial phase of my PhD. Those initial learnings has helped me at several other occasions later.

My special gratitude goes to my former undergraduate advisor, Professor Jhareswar Maiti, for instilling the passion of research in me. His strong confidence in me is one of the reasons that I pursued graduate studies and, for which, I will be always thankful.

Finally, I want to express my deep gratitude to my parents for their love and support, which gave me the will and strength to keep up in all of my life. Also, hearty thanks to my siblings Madhuri and Rashmi, and my friends Samaneh and Nilesh to name a few, without their steady availability and encouragement it would not have been easy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation focuses on solving problems for service systems improvement using newly developed data mining methods. A large variety of problems fall under the service systems category. Chapter 1 elucidates more on this and, also, the need of data driven decision making for these systems. Among a large plethora of problems in this realm, this dissertation attempts to solve three distinct and critical research problems. Chapter 1 briefly discusses the motivation and challenges behind these problems, while Chapter 2-4 explores each of them in detail and presents a novel solution.

In Chapter 2, a classical problem of accurately forecasting patient census, and thereby workloads, for hospital management is studied. Majority of current literature focuses on optimal scheduling of inpatients, but largely ignores the process of accurate estimation of the path of patients throughout the treatment and recovery process. The result is that current scheduling models are optimized based on inaccurate input data. We developed a Clustering and Scheduling Integrated (CSI) approach to capture patient flows through a network of hospital services. CSI works differently by clustering patients into groups based on the similarity of paths, instead of admit, condition, or other physical attributes. To that end, we develop a novel Semi-Markov model (SMM)- clustering scheme. The methodology is validated by simulation and then applied to real patient data from a partner hospital where we see it outperforms current methods. Further, we demonstrate that extant optimization methods achieve significantly better results on key hospital performance measures under CSI, compared with traditional estimation approaches, increasing elective admissions by 97% and utilization by 22% compared to 30% and 8% using traditional estimation techniques. From a methodological standpoint, the SMM-clustering is a novel approach applicable to any temporal-spatial stochastic data that is prevalent in many industries and application areas.

In Chapter 3, data analysis problems in a special scenario — longitudinal data with measurement errors but absence of replicates — is studied. Longitudinal data is commonly

found across fields, and sometimes has measurement errors. Especially, if the data collection has several processing stages, like MRI scans in medical fields. Multiple measurements (replications) are often taken at the same time to gauge its error and correct the analysis. However, obtaining replicates are sometimes not possible due to cost or associated risks, for instance, MRI scans are taken at long intervals due to high costs. Inferences derived from such erroneous data can be unreliable and, in medical diagnosis, can be fatal. We, therefore, devise a new estimation approach, called as EM-Variogram, that utilizes the autocorrelation aspect of longitudinal data to isolate the variance from measurement errors. This estimation approach enables a more reliable data analysis and a powerful statistical test of model parameters. Upon using this methodology on Alzheimer disease patients, we could quickly and precisely detect any signal of decline in patients' conditions. This can prove to be extremely useful for providing any required treatment to the patients to improve their conditions. Besides, other possible applications are also discussed in the chapter.

Chapter 4 works on one of the most commonly found data type – sequences. It has a ubiquitous presence across fields, like, web, healthcare, bioinformatics, text mining, etc. This has made sequence mining a vital research area. However, sequence mining is particularly challenging because of an absence of an accurate and fast approach to find (dis)similarity between sequences. As a measure of (dis)similarity, mainstream data mining methods like k-means, kNN, regression, etc., have proved distance between data points in a euclidean space to be most effective. But a distance measure between sequences is not obvious due to their unstructuredness — arbitrary strings of arbitrary length. We, therefore, propose a new function, called as Sequence Graph Transform (SGT), that extracts sequence features and embeds it in a finite-dimensional euclidean space. It is scalable due to a low computational complexity and has a universal applicability on any sequence problem. We theoretically show that SGT can capture both short- and long- term patterns in sequences, and provides an accurate distance-based measure of (dis)similarity between them. This is also validated experimentally. Finally, we show its real world application for clustering, classification, search and visualization on different sequence problems.

Lastly, Chapter 5 concludes the dissertation by summarizing the research contributions,

outcomes and future directions.

CHAPTER I

INTRODUCTION

1.1 Background

A service system is a configuration of technology and organizational network or enterprise developed and designed to deliver services that satisfy the needs, wants, or aspirations of customers (Cardoso et al., 2014). They range from providing basic services, like, daily consumer needs, to complex operations, like, healthcare, business, finance, weather forecast, biological studies, etc. All service enterprises aspire to continually improve the quality of their services for a better customer experience and gaining competitive advantages.

This urge for improvement has led to extensive investments in data collection infrastructure throughout the enterprise, primarily for the purpose of extracting useful information. These information and knowledge are used for data-driven decision making (DDM) and performance improvements. Economist Erik Brynjolfsson and his colleagues from MIT and Penn's Wharton School conclusively demonstrated the benefits of DDM. Brynjolfsson et al., (2011) studied the effects of DDM on the performance of a system and showed that statistically — even controlling for a wide range of possible confounding factors — the more data-driven a system is, the higher productivity it has.

These developments have led to a significant growth in data collection for analysis. The volume and variety of data have far surpassed the capacity of manual analysis. At the same time, computing technology has seen tremendous developments. The convergence of these phenomena has given a strong impetus to the development of more statistical learning and data mining methods for application in these areas (Provost & Fawcett, 2013 and Friedman, et al., 2001).

Over the past two decades several areas of research have focused on for system improvement. Still we find critical deficiencies in performance of most service systems, providing a large scope for new research.

In this dissertation, we focus on some of the important problem areas in this realm and present our proposed statistical learning and data mining solutions to improve a system's performance. In the following section, we briefly explain the problems, the motivation behind them, challenges, and our proposed solution.

1.2 Research topics

1.2.1 The Impact of Estimation: A New Method for Clustering and Trajectory Estimation in Patient Flow Modeling

Despite of an overwhelming expenditure on healthcare, this sector is plagued with severe inefficiencies in the U.S. There are about 37 million admissions in US hospitals every year, however, the experiences of these patients are usually not pleasant. The wait times and delays in service have become intrinsic and intractable in the healthcare system. A study in 2007 shows about 1.9 million patients walk-out without treatment (Niska, et al., 2007). Moreover, the inefficiencies sometimes lead to preventable medical errors. These medical errors are fatal to patients, with losses estimated to be $21 billion and 10 million days of lost productivity. In addition, there is about 20% increase in patient visits year-over-year, which may make the situation worse if the inefficiencies are not addressed.

A cursory look on this problem would suggest a lack of adequate resources to meet the demand, for which, an increase in healthcare resources should mitigate the problem. However, contrary to this perception, a recent healthcare statistics survey shows an average utilization of just 49% of hospital resources. A deeper look indicates that the fundamental issue lies in the mismanagement of existing hospital resources.

Over last two decades, several researchers developed optimization models for optimal resource scheduling. In fact, most current models, are efficient, scalable, and non-heuristic. However, their performance relies primarily on the quality of patient flow estimations, for which existing methods are still ineffective. Generally, hospital management has little or ad-hoc knowledge about the expected patient flows at any point of time, leading to poor utilization of their resources.

Therefore, the main challenge faced by healthcare systems is the estimation of patient flow inside a hospital. There is a unanimous consensus among healthcare institutions, viz.

Joint Commission, Institute of Healthcare Improvement and Institute of Medicine, on "accurate patient flow estimation as a panacea to the hospital system inefficiencies".

Patient flow estimation is a non-trivial problem attempted by a few researchers. Simulation based methods were developed by Hancock & Walter, (1979), Hancock & Walter, (1983), Griffith, et al., (1976), Harper & Shahani, (2002), Jacobson, et al., (2006), Konrad, et al., (2013), Zeltyn, et al., (2011) to mimic patient flows and perform optimal resource scheduling, but such methods are limited to specific case scenarios, and thus, difficult to generically apply on any hospital system. Empirical patient flow distributions were used by Helm & Van Oyen, (2015), after grouping the patients into homogeneous groups with respect to their attributes, using methods like k-means, Diagnosis-related-groups (DRG), classification trees, etc. (Fetter, et al., 1980 and Harper, 2005). These approaches highlight and address a critical challenge in patient flow estimation – heterogeneity in patients. Any general hospital witnesses a wide variety of heterogeneous mix of patients having different age groups, genders, disease, diagnosis, nationality, etc.

However, these methodologies are built on a weak assumption: patients with same attribute will exhibit similar flows inside a hospital. For example, two patients with same age group, sex and diagnosis, are assumed to have similar flows. On the contrary, we find that patients with same attributes can have entirely different paths — a possible reason being inability of capturing all patient attributes that characterizes his/her path.

We, thus, develop a novel semi-Markov mixture (SMM) model clustering that groups patients based on similarity of their paths and, also, estimates the path distributions for each group. The proposed clustering method models a patient flow as a semi-Markov process. A semi-Markov process is a stochastic process that satisfies the Markov property (i.e. conditional probability distribution of future states of the process depends only upon the present state) and has different holding time distributions for each possible transition between states. In a patient-hospital scenario, a state is the location or ward of the patient at any point of time.

The semi-Markov assumption on patient flow has been proposed and proved in early works by Thomas (1968), Kao (1972, 1974), and Weiss et al. (1996). Similar stochastic

process are found in data collected for weather, human behavior (say, purchase pattern, click streams), gambling, birth-death processes, etc. Many such problems require clustering the flow (trajectory) data. However, despite the importance of such clustering technique, very few research has been done to address it.

Cadez et al. (2003) developed a Markov mixture model based clustering method. Their proposed method made a significant development in addressing the above-mentioned problem. However, their model had a critical deficiency — assuming the same holding time distribution for all possible transition between states. If applied to our problem, this will lead into two important shortcomings, a) grouping patients into one cluster even when their length-of-stays (LOS) in each state are different, and b) making inaccurate temporal estimation of a patient's location (ward).

The proposed SMM clustering is, thus, a significant contribution from a methodological and application standpoints in several ways: a) it overcomes the foregoing shortcomings, b) it accurately clusters a data with a spatio-temporal structure, and, c) it has broad application beyond health care including user behavior analysis in online industry, weather forecast, etc.

To implement the methodology, we incorporate a Bayesian framework with non-informative priors to address potential occurrences of missing information (transitions between certain wards). EM algorithm was applied and closed form update rules were derived, which guarantees convergence. We, then, seamlessly integrate the clustering approach with a mixed integer programming model for optimal resource scheduling. We call this framework as Clustering and Scheduling Integrated (CSI) approach. The proposed CSI approach is/addresses

- *Scalable*: It can be applied on a hospital system of any *size*, with any number of wards and discharge states, and any complex interconnections between them.

- *Generic*: Any specialty, for example infants or geriatric, or multi-specialty hospital, for example community hospital, can use the CSI approach.

- *Ward interactions*: Unlike several existing methods, CSI takes into account the complex connections, and thus, interactions between hospital wards — performs a holistic optimization by treating the hospital as a system.

- Patient *heterogeneity*: It effectively incorporates the phenomena that different patient types behave differently, and thus, require different hospital resources/services.

Overall, the proposed approach effectively addresses the deficiencies of existing research and significantly outperforms them in terms of accuracy and optimization. In this dissertation, we present and discuss the approach in detail in Chapter 2, validate it using simulation, and demonstrate its applicability and superiority over existing methods through a case study.

1.2.2 Longitudinal MRI Data Analysis in presence of Measurement Error and absence of Replicates

Tremendous advancements have been made in measurement systems in various fields. Analysis of data acquired by measurement systems provides a basis for decision making and planning. Therefore, the accuracy and precision of measurement systems is of special importance. However, if the measurement system leads to significant amount of measurement errors in the collected data, the resultant data analysis will be unreliable. Impact of any such unreliable analysis can be quite adverse. Chapter 3 works on longitudinal processes under such circumstances with a focus on medical decision making. It must be noted that poor inferences can be even more severe and can have fatal consequences in medical diagnosis and treatments.

Take for example, analysis of MRI scans of brain of Alzheimer disease patients. The MRI scans are taken over time to collect longitudinal data. The scans are, however, underwent several stages of image and other processing to yield some tangible metric levels. For instance, size of hippocampal, a part of the brain. This multi-stage data collection process can lead to inclusion of measurement errors.

An accurate data analysis of such data is rather straightforward if data replicates are available — variance among replicates provide an estimate for the measurement error. In fact, almost all measurement system error analysis methods rely on some type of replicates in the data, discussed in the literature review in Chapter 3. On the contrary, some longitudinal processes do not necessarily have replicates. Such scenarios are common if the data collection is costly or has inherent risks in acquisition. For instance, in the above MRI example, data

is collected at long time intervals due to high cost.

Under such circumstances, it is unclear whether the observed variance is due to an actual change or because of measurement error in data collection. Besides, measurement errors also inflate the variance of any true underlying effect, making the effect hard to detect. For example, due to these errors, it can be difficult to detect if a patient's condition is declining over time due to a disease effect, or if she is positively responding to some treatment.

Such undetected effects can be particularly harmful for patients — it can lead to a late detection or a poor treatment. Therefore, there is a critical need of a methodology that can isolate any measurement error in the absence of replicates and provide an accurate data analysis.

In Chapter 3, we develop a new estimation technique, called EM-Variogram, that utilizes autocorrelation property in longitudinal data to use a special parametric error covariance structure in a linear mixed effect (LME) model to decouple the measurement errors from the overall error. The developed methodology is robust to missing values, a common phenomenon in longitudinal data. Besides, it provides a more powerful statistical hypothesis test due to isolation of the measurement errors. This leads to quicker and precise detection of any true underlying effects.

Chapter 3 further discusses this problem and explores other related methods and their shortcomings. The methodology development is then presented, and experimentally validated. Besides, its application is shown on analysis of progression of cognitive decline in Alzheimer disease patients. Besides, other possible applications, outside of the medical field, are also discussed.

1.2.3 Sequence Graph Transform (SGT): A Feature Extraction Function for Sequence Data Mining

Sequences are pervasively found around us in diverse fields like healthcare, bioinformatics, web, marketing, text mining, social science, etc. A sequence is a series of discrete objects or events, e.g., `BAABCCADECDBBA`, sometimes also called as strings. A sequence of events are closely related to a time-series, but, represent discrete events denoted by a set of symbols or alphabets. Some examples of sequences are, web logs, music listening history, patient

movements through hospital wards, DNA, RNA and protein sequences in bioinformatics.

This common presence of sequence data has propelled development of new sequence mining methods. Some of its motivating applications are, a) understanding users behavior from their web-surfing or buying sequences data, to serve them better advertisements, product placements, promotions, and so on, b) assessing process flows (sequences) in a hospital to find the expected patient movement based on her diagnostic profile, to better optimize the hospital resource and service, c) analysis of biological sequences to understand human evolution, physiology and diseases, etc.

The main challenge for these analyses is finding similarity between sequences. This is particularly difficult for sequences because they are made of arbitrarily ordered alphabets for any arbitrary length. A large amount of research has been done in this area, but still there is a lack of an efficient method that can work on any sequence data. Several methods exist in bioinformatics, like UCLUST (Edgar, 2010), CD-HIT (Fu et al., 2012), MUSCLE (Edgar, 2004), etc., but they cannot be easily extended to sequences outside of bioinformatics. Besides, more common methods work on one of the following assumptions, i) sequence process has an underlying parametric distribution, ii) similar sequences have common substrings, or iii) sequence evolves from hidden strings.

Parametric methods usually work on a Markovian assumption, of first-order due to high computation otherwise (Cadez et al., 2003 and Ranjan et al., 2015). This results in over-simplification of the problem at hand by ignoring higher order correlations. Hidden Markov model based approaches are also developed that relaxes the first-order constraint in the observed sequence (HHblits: Remmert et al., 2012; Helske and Helske, 2016). However, optimizing an HMM (finding optimal hidden states) is difficult and it is computationally intensive, thus, affecting its *generality* and *scalability*. Moreover, if the distributional assumption is invalid, these methods may yield poorer result.

Common substrings (like, n-gram, Tomović et al., 2006) and evolution-from-strings (Siyari et al., 2016) methods attempt to match substrings derived from sequences to measure their similarity. In the former methods, optimizing length of common substrings for comparison is difficult. Besides, the latter methods require a search in an unobservable universe to

find the hidden evolutionary tree of substrings, causing identifiability and accuracy issues.

To sum up, the existing sequence data mining methods lack in effectiveness due to the absence of a good measure of (dis)similarity between sequences. It is known that *distance* between objects in a euclidean space proves to be an efficient (dis)similarity measure. Almost all mainstream data mining methods use a euclidean measure, e.g., in k-means clustering the *distance* between objects (data points) within a cluster are minimized while *distance* between clusters are maximized, in classification models, like SVM or logistic regression, the *distance* of a boundary is minimized or maximized from the objects.

Therefore, in Chapter 4, we develop a Sequence Graph Transform (SGT) function, that performs a feature extraction of sequences in a finite-dimensional euclidean feature space. This can also be viewed as an embedding space, where the objective is to transform a sequence into a feature vector, such that the features capture the sequence characteristics. Besides, by definition, the embedding space has the same dimension for all sequences in a data corpus. This will facilitate computation of (dis)similarity between two sequences by measuring the *distance* between their embeddings. Simply put, (dis)similarity computation will be finding a euclidean distance between the two sequences' feature vectors.

While related methods fail in at least one of the following key challenges in sequence mining, SGT addresses all of them: a) Feature mapping: Effective extraction of sequence characteristics into a finite-dimensional euclidean space (a vector), b) Universal applicability: This mainly requires absence of any distributional or a domain specific assumption, and a small number of tuning hyperparameters, and c) Scalability: It relies on the computational complexity, which should be small with respect to sequence length, size of the database and the alphabets set.

SGT works by quantifying the pattern in a sequence by scanning the positions of all alphabets relative to each other. We call it a *graph* transform because of its inherent interpretation property as a graph, where the alphabets form the nodes and a directed connection between two nodes shows their "association". These "associations" between all alphabets represent the characteristic features of a sequence. A Markov model transition matrix can be compared analogously with the SGT's feature space, however, a) the associations (graph

edges) do not represent a probability and SGT is non-parametric, and b) SGT captures both short- and long- term patterns.

Chapter 4 presents this approach in detail and highlights the major contributions, viz. a) development of a new feature extraction function, SGT, for sequences, b) a theoretical and experimental evaluation of SGT, and c) illustration through real data examples that SGT bridges the gap between sequence mining and mainstream data mining through implementation of fundamental methods, viz. PCA, k-means, SVM and graph analysis techniques for sequence clustering, classification, visualization and search operations.

1.3 References

Brynjolfsson, E., Hitt, L. M., & Kim, H. H. (2011). Strength in numbers: How does data-driven decisionmaking affect firm performance?. Available at SSRN 1819486.

Cadez, I. et al., 2003. Model-based clustering and visualization of navigation patterns on a web site. Data Mining and Knowledge Discovery, Volume 7(4), pp. 399-424.

Edgar, R. C., 2004. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC bioinformatics, pp. 5(1), 113.

Edgar, R. C., 2010. Search and clustering orders of magnitude faster than BLAST. Bioinformatics, pp. 26(19), 2460-2461.

Fetter, R. B., Shin, Y., Freeman, J. L., Averill, R. F., & Thompson, J. D. (1980). Case mix de nition by diagnosis-related groups. Medical care, 18(2), i-53.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer, Berlin: Springer series in statistics.

Fu, L. et al., 2012. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics, pp. 28(23), 3150-3152.

Griffith, J. R., Hancock, W. M., & Munson, F. C. (Eds.). (1976). Cost control in hospitals. Health Administration Press.

Hancock, W. M., & Walter, P. F. (1979). The use of computer simulation to develop hospital systems. ACM SIGSIM Simulation Digest, 10(4), 28-32.

Hancock, W. M., & Walter, P. F. (1983). The "ASCS": Inpatient Admission Scheduling

and Control System. Health Administration Press.

Harper, P. R., & Shahani, A. K. (2002). Modelling for the planning and management of bed capacities in hospitals. Journal of the Operational Research Society, 53(1): 11-18.

Harper, P. R. (2005). A review and comparison of classi cation algorithms for medical decision making. Health Policy, 71(3), 315-331.

Helm, J. E., & Van Oyen, M. P. (2015). Design and optimization methods for elective hospital admissions. Operations Research, 62(6), 1265-1282.

Helske, S. and Helske, J. (2016). Mixture hidden markov models for sequence data: the seqhmm package in R.

Jacobson, S. H., Hall, S. N., & Swisher, J. R. (2006). Discrete-event simulation of health care systems. In Patient ow: Reducing delay in healthcare delivery (pp. 211-252). Springer US.

Kao, E. P. (1972). A semi-Markov model to predict recovery progress of coronary patients. Health Services Research, 7(3), 191.

Kao, E. P. (1974). Modeling the movement of coronary patients within a hospital by semi-Markov processes. Operations Research, 22(4), 683-699.

Konrad, R., DeSotto, K., Grocela, A., McAuley, P., Wang, J., Lyons, J., & Bruin, M. (2013). Modeling the impact of changing patient ow processes in an emergency department: Insights from a computer simulation study. Operations Research for Health Care, 2(4), 66-74.

Niska, R., Bhulya, F. & Xu, J., 2007. National Hospital Ambulatory Medical Care Survey: 2007 Emergency Department Summary, Hyattsville, MD: National Center for Health Statistics; 2010..

Provost, F., & Fawcett, T. (2013). Data Science for Business: What you need to know about data mining and data-analytic thinking. " O'Reilly Media, Inc.".

Ranjan, C., Paynabar, K., Helm, J. E., & Pan, J. The Impact of Estimation: A New Method for Clustering and Trajectory Estimation in Patient Flow Modeling.

Remmert, M., Biegert, A., Hauser, A., & SÃ¶ding, J. (2012). HHblits: lightning-fast

iterative protein sequence searching by HMM-HMM alignment. Nature methods, 9(2), 173-175.

Siyari, P., Dilkina, B., and Dovrolis, C. (2016). Lexis: An optimization framework for discovering the hierarchical structure of sequential data. arXiv preprint arXiv:1602.05561.

Thomas, W. H. (1968). A model for predicting recovery progress of coronary patients. Health services research, 3(3), 185.

Tomović, A., Janičić, P., and KeÅ¡elj, V. (2006). n-gram-based classification and unsupervised hierarchical clustering of genome sequences. Computer methods and programs in biomedicine, 81(2):137–153.

Weiss, E. N., Cohen, M. A., & Hershey, J. C. (1982). An iterative estimation and validation procedure for speci cation of semi-Markov models with application to hospital patient ow. Operations Research, 30(6), 1082-1104.

Zeltyn, S., et al. (2011). Simulation-based models of emergency departments: Operational, tactical, and strategic sta ng. ACM Transactions on Modeling and Computer Simulation (TOMACS), 21(4), 24.

CHAPTER II

THE IMPACT OF ESTIMATION: A NEW METHOD FOR CLUSTERING AND TRAJECTORY ESTIMATION IN PATIENT FLOW MODELING

2.1 Introduction

The mismatch between demand for and supply of medical services caused by high hospital census variability has challenged hospital managers for decades. High census variability is a common problem in hospitals and healthcare centers around the world. This problem leads to poor quality of care, blocking in hospital wards, increase in inpatient length of stay and ultimately causes significant increase in cost for both patient and hospital (Helm and Van Oyen 2015). Aiken et al. (2002) studied the effect of overloaded nursing staff induced by census variability and showed its effect on mortality rate, nurse burnout and job dissatisfaction. A common approach to managing census variability in practice involves hospitals procuring excess resources including material, staff, and equipment, leading to frequent instances of under-utilization for very expensive resources (Griffin et al. 2012). A better approach is to optimize the utilization of available hospital resources based on patient census estimations. This long-standing problem has been termed the *Hospital Admission Scheduling and Control* (HASC) problem, which can be decomposed into two main steps: *census modeling* (CM) and *resource scheduling* (RS). CM estimates distributional information (typically mean and variance) on patient census at the ward level, which is used as an input to the RS to find the optimal resource allocation plans and schedules for elective inpatient admissions.

A significant body of work addresses the RS through a variety of optimization approaches, however research on effective census models that integrate with RS is less developed. In this paper we develop a CM method that integrates well with existing RS methods to solve the HASC. We further demonstrate the importance of the CM component with respect to the outcomes of the RS optimization; a factor that has, to our knowledge, been unaddressed

in the current literature. Namely, the method of CM proposed by RS optimization papers leads to markedly inferior optimization results when compared with our new method in a case study based on data from our industry partner. To conclude this section, we give a short description of the current state of the hospital census forecasting and optimization industry from the experiences of our industry co-author and CEO of a healthcare analytics company. Then we address the challenges posed by the gaps in CM theory that represent a major hurdle for this burgeoning industry and discuss how our approach seeks to bridge those gaps.

2.1.1 Real-world Challenges in the Hospital Census Forecasting Industry

Predicting future hospital census levels is a key challenge in the Hospital Admission Scheduling and Control (HASC) problem. Without accurate forecasting mechanisms, controlling the variability in hospital census levels is a major barrier to low cost, high quality inpatient care. These consequences of inadequate forecasting are drawn from real-world experience, where our co-author has worked with clients and collaborators globally - Asia, Europe and North America. All the hospitals he has worked with experience significant mid-week congestion and high levels of blocking.

Current methodologies used in hospitals are ineffective to solving the HASC problem. Almost all the hospitals have lean teams focused on process improvement and some of the bigger hospitals have small analytics teams that use rudimentary models which are ineffective at implementing changes made to solve the HASC problem. All the work done at the hospital level are reactive models (predicting census levels using historical census means, and applying control by canceling surgeries the day before) versus proactive models (implementing control measures in advance). Recently, some hospitals have been attempting to shift to proactive measures. This has typically involved increasing capacity and lowering utilization, which is cost prohibitive in the long-run. The real solution is to improve the forecasting technology. The methods outlined in this paper have proven to be effective on a conceptual level with results shared in the later sections.

Our collaborator, company XYZ (the real name of the company is currently disguised for

the review process), is one of the first to provide a patient level forecasting tool; i.e. predicting individual flows and trajectories of each type of patient entering the hospital. A patient level forecasting and control tool is imperative for hospitals to effectively solve the HASC problem. While forecasting is the backbone to the solution, XYZ also provides the ability for hospitals to create what-if scenarios by modifying admission plans and schedules and to use optimization techniques to customize a dynamic admission plan to minimize blocking and surgical cancellations. This type of analysis and decision support is only possible through patient-level forecasting as it requires understanding how patient-by-patient modifications to the admission schedule impact hospital census and blocking. This is precisely the type of forecasting that we propose in this paper. In fact, workload forecasting is not only useful for bed planning purposes, but is key to allocating resources to the various aspects of the hospital. Most notably, workforce planning for front and back end staff accounts for over 50% of the hospital costs. Based on the feedback received from XYZ clients, properly allocating staffing reduces various costs, like overtime, and improves staff satisfaction. Overall, it is one key in keeping hospitals profitable and delivering top quality healthcare. After discussing the various needs of the hospitals, it is clear that the key issues in patient flow management, staffing, and scheduling all rely on the critical role of forecasting flexibility and accuracy.

One ongoing challenge for XYZ is the issue of defining Patient Types (PTypes) and estimating their probabilistic trajectories over the course of their hospital stay, both of which have a major affect on forecast accuracy. From a computational standpoint, it requires clustering patients into groups, where each group represents one type of patient. Currently, XYZ employs various forms of regressions to determine factors to group similar patients together into clusters based on patient characteristics. Many assumptions must be made to fit data into logical PTypes that are scalable and yet give enough information to statistically differentiate patients and enable accurate forecasting. This includes applying numerous heuristics and unfortunately, sacrificing the accuracy of the forecast. At XYZ, this process is currently done manually for each hospital, often requiring weeks to months of effort to properly tailor the PTypes for accurate forecasting. These issues of scalability, repeatability,

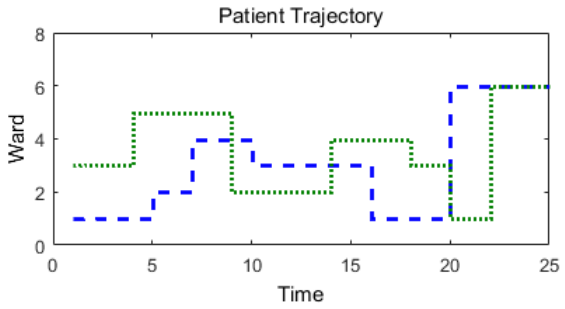


Figure 1: Example of sample path outcomes of a stochastic location process for patient flow. The x-axis shows the time after admission, while the y-axis denotes the ward the patient is in at time t; each step is a change of ward for the patient.

and demonstrated statistical accuracy represent one of the major hurdles for XYZ and other participants in the patient-level forecasting space. The methods presented in this paper help solve a key problem in parameterizing models for each hospital. Specifically, by clustering patients based on trajectory (rather than extrinsic characteristics as in current practice) this paper significantly improves upon the currently time consuming and heuristic step of assigning PTypes. Our approach is shown to be scalable, statistically rigorous, accurate, and repeatable. This eliminates the time consuming, gestalt guess work inherent in current practice and has proven to significantly increase forecast accuracy in addition to improving the results from current decision support methods for admission scheduling.

2.1.2 Failures of Traditional CM Methods.

As noted in Fetter et al. (1980) and Helm and Van Oyen (2015), an appropriate HASC model should have three characteristics: *scalable* to hospital of any size, considers *ward interactions*, and accounts for *patient heterogeneity*. Most work in the RS step assumes that patient types are given and uses simple methods for estimating patient trajectories, then employs analytical techniques to capture key hospital metrics in an optimization model. A patient trajectory is characterized by the transitions between wards in a hospital and patient Length of Stay (LOS) in each ward, and can be expressed as a stochastic function called a location process that maps time to a set of locations — see Fig. 1 for an example of two sample path outcomes of a location process.

While the optimization methods are generalizable, the previous approaches to CM for

RS optimization lack scalability and are not well suited for capturing patient heterogeneity or sufficient depth in terms of ward interactions. In this paper we address these issues by developing new methods for clustering patient location processes based on historical patient flow data.

As an example of how clustering impacts scalability and patient heterogeneity, consider the following. Many traditional approaches to HASC cluster patients by their diagnosis related group (DRG) or admitting service. However, in working with a large hospital such as our partner hospital, there can be close to one hundred such patient types with quite a few of them being very rare. With such a large number of patient types we have found that there is insufficient data to properly estimate patient trajectories even with two or more years of historical data. When further including other important factors such as gender and age, which have been found to be important in determining a patient's trajectory, data scarcity becomes an even larger problem. Current solutions include combining different patient types that are deemed "similar" in order to have sufficient historical data for trajectory estimation. This is a *clustering* problem. Deciding how to combine patient types, however, is a non-trivial effort considering the entire location process (time and location) must be compared to ensure an accurate pairing of two patient types. For example, two patient types may have the same average length of stay (LOS) in the hospital but visit different wards. Another example is if two patients visit similar locations with similar mean LOS, but one has a skewed LOS distribution and the other does not. These factors can all have a significant impact on census forecast accuracy (see Littig and Isken 2007). Because different hospitals have different methods for categorizing patients (different admitting services, DRGs served, etc.), this requires a lengthy and ad-hoc procedure to be performed at each new hospital, significantly impacting scalability. For example, our industry co-author has indicated that this process of clustering under current methods is unique to each hospital and can take months to adequately determine patient types in large hospitals.

A second problem is that once the patient types have been identified, trajectories are assigned based solely on what patient type the patient is identified as. For example, if the patient is a bladder cancer surgery patient their cluster will be bladder cancer surgery.

However, other factors that may impact the patient's trajectory and LOS, such as age and gender, cannot be considered after the patient types are defined. This approach is only as good as the granularity of each cluster. However, the clusters are not defined based on the shape of patients' location functions, but rather on other factors available in the data that are believed to be associated with the shape of the location function, but have not been statistically validated. Finally, clusters cannot be too granular or data will be insufficient. This phenomenon impacts both the ability to capture patient heterogeneity and to accurately estimate patient paths because patients are forced into predefined groups rather than assigned a type that most closely matches their projected trajectory.

In contrast, in this paper we develop a new clustering approach that clusters patients directly according to similarity of their trajectories (which is what we want to estimate) in a statistically rigorous manner, rather than using these ad-hoc proxies (e.g. DRG, age, gender). Specifically, we seek to close the gap in the literature by developing new methods for the CM step that provide more effective and scalable clustering of patient types, and a better estimation of the patient trajectories for each patient type. The proposed model, which we call *clustering and scheduling integration* (CSI) is scalable, captures the interactions between hospital wards, and is capable of handling patient heterogeneity. CSI begins with the CM module in which heterogeneous patients are clustered based on the similarity of their trajectories. This provides patient types for accurate estimation of patient trajectories and patient census distributions at the ward level. Finally, these estimates serve as inputs to the RS module to find an optimal hospital resource schedule, which is then shown to outperform the same optimization model using traditional CM methods.

For CM, we propose a novel semi-Markov mixture model (SMM) that integrates the mixture clustering method and semi-Markov models accurately describing stochastic location processes of patient trajectory. To the best of our knowledge, this SMM clustering technique has not been proposed before in the literature, either for the HASC problem or any other problem. The SMM not only clusters patients based on their trajectory, but also provides accurate estimates for the trajectory distribution of each group of patients. In the RS module, the output of the CM is fed into an MIP model similar to the model proposed by

Helm and Van Oyen (2015) to find the optimal resource schedule for hospitals.

We further show through a case study using real data from a partner hospital that system performance is significantly impacted by the quality of the input from the CM step. In fact, using CSI to parametrize the optimization can enable up to a 50% increase in elective admissions while maintaining the same level of blocking and internal congestion when compared with the same optimization using the traditional estimation approach. Similarly, it is possible to have higher ward utilization compared with traditional CM approaches holding all other metrics constant.

The remainder of this chapter is organized as follows. We first review the literature and position the paper in Sec. 2.2. Next, we develop the new CSI methodology in Sec. 2.3, in which the SMM clustering method for CM is discussed in detail, followed by a brief description of the MIP model used for RS. Then in Sec. 2.4 we use simulation to validate the proposed CSI model in terms of the accuracy of estimates and the optimality of solutions. Finally, in Sec. 2.5, we apply our CSI methodology in a case study based on historical data from a partner hospital.

2.2 Literature

Most existing research in the HASC area has focused on either CM or RS, separately, and little work can be found on integrating CM and RS in a cohesive framework. Additionally, existing HASC approaches lack at least one of the aforementioned characteristics of an effective HASC model. The aim of this paper is to develop an HASC framework that is scalable, accounts for patient heterogeneity, and considers ward interactions through effective integration of CM and RS.

In the HASC literature, various stochastic and deterministic models have been developed for RS. Green (2006) and Armony et al. (2011) used queuing models for patient arrival to optimize resource scheduling. Ward interactions were not taken into account in either of these models. Unlike the queuing models, simulation models developed for RS are more flexible and consider the interaction between wards, mostly by using patient pathways between wards in a hospital. Examples of simulation-based models include Hancock and

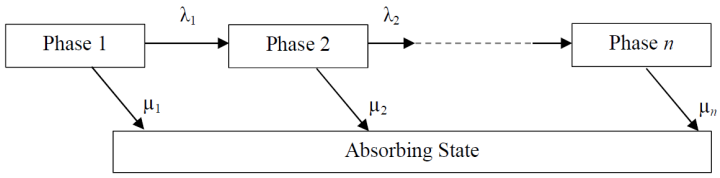
Walter (Hancock and Walter (1979, 1983)), Griffith et al. (1976), Jacobson et al. (2006), Harper and Shahani (2002), Zeltyn et al. (2011), and Konrad et al. (2013). However, simulation models are case-specific, cannot be easily generalized or scaled, and rely on the same, less effective PType and path estimation techniques mentioned earlier. Adan et al. (2009), Bekker and Koeleman (2011), and Zhang et al. (2009) used Mixed Integer Programming (MIP) models for optimal RS. These works, however, only focus on either one ward or an isolated feed-forward subset of the hospital, ignoring ward interactions. To address this issue, Helm and Van Oyen (2015) proposed a non-heuristic MIP scheduling model that also used patient pathways to model ward interactions of an entire hospital. Although the RS portion of the model is scalable and considers ward interactions, it does not properly handle patient heterogeneity. Moreover, an empirical method (similar to the traditional method described above) was used to estimate the patient census at the ward level, which we show can degrade the value of the optimal solution.

For RS optimization to be maximally effective, an accurate CM is required to estimate patient arrival rates, their trajectory through the hospital, and, by combining arrival and trajectory, the patient census at both the ward and hospital levels. Regression analysis and time-series modeling have been widely used for forecasting inpatient admissions and hospital occupancy (Channouf et al. (2007), Earnest et al. (2005) and Jones et al. (2002)). Abraham et al. (2009) reviewed and compared several models for forecasting daily emergency inpatient admissions and occupancy. They found that the admissions are largely random and hence non-predictable, whereas occupancy can be forecasted using a model combining regression and ARIMA, or a seasonal ARIMA, for up to a week ahead. Their model is capable of forecasting the overall hospital occupancy, but not the occupancy at the ward level. Consequently, it does not take the ward interaction into account. These approaches are also incapable of capturing what-if scenarios or optimization with respect to inpatient admission decisions mentioned previously as a demonstrated need in the census forecasting and decision support industry. Littig and Isken (2007) used occupancy flow equations to estimate occupancy at different units or wards of a hospital. They predicted patient in- and out-flow using time series and multinomial logistic regression models. They combined these

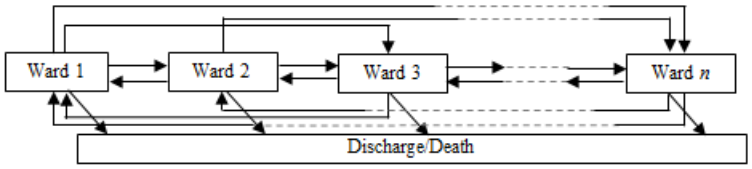
predictions and fed them into a set of flow equations to find the net estimate of the number of patients in a given ward. However, implementing this model in real time presents a major challenge, as even a simple model requires coordination between a variety of real time data sources and the computational burden of the method is high, so scaling this model to large hospital would be difficult.

To model patient trajectory and LOS, Irvine et al. (1994) and Taylor et al. (2000) proposed a continuous time Markov model for geriatric patients. This model, however, was developed for few wards and lacks scalability. Moreover, the assumption that the LOS at each ward follows the same exponential distribution is not often a good model of reality. Faddy and McClean (2000) used Phase-type distributions for patient flow modeling. They interpreted phase-type distributions as a mixture of components (phases) characterized by the severity of patient's illness. Marshall and McClean (2003) extended this idea and developed a model based on Conditional Phase-type distributions combined with a Bayesian Network to be able to include a network of inter-related variables representing causality. In phase-type methods, it is assumed that the process begins in the first phase and may either progress through the phases sequentially or enter an absorbing state (see Fig. 2a). Consequently, these methods cannot be extended to capture patient trajectories, where patients revisit a ward several times or transition from any ward to any other ward, which is a significant feature according to our data. Thomas (1968) and Kao (1972, 1974) proposed a semi-Markov model to predict recovery progress of coronary patients. This can model any hospital system with complicated ward interactions in any direction (See Fig. 2b). Thus, this model has *scalability* and can fully model *ward interactions* but is built only for a "homogenous" mix of patients, i.e. coronary.

Patient heterogeneity is another challenge in CM and, consequently, patient trajectory estimation. To address this challenge, Helm and Van Oyen (2015) partitioned patients into homogeneous clusters with respect to their diagnosis using diagnosis related groups (DRGs). DRGs have been also used by Fetter et al. (1980) for regional planning. Harper (2005) provided a comprehensive review on clustering techniques, including CART, k-means, neural network, etc. that use more patient attributes (e.g., age, sex, diagnosis) to find more



(a) Phase-type model: Patients can transition in a sequential order or leave the system



(b) Semi-Markov model: Any back and forth transition from any ward to any ward is possible

Figure 2: Illustration of patient trajectory models for a hospital system

homogeneous clusters. The main assumption of the DRG and attribute-based methods is that patients who belong to a cluster, follow similar trajectory and thus have similar expected services. However, this is not necessarily true, because although patients in a cluster share similar attributes (e.g., age, sex, diagnosis, etc.), they often have different trajectories as pointed out by Littig and Isken (2007). As an example from the hospital data used for our case study in Sec. 2.5, Fig. 8a shows that although two patients shared the same age, sex, and diagnosis, their trajectories were very different.

In conclusion, the problem of trajectory estimation from a heterogeneous cohort of patients is important. Still, to our knowledge, existing literature lacks in addressing at least one or more challenges among: scalability, ward interaction, and heterogeneity. In the next section, we propose and develop our methodology to solve the problem and address all three challenges.

2.3 Clustering and Scheduling Integrated (CSI) Model for HASC

Fig. 11 provides a high level overview of our methodology. First, historical patient flow data, taken from admit-discharge-transfer (ADT) records, is used to group the patients based on their trajectory using a semi-Markov Mixture (SMM) model based clustering approach. The parameters for the semi-Markov processes that model patients' stochastic location processes (trajectories through the hospital) for each patient cluster are estimated as a part of the

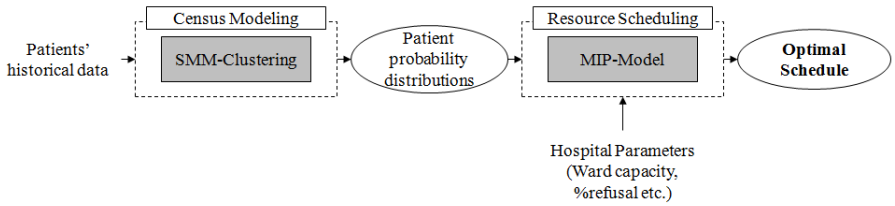


Figure 3: Clustering and Scheduling Integrated (CSI) Model overview

clustering process. These stochastic location processes are then combined with a model of the non-stationary patient arrival process to form a stochastic process (a Poisson arrival-location model or PALM, see Massey and Whitt (1993)) that captures the ward-network census levels. Estimation of this stochastic network census process enables the derivation of three important products for hospital managers: (1) Descriptive: accurate census forecasting, (2) what-if scenarios regarding potential modifications to admission schedules, and (3) Prescriptive: a Mixed Integer Programming (MIP) based elective admission scheduling optimization.

2.3.1 Novel Semi-Markov Mixture (SMM) Clustering for Modeling Patient Trajectories

When a new patient arrives to the hospital, they are initially assigned a bed in a hospital ward. The patient will stay at that ward for a stochastic duration and then may be transferred to another ward or be discharged from the hospital. This process repeats if the patient is transferred to another ward of the hospital.

A general hospital serves a cohort of many different types of patients. Each type of patient requires different services (or resources) which can be identified by the trajectory they follow during their hospital stay. The first task is to identify patient types through clustering. As mentioned previously, conventional clustering methods are not applicable to this problem due to the fact two patients with the same observed attributes often have different trajectories.

To manage the heterogeneous mix of patients in a hospital, we develop a semi-Markov mixture model for clustering based on patient trajectory rather than predefined groupings based on patient attributes. Patients in each cluster are assumed to follow a semi-Markovian

trajectory through the hospital, which has been validated in the literature (e.g. Hancock et. al. 1983). The SMM produces three important products that significantly improve the generality and scalability of our method: (1) appropriate patient groupings based on trajectory, (2) the optimal number of patient types, and (3) accurate trajectories for each patient type. In Sec. 2.5, we show that this approach yields more efficient patient clusters and more accurate trajectory estimations than traditional approaches.

Moreover, to the best of our knowledge, there is no existing approach for developing a semi-Markov mixture model and using it for clustering spatio-temporal data. In this section, we formulate a semi-Markov mixture model, and develop new EM-algorithm steps for clustering and estimating the semi-Markov process parameters.

2.3.1.1 SMM Model Structure

Let $\mathcal{K}$ be the set of unknown patient types, where each patient type's trajectory follows a unique semi-Markov process. The population of patient trajectory data, thus, follows a mixture of an unknown number of semi-Markov processes equal to $|\mathcal{K}|$. Each mixture component, which we call a *cluster* henceforth, has a different semi-Markov process distribution. The first step is to determine the set of clusters ($\simeq \mathcal{K}$) and estimate their corresponding trajectory distributions.

Consider a sample of trajectory data for N patients observed over a maximum time period of length T. Time is measured by discrete units, for example, a day, quarter of day, hour, to be chosen depending on the desired granularity. Thus, the set of possible length of stays is denoted by $\mathcal{T}$, where the maximum of $\mathcal{T}$ corresponds to T. Let $\mathcal{U} = \{\underline{\mathcal{U}}, \bar{\mathcal{U}}\}$ denote the set of all states (wards) where $\underline{\mathcal{U}}$ is the set of all transient states and $\bar{\mathcal{U}}$ is the set of all absorbing states. The first state when the patient enters the system (hospital) is called the initial state and the last state, which is an absorbing state, indicates a patient's end of stay in the form of discharge or death. All the states during the patient's hospital stay are transient states. The set of initial and transient states are the same, as a patient may enter the hospital at any arbitrary location.

A patient's trajectory is denoted by a random variable Y. An observed patient trajectory

with $L-1$ transitions is represented as $\mathbf{y} = (\{u_1, \nu_1\}, \ldots, , \{u_L, \nu_L\})$, where $u_l \in \mathcal{U}$ indicates the visited ward, $\nu_l \in \mathcal{T}$ is the length of stay at the corresponding ward and subscript l, $l = 1, 2, .., L$, indicates the sequence of ward visits (*state* and *ward* are used synonymously in this paper). This model can capture general network behavior, as there is no restriction on the number of times a patient can visit any particular ward.

We formulate the problem by defining a set of parameters, $\mathbf{\Theta} = \{\Theta^{(k)}\}$, $k \in \mathcal{K}$. Each $\Theta^{(k)}$ is comprised of the mixture weight, $\pi^{(k)}$, and semi-Markov process parameters, $\{\rho^{(k)}, P^{(k)}, H^{(k)}\}$, for the k-th mixture. The mixture weight, $\pi^{(k)}$, denotes the probability of a patient belonging to cluster k. Letting Z be a hidden variable representing the cluster index, then the mixture weight can be expressed as, $\pi^{(k)} = p_{\mathbf{\Theta}}(Z = k)$. Also, $\sum_{k \in \mathcal{K}} \pi^{(k)} = 1$.

Of the remaining mixture parameters, $\boldsymbol{\rho}^{(k)} = \{\rho_i^{(k)}\}$, $i \in \mathcal{U}$, denotes the initial state probability. It can be expressed as $\rho_i^{(k)} = p_{\mathbf{\Theta}}(u_1 = u | Z = k)$, the probability of the first state of a patient trajectory from cluster k being at ward u. The matrix $\mathbf{P}^{(k)} = [P_{ij}]$, $i, j \in \mathcal{U}$, is the transition probability matrix, where $P_{ij}^{(k)} = p_{\mathbf{\Theta}}(u_l = i | u_{l-1} = j, Z = k)$, the probability of transitioning from ward j to i for a patient in cluster k. Finally, $\mathbf{H}^{(k)} = [H_{ij}^{(k)}(\nu)]$, $i, j \in \mathcal{U}$, $\nu \in \mathcal{T}$, is a three-dimensional tensor representing the holding mass distribution, where $H_{ij}^{(k)}(\nu) = p_{\mathbf{\Theta}}(\nu_l = \nu | u_l = i, u_{l-1} = j, Z = k)$ gives the probability of a patient in cluster k spending ν time units in ward i, after transitioning to ward j from i. As $\{\rho^{(k)}, P^{(k)}, H^{(k)}\}$ are probability distributions, the following hold:

$$\sum_{i \in \mathcal{U}} \rho_i^{(k)} = 1, \sum_{j \in \mathcal{U}} P_{ij}^{(k)} = 1, \textit{and} \sum_{\nu \in \mathcal{T}} H_{ij}^{(k)}(\nu) = 1 \tag{1}$$

Using this parameterization, we represent the conditional probability of any patient i's trajectory, $\mathbf{y}^{(i)}$, given it is generated by cluster k, in Eq. 2. The first part of the equation is the initial state probability. The terms inside the product is the transition probability times the holding time probability corresponding to the transition the patient made, and the amount of time the patient spent at the ward before transitioning.

$$
\begin{aligned}
p_{\boldsymbol{\Theta}}(Y = \mathbf{y}^{(i)} | Z = k) &= p(u_1^{(i)} | \rho^{(k)}) \prod_{l=2}^{L} p(u_l^{(i)} | u_{l-1}^{(i)}; \mathbf{P}^{(k)}) p(\nu_l^{(i)} | u_l^{(i)}, u_{l-1}^{i}; \mathbf{H}^{(k)}) \\
&= \rho^{(k)}_{u_1^{(i)}} \prod_{l=2}^{L} \left\{ P^{(k)}_{u_{l-1}^{(i)}, u_l^{(i)}} \cdot H^{(k)}_{u_{l-1}^{(i)}, u_l^{(i)}}(\nu_l^{(i)}) \right\}. \qquad (2)
\end{aligned}
$$

Consequently, by considering the probability of belonging to each cluster, k, the probability distribution function (pdf) of the SMM model with $\mathcal{K}$ components is written as

$$
\begin{aligned}
p(\mathbf{y}^{(i)} | \boldsymbol{\Theta}) &= \sum_{k \in \mathcal{K}} p_{\boldsymbol{\Theta}}(Z^{(i)} = k) p_{\boldsymbol{\Theta}}(\mathbf{y}^{(i)} | Z^{(i)} = k) \\
&= \sum_{k \in \mathcal{K}} \pi^{(k)} \left[\rho^{(k)}_{u_1^{(i)}} \prod_{l=2}^{L} \left\{ P^{(k)}_{u_{l-1}^{(i)}, u_l^{(i)}} \cdot H^{(k)}_{u_{l-1}^{(i)}, u_l^{(i)}}(\nu_l^{(i)}) \right\} \right]. \qquad (3)
\end{aligned}
$$

Given an i.i.d. sample of N patient trajectories, $\mathbf{Y} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}\}$, the likelihood function is, thus, given by

$$
p_{\boldsymbol{\Theta}}(\mathbf{Y}) = \prod_{i=1}^{N} p(\mathbf{y}^{(i)} | \boldsymbol{\Theta}) = \prod_{i=1}^{N} \sum_{k \in \mathcal{K}} \pi^{(k)} \left[\rho^{(k)}_{u_1^{(i)}} \prod_{l=2}^{L} \left\{ P^{(k)}_{u_{l-1}^{(i)}, u_l^{(i)}} \cdot H^{(k)}_{u_{l-1}^{(i)}, u_l^{(i)}}(\nu_l^{(i)}) \right\} \right]. \qquad (4)
$$

The parameters of the SMM mixture model, $\boldsymbol{\Theta}$, can be estimated by maximizing the (log)likelihood function in Eq. 4. However, if there is no observed transition between any two states or no instance of any particular length of stay, the likelihood function becomes zero. To avoid this issue, we use a Bayesian approach that assigns very small prior probabilities to all model parameters, denoted by $p(\boldsymbol{\Theta})$. Thus, according to Bayes rule, the posterior probability for $\boldsymbol{\Theta}$ can be expressed as $p(\boldsymbol{\Theta}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta})}{p(\mathbf{Y})}$. Since $p(\mathbf{Y})$ is independent of $\boldsymbol{\Theta}$, it suffices to maximize the non-normalized posterior log-likelihood in Eq. 5 to obtain the optimal $\boldsymbol{\Theta}^*$, also known as the *maximum a posteriori* (MAP) estimates of $\boldsymbol{\Theta}$.

$$
\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \log \{ p(\mathbf{Y}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}) \} \qquad (5)
$$

The optimization problem in Eq. 5 does not have a closed-form solution. Further, the non-normalized posterior log-likelihood is a non-convex function and thus, the optimization problem in Eq. 5 cannot be solved using standard convex optimization methods. As a result, we develop an iterative *expectation-maximization* (EM) procedure in the following section to obtain the parameter estimates.

2.3.1.2 Parameter Estimation via Expectation-Maximization (EM) Algorithm

An Expectation-Maximization (EM) algorithm is an effective approach for learning maximum likelihood or maximum a posteriori (MAP) estimates, where the likelihood is a function of unobserved latent variables (in our case, Z). It is an iterative approach comprising of an Expectation (E-step) and Maximization (M-step) in each iteration. In the E-step of any iteration p, we obtain a lower bound on the objective function by taking its expectation at the current parameter estimate, $\mathbf{\Theta}^{(p)}$. Then, in the M-step, we re-estimate the parameters (update), to obtain $\mathbf{\Theta}^{(p+1)}$, that maximizes the expectation from E-step. This procedure results in an increase of the likelihood function, guaranteed convergence under some weak regularity conditions that are satisfied in most practical situations (Wu 1983). The specific EM algorithm we develop for the SMM mixture model is as follows:

E-step

We find the expected value of the maximum a posteriori function in Eq. 5 with respect to the current parameter estimate, $\mathbf{\Theta}^{(p)}$, denoted by $Q(\mathbf{\Theta}|\mathbf{\Theta}^{(p)})$ in Eq. 6.

$$Q(\mathbf{\Theta}|\mathbf{\Theta}^{(p)}) = \mathbb{E}_{\mathbf{\Theta}^{(p)}}[\log(p(\mathbf{Y}|\mathbf{\Theta})p(\mathbf{\Theta})] \tag{6}$$

For a simpler expression of the Q function in Eq. 6, we define a *membership* probability distribution. Membership probability, denoted by Ω_{ik}, is the probability of observing any patient i's trajectory, $\mathbf{y}^{(i)}$, generated by cluster k, given parameters $\mathbf{\Theta}$ (see Eq. 7).

$$\begin{aligned} \Omega_{ik}(\mathbf{\Theta}) &= \frac{\pi^{(k)} p_{\mathbf{\Theta}}(\mathbf{y}^{(i)}|Z^{(i)}=k)}{\sum_{k' \in \mathcal{K}} \pi^{(k')} p_{\mathbf{\Theta}}(\mathbf{y}^{(i)}|Z^{(i)}=k')} \\ \Omega(\mathbf{\Theta}) &= [\Omega_{ik}(\mathbf{\Theta})], \quad i = 1, \ldots, N,\ k \in \mathcal{K} \end{aligned} \tag{7}$$

The Q function, can thus be expressed as,

$$\begin{aligned} Q(\mathbf{\Theta}|\mathbf{\Theta}^{(p)}) &= \mathbb{E}_{\mathbf{\Theta}^{(p)}}\left[\log(p(\mathbf{Y}|\mathbf{\Theta})p(\mathbf{\Theta})\right] \\ &= \sum_{i=1}^{N} \sum_{k \in \mathcal{K}} \Omega_{ik}(\mathbf{\Theta}^{(p)}) \log\left[\pi^{(k)} p_{\mathbf{\Theta}}(\mathbf{y}^{(i)}|Z^{(i)}=k)\right] + \log p(\mathbf{\Theta}) \end{aligned} \tag{8}$$

M-step

In the *maximization* step, the parameters that maximize the Q function are estimated. The updated parameters are, thus,

$$\mathbf{\Theta}^{(p+1)} = \arg\max_{\Theta} \left\{ Q(\mathbf{\Theta}|\mathbf{\Theta}^{(p)}) \right\} \tag{9}$$

To solve Eq. 9, we will estimate the posterior of the parameters using a Dirichlet prior probability distribution for $\mathbf{\Theta}$, $p(\mathbf{\Theta})$. The Dirichlet distribution is chosen because 1) the parameters of a first-order semi-Markov mixture are in the form of multinomial probabilities, which are suitably represented by Dirichlet distribution, and 2) the conjugate of Dirichlet is also a Dirichlet distribution, thus posterior computation is straightforward.

For any set of multinomial parameters, $x = (x_1, \ldots, x_m)$, such that $\sum_{i=1}^{m} x_i = 1$, $0 \leq x_i \leq 1$, a Dirichlet distribution is given by,

$$p(x_1, \ldots, x_m | a_1, \ldots, a_m) = \frac{1}{B(a)} \prod_{i=1}^{m} x_i^{a_i - 1} \tag{10}$$

where a_i's are hyperparameters for x, and $B(a) = \dfrac{\prod_{i=1}^{m} \Gamma(a_i)}{\Gamma\left(\sum_{i=1}^{m} a_i\right)}$, a constant factor for the Dirichlet probability distribution function.

Using the prior probability distributions, assumption of independence of parameters, and plugging Eq. 2 into Eq. 8, we obtain the posterior distributions. We show in Online Appendix A, the posterior distributions are Dirichlet, and how to update parameters to maximize Eq. 8.

The developed EM algorithm procedure is summarized in Table 1. We, first, initialize each parameter. Then, for any iteration p, we compute the membership probabilities, $\Omega(\mathbf{\Theta}^{(p)})$, given in Eq. 7 (E-step), followed by updating each parameter (M-step). Thereafter, we compute the Q function. The E- and M-step are repeated until there is no significant relative change in the Q function, to obtain the optimal estimates, $\hat{\mathbf{\Theta}} = \{\hat{\Theta}^{(k)}\}, k \in \mathcal{K}$. As also mentioned before, the EM algorithm guarantees convergence to a local *maxima*. Different initializations can be used to search for the global *maxima*.

Table 1: Semi-Markov model (SMM) based clustering algorithm.

Given a value for number of clusters $\mathcal{K}$ and trajectory data of size N.

1 Initialization: Randomly assign a cluster to each data point and compute the initial values of $\mathbf{\Theta}^{(0)} = \{\Theta^{(k)(0)}\}$ for $k \in \mathcal{K}$ based on it.

For iteration, $p = 0, 1, 2, \ldots$

2.1 Compute the membership $\Omega^{(p)}(\mathbf{\Theta}^{(p)}) = \left[\Omega_{ik}(\mathbf{\Theta}^{(p)})\right]$ for $i = 1, \ldots N,\ k \in \mathcal{K}$ using Eq. 7 by plugging in the values of $\mathbf{\Theta}^{(p)}$ in Eq. 2.

2.2 The values of the parameters in $\Theta^{(k)} = \left\{\pi^{(k)}, \boldsymbol{\rho}^{(k)}, \mathbf{P}^{(k)}, \mathbf{H}^{(k)}\right\}$ is updated as per following equations (from Online Appendix A)

$$\pi^{(k)(p+1)} = \frac{\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) + a_{\pi}^{(k)}}{\sum_{k' \in \mathcal{K}} \left[\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) + a_{\pi}^{(k')}\right]} \text{ for } k \in \mathcal{K}.$$

$$\rho_{u}^{(k)(p+1)} = \frac{\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) \kappa(u_1^{(i)}, u) + a_{\rho,u}^{(k)}}{\sum_{u' \in \mathcal{U}} \left[\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) \kappa(u_1^{(i)}, u') + a_{\rho,u'}^{(k)}\right]} \text{ for } k \in \mathcal{K} \text{ and } u \in \mathcal{U}.$$ $\kappa(x, y)$is an indicator function equal to 1 if $x = y$.

$$P_{uj}^{(k)(p+1)} = \frac{\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) \bar{\kappa}_{uj}(\mathbf{y}^{(i)}) + a_{P,uj}^{(k)}}{\sum_{j' \in \mathcal{U}} \left[\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) \bar{\kappa}_{uj'}(\mathbf{y}^{(i)}) + a_{P,uj'}^{(k)}\right]} \text{ for } k \in \mathcal{K} \text{ and } (u, j) \in \mathcal{U}.$$ $\bar{\kappa}_{uj}(\mathbf{y}^{(i)})$ is the count function equal to the number of times transition was made from state u to j in trajectory $\mathbf{y}^{(i)}$.

$$H_{uj}^{(k)}(\nu)^{(p+1)} = \frac{\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) \tilde{\kappa}_{uj,\nu}(\mathbf{y}^{(i)}) + a_{H,uj}^{(k)}(\nu)}{\sum_{\nu' \in \mathcal{T}} \left[\sum_{i=1}^{N} \Omega_{ik}(\mathbf{\Theta}^{(p)}) \tilde{\kappa}_{uj,\nu'}(\mathbf{y}^{(i)}) + a_{H,uj}^{(k)}(\nu')\right]} \text{ for}$$

$k \in \mathcal{K}$; $(u, j) \in \mathcal{U}$ and $\nu \in \mathcal{T}$. $\tilde{\kappa}_{uj,\nu}(\mathbf{y}^{(i)})$ is the count function equal to the number of times transition was made from state u to j, in trajectory $\mathbf{y}^{(i)}$, when length of stay at state u was ν time units.

$a \in (0, 1)$ denotes the hyperparameter of the corresponding Dirichlet prior distribution.

3 Compute $Q^{(p+1)}$ using Eq. 8.

4 Repeat 2-3 until convergence, to obtain the optimal estimates, $\hat{\mathbf{\Theta}} = \{\hat{\Theta}^{(k)}\}, k \in \mathcal{K}$.

2.3.1.3 Determining the number of clusters

To determine the appropriate number of clusters, we estimate the SMM model and compute the Q function, which is analogous to the likelihood. We then increase the number of clusters, $|\mathcal{K}|$, stopping when there is no obvious change in the Q function (popularly known as the *elbow* method). To ensure that redundant clusters are not created we perform pairwise hypothesis tests with controlled type-I error for the identified clusters. We use the Chi-square hypothesis test developed by Billingsley (1961a, 1961b) for comparing transition probabilities and Kolmogorov-Smirnov for comparing the distributions on the initial state and the holding time. We merge any clusters that are found similar by these tests and then perform the tests again in iterative fashion until no redundant clusters are detected. A similar approach for removing redundant clusters was used by Weiss et. al (1982).

2.3.1.4 Trajectory estimation for each cluster

After parameter estimation, the next step is to estimate the patient trajectory distributions which are characterized by the visited wards and length of stay at each ward. Using the selected number of clusters and corresponding semi-Markov process estimates from the previous step, we compute the probability distribution of patient trajectory, denoted by $\Gamma(d) = [\gamma_u^{(k)}(d)];\ \ u \in \mathcal{U}, k \in \mathcal{K}$ and $d = 1, 2, \ldots$, where $\gamma_u^{(k)}(d)$ is the probability that a patient of cluster k is in ward u after d days (we use a day as a time unit, ν). This distribution is one of the key inputs to the scheduling optimization.

To estimate $\mathbf{\Gamma}(d)$ we use interval transition probabilities, $\Phi^{(k)} = [\phi_{ij}^{(k)}(d)];\ \ (i,j) \in \mathcal{U}, k \in \mathcal{K}$ and $d = 1, 2, \ldots$, where $\phi_{ij}^{(k)}(d)$ is the probability that a patient in cluster k is in ward j on day d, given that the patient entered the hospital in ward i. Recalling that, for a type k patient, $H_{ij}^{(k)}(d)$ is the holding time probability distribution in ward i before transitioning to ward j and $P_{ij}^{(k)}$ is the probability of transitioning from ward i to j, then $\phi_{ij}^{(k)}(d)$ is computed as

$$\phi_{ij}^{(k)}(d) = P_{ij}^{(k)} H_{ij}^{(k)}(d) + \delta_{ij} \sum_{l \in \mathcal{U} \setminus \{i\}} \sum_{d'=d+1}^{\infty} P_{il}^{(k)} H_{il}^{(k)}(d') + \sum_{l \in \underline{\mathcal{U}} \setminus \{j\}} \sum_{d'=1}^{d} P_{il}^{(k)} H_{il}^{(k)}(d') \phi_{lj}^{(k)}(d - d'), \tag{11}$$

where $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$ and $\phi_{ij}^{(k)}(0) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$. A patient starting in state i can be in state j on day d either if the patient stays in ward i for d days before transitioning to ward j (the first term of Eq. 11), or $i = j$ and they never left i during the period $[0, d]$ (the second term of Eq. 11), or the patient left i at least once and finally reached j by day d (the third term of Eq. 11). Consequently, $\gamma_u^{(k)}(d)$ can be expressed as sum-product of all possible initial states to ward u (Eq. 12).

$$\gamma_u^{(k)}(d) = \sum_{i \in \mathcal{U}} \rho_i^{(k)} \phi_{iu}^{(k)}(d) \tag{12}$$

γ computed here is used as an input to the *scheduling* model explained in next section. The semi-Markov process estimates, $\hat{\Theta}^{(k)}$, can be used for finding the length-of-stay distribution of each patient type as well as the expected mean length of stay in each ward and its variance. Equations to compute these are given in the following subsection as they may be useful for other research objectives or purposes.

2.3.1.5 Computing Patient length-of-stay distributions

Length-of-stay in a ward (V).

For a patient of type k, we estimate the expected days spent by the patient in each ward using the indicator function approach on the interval transition probability $\Phi^{(k)}$. Let $\bar{V}^{(k)} = \left[\bar{v}_{ij}^{(k)}\right]$; $i, j \in \mathcal{U}$; $k \in \mathcal{K}$, where $v_{ij}^{(k)}$ denotes the number of days the patient will spend in j given their initial state was in ward i. The mean of $v_{ij}^{(k)}$ can be computed using Eq. 13 given below.

$$\bar{v}_{ij}^{(k)} = \sum_{d=1}^{\infty} \phi_{ij}^{(k)}(d) \tag{13}$$

The second moment of $v_{ij}^{(k)}$is given by

$$\bar{v}_{ij}^{2(k)} \quad = \quad \bar{v}_{ij}^{(k)}(2\bar{v}_{ij}^{(k)} - 1) \tag{14}$$

Thus, the variance of the days spent by a patient in a state can be given by

$$\check{v}_{ij}^{(k)} \quad = \quad \bar{v}_{ij}^{2(k)} - (\bar{v}_{ij}^{(k)})^2 \quad \forall i, j \in \mathcal{U}. \tag{15}$$

Total hospital length-of-stay (LOS).

To get the distribution for LOS for entire hospital stay, we will use first-passage-time probabilities, denoted by F. $F^{(k)} = \left[f_{ij}^{(k)}(d)\right]$, $i, j \in \mathcal{U}$; $\nu = 1, 2, \ldots$; $k \in \mathcal{K}$, where $f_{ij}^{(k)}(d)$ is the probability that the first passage from state i to j will take exactly d days for patients of type k. This event can occur if a patient makes a direct transition from i to j on day d, or the patient transitions to any other state l on any day before d and then takes first passage from l to j. The second component is recursive and thus takes into account any number of transitions between any states (except the absorbing state) to reach from i to j in d days.

$$f_{ij}^{(k)}(d) \quad = P_{ij}^{(k)} H_{ij}^{(k)}(d) + \textstyle\sum_{l \in \underline{\mathcal{U}} \setminus \{j\}} \sum_{d'=1}^{d} P_{il}^{(k)} H_{il}^{(k)}(d') f_{lj}^{(k)}(d - d'). \tag{16}$$

Using $f_{iA}^{(k)}$, where $A \in \bar{\mathcal{U}}$, and the initial state probability ρ we can get the distribution for LOS. $f_{iA}^{(k)}$ denotes the first-passage-probability for a patient's flow from any initial state i to a discharge state A. If the initial state is unknown then we use Eq. 17 . Otherwise if the initial state is known, say l, then the distribution is given by $f_{iA}^{(k)}$ itself.

$$L^{(k)}(d) \quad = \quad \sum_{i \in \underline{\mathcal{U}}} \rho_i^{(k)} f_{iA}^{(k)}(d) \quad d = 1, 2, \ldots \tag{17}$$

2.3.1.6 Elective and emergency inpatient census model

In this section, we describe how we integrate the semi-Markov stochastic location processes generated from our SMM method with different arrival processes to create a stochastic

ward census process. This section, as well as Sec. 2.3.2, present an elective scheduling optimization approach focused on hospital patient throughput (i.e. admission volume) and congestion (e.g. bed block, off-ward placement of patients) that is based on the work by Helm and Van Oyen (2015). The purpose of these sections is to provide relevant background for possible applications of our CM method in the hospital census forecasting industry as described by our industry co-author. We use the aforementioned optimization approach as a proof of concept to test the value of our improved CM method. These sections are, therefore, intentionally brief and not intended to present new research in the area of resource optimization. The focus of this paper is on the development and analysis of a CM method (patient type clustering and trajectory estimation) that is designed to integrate with existing optimization approaches such as the one presented herein.

There are two broad categories of patients that a hospital serves, elective (EL) and emergency (EM). In developing our census model we separate the two because in the optimization in Sec. 2.3.2, emergency arrivals are considered uncontrollable while the scheduled elective arrivals become the primary decision variable. To integrate our SMM clustering and trajectory estimates with the optimization as well as the what-if scenarios of interest to the industry, we run the clustering method on EL and EM patients separately. Hence each stream, EL and EM, will have its own set of patient types, $\mathcal{K}$, with their own trajectories determined by our SMM.

As explained in previous sections (2.3.1.1-2.3.1.3), we cluster the EL patients into homogeneous groups with similar trajectories. Trajectory estimates, one for each patient *type* (cluster), are computed using Eqs. 11 and 12. Combining the EL arrival pattern with the semi-Markov trajectory distributions for each patient type, discussed in Sec. 2.3.1.4, creates a stochastic census process that can be used to calculate the distribution on patient demand for beds at each ward at any time, t. The exact distribution depends on the arrival process.

For EL admissions we consider a deterministic arrival process, which, when combined with the semi-Markovian patient trajectories, yields a Poisson-Binomial distribution on bed demand at fixed time point t. The deterministic assumption is an approximation of reality, but has been widely used in the literature due to the fact that elective arrivals are controlled

and scheduled in advance. Therefore it is (1) theoretically possible to achieve close to a deterministic arrival stream, (2) it is highly beneficial to patient flow for hospital managers to work toward a deterministic elective arrival stream and should be a management priority, (3) deviations from the deterministic arrivals can be incorporated for certain distributions and approximated for others — particularly if the variance of the arrival pattern can be adequately approximated as a linear function of the mean.

The uncontrollable arrival of emergency patients are taken into account by assuming that their arrival pattern follows a non-homogeneous Poisson process that varies by day of week. Combining these Poisson arrivals with the semi-Markov stochastic location processes yields a Poisson-arrival-location model (PALM) of emergency census, (see Massey and Whitt (1993) for more details). One feature of a PALM model is that the distribution on demand for beds in any ward for fixed t follows a Poisson distribution.

Having defined the distribution on demand for beds for emergency and elective patients, we now briefly describe an optimization model from the literature (Helm and Van Oyen (2015)) that is subsequently used to demonstrate the importance of a rigorous patient trajectory estimation procedure. We designed our estimation approach to integrate with optimization and what-if scenarios, with this particular optimization being used as a proof of concept that (1) our method integrates well with current optimization approaches, and (2) our method significantly improves the outcome of the optimization when compared with traditional approaches proposed for use with these types of models.

2.3.2 Resource Scheduling (RS) MIP model for Elective Admission Scheduling

As mentioned above (Sec. 2.3.1.6), the schedule of EL admissions can be controlled while EM arrivals are not in hospital's control. The RS model we use as proof of concept integrates both EM and EL census models to capture metrics such as blocking and off-ward placement of patients. The two common objectives from the literature that we focus on are: 1) maximizing the number of elective admissions while constraining congestion metrics and 2) minimizing the congestion (e.g. blocking) while maintaining patient throughput. From a management perspective, the first objective allows for increased revenue, while the second

objective provides better access and consequently better outcomes for patients. For ease of reference, we present this optimization model in Online Appendix B.

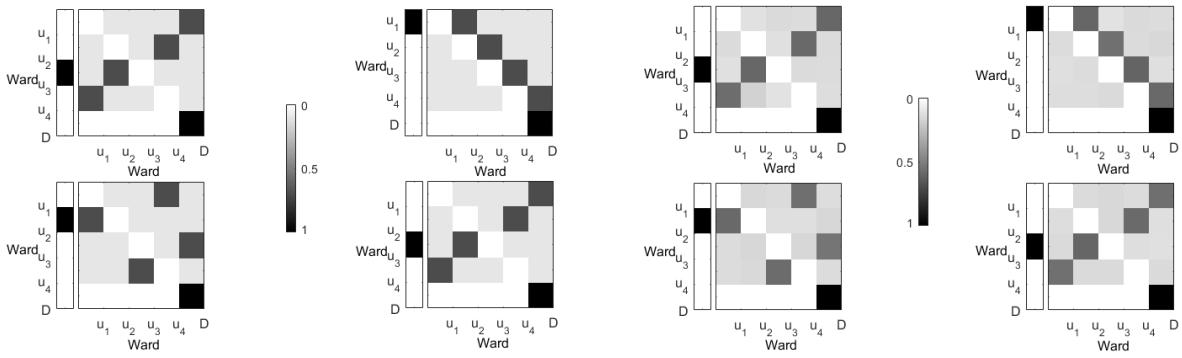
This concludes the presentation of our CSI approach that develops an optimal design of patient types and trajectory estimations for integration into an inpatient admission scheduling optimization model. In the next section we develop a simulation to validate the accuracy of the SMM approach for patient clustering and trajectory estimation and to determine the impact of the SMM on optimal solutions to the MIP model.

2.4 SMM Validation and Impact on Optimal Scheduling Solutions

In this simulation study, we considered a hospital system with four transient states (analogous to wards), $\underline{\mathcal{U}} = \{u_1, \ldots, u_4\}$ and one absorbing state (analogous to discharge or death) $\bar{\mathcal{U}} = \{D\}$. Flow sequences for 1000 patients were generated from four different semi-Markov models (corresponding to four different patient types), denoted by $C_s^{(1)}, \ldots, C_s^{(4)}$. As two clusters could be different in $\mathbf{P}$, $\mathbf{H}$, and/or both, we used the following setting that covers all possible scenarios. In the data generating model, $C_s^{(1)}$ and $C_s^{(2)}$ have different $\mathbf{P}$ but same $\mathbf{H}$, $C_s^{(3)}$ and $C_s^{(4)}$ have same $\mathbf{P}$ and different $\mathbf{H}$, while $C_s^{(2)}$ and $C_s^{(3)}$ have different $\mathbf{P}$ and $\mathbf{H}$. A pictorial representation of the transition probability matrix combined with the initial state probability is shown in Fig. 4a. In these plots, the darker the color, the higher the probability. The component mixture weights, π, of the four clusters are $\{0.17, 0.33, 0.25, 0.25\}$ respectively. Additionally, the assignment probabilities in the generating distributions were set less than 0.7 to ensure that the simulation output would be similar to that of a general hospital scenario.

2.4.1 Evaluating the accuracy of the SMM method

The proposed SMM mixture model was applied to the generated data for various numbers of clusters and the Q function was plotted against the number of clusters, $|\mathcal{K}|$ as shown in Fig. 5. As can be seen from the figure, the absolute slope of the Q estimates significantly drops at $|\mathcal{K}| = 4$ with estimated $\hat{\pi} = \{0.169, 0.332, 0.253, 0.246\}$, which indicates that the true number of clusters and mixture weights were accurately identified by the SMM estimation model. No similar clusters were found by the pairwise hypothesis tests discussed



(a) Data generating transition probabilities

(b) Estimated cluster's transition probabilities

Figure 4: Pictorial representation of transition probabilities as a gray scale heat-map; with higher intensity of gray for higher probability. The heat-map for generating and estimated cluster transition probabilities are shown side-by-side for visual comparison.

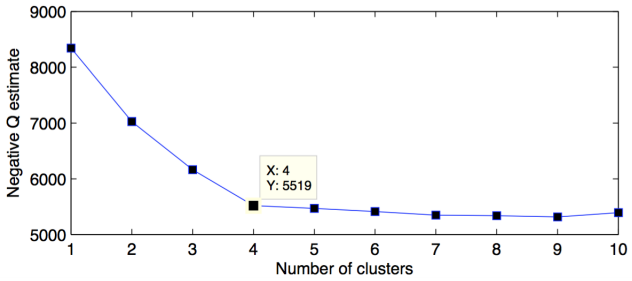


Figure 5: Q function estimates against number of clusters for the simulated data. The improvement in Q estimate becomes insignificant after 4 clusters.

in Sec. 2.3.1.3. To assess the accuracy of the estimated parameters $\hat{\pi}^{(j)}, \hat{\rho}^{(j)}, \hat{\mathbf{P}}^{(j)}, \hat{\mathbf{H}}^{(j)}$ for each of the estimated clusters, we compared them with the parameters of the data generating model. The pictorial representation of estimated and true probabilities is shown in Figures 4a and 4b, respectively. The high degree of similarity between the plots in these two figures implies a highly accurate estimation of initial state and transition probabilities. Additionally, we conducted Chi-square and Kolmogorov-Smirnov tests to verify the equality of estimated and true parameters. The p-values of these tests reported in Table 2 are all greater than 0.05, indicating that the equality of estimated and true parameters (null hypothesis) cannot be rejected, i.e., they are *statistically* the same at 95% confidence level. In summary, all the results show a clear one-to-one mapping between estimated and generating (true) cluster parameters, demonstrating the effectiveness of our SMM clustering model at identifying the underlying parameters of the patient flow system.

Table 2: p-values for matched cluster parameters. Higher p-values compared to the significance level indicates that the two compared distributions were same.

i	j	ρ	**P**	**H**
1	2	0.99	0.54	0.99
2	3	0.99	0.17	0.83
3	4	0.99	0.23	0.98
4	1	0.99	0.29	0.99

As discussed in Sec. 2.3.1.4 and 2.3.1.6, the MIP-scheduling model for RS uses the estimated trajectory distribution of each patient type as an input. We, now study the effect of the SMM estimates' accuracy on the value of the optimal MIP solution. The MIP scheduling model will give the true 'optimal' schedule, if it is given the true trajectory distribution of all patient types. From the data generating model, we know the true number of patient types and their underlying trajectory distributions. This is used to obtain the optimal schedule and corresponding utilization (i.e., the scheduled ward workload over ward maximum capacity). We use this optimal schedule as the baseline and compare it with the MIP solutions obtained from estimated trajectories. Specifically, we compare the increase in elective admissions and resource utilization under the optimal schedule when using the *true* trajectories as input against that of, a) our CSI approach that utilizes the trajectory estimates from SMM in the MIP model, and b) traditional approach that uses empirical trajectory estimates to find the MIP solutions.

2.4.2 Estimation using the traditional approach

In the traditional approach, patient attributes including age, gender and diagnosis are used to identify patient types using a conventional clustering method (e.g. k-means clustering). Then, the trajectory distribution is empirically estimated for each patient type and these estimates are used in the MIP model. Data generation and estimation for the traditional approach is described in the following Sec. 2.4.2.1 and Sec. 2.4.2.2. First, attributes were assigned to each patient according to the method presented in Sec. 2.4.2.1. Next, we used k-means clustering, with four clusters, to perform the attribute-based clustering. This should yield a conservative estimate of the benefits of our method, since we are giving the traditional method the correct number of clusters to begin with. In general, the traditional method

will not choose the same number of clusters as the SMM method. For example, in our case study of Sec. 2.5 our SMM found 32 clusters, where our industry co-author noted that, using age and DRG the traditional method would have found over 100 clusters. Finally, the traditional approach from the literature for empirical estimation of patient trajectories is applied to each cluster, the method for which is summarized in Sec. 2.4.2.2.

2.4.2.1 Assigning Patient Attributes to Clusters

In the data generation step, after patient trajectories were generated from four semi-Markov processes, three attributes, viz. age, gender and diagnosis (with three diagnoses being D1, D2, D3), were assigned to the patients such that any attribute triplet has the possibility of being in any cluster; e.g. a 30 year old female with diagnosis D1 could potentially be from any of the four clusters. This resembles real-world challenges involved in patient trajectory estimation by simulating the fact that two patients with the same attributes may have different trajectories; i.e. the attributes are not adequately capturing patient heterogeneity. In practice, patient attributes are capable of capturing some of the patient heterogeneity so we ensure that clusters contain patients whose attributes are mostly similar by adhering to a near-Pareto principle (see the three attribute generating tables in Online Appendix C). That is, clusters are composed mostly of similar patient attributes with a mix of patients who have different attributes. This distribution of attributes is designed to be fair to the traditional approach and capture the reality that attributes do have differentiating power, but cannot completely specify a patients likely trajectory. More details are provided in Online Appendix C.

2.4.2.2 Empirical Estimation of Patient Trajectories

Once the clusters have been formed using k-means clustering, the trajectory distribution is computed for each cluster independently by normalizing the frequency of transitions of patients between wards as follows:

1. $\rho_j^{(k)} = \frac{\sum_{i=1}^{N} \kappa(\mathbf{y}_1^{(i)}, j)}{\sum_{j' \in \mathcal{U}} \left[\sum_{i=1}^{N} \kappa(\mathbf{y}_1^{(i)}, j')\right]}$ for $k \in \mathcal{K}$ and $j \in \mathcal{U}$.

2. $P_{jl}^{(k)} = \frac{\sum_{i=1}^{N} \bar{\kappa}_{jl}(\mathbf{y}^{(i)})}{\sum_{l' \in \mathcal{U}} \left[\sum_{i=1}^{N} \bar{\kappa}_{jl'}(\mathbf{y}^{(i)})\right]}$ for $k \in \mathcal{K}$; $j \in \mathcal{U}$ and $l \in \mathcal{U}$.

3. $H_{jl}^{(k)}(\nu) = \frac{\sum_{i=1}^{N} \tilde{\kappa}_{jl,\nu}(\mathbf{y}^{(i)})}{\sum_{\nu' \in \mathcal{T}} \left[\sum_{i=1}^{N} \tilde{\kappa}_{jl,\nu'}(\mathbf{y}^{(i)})\right]}$ for $k \in \mathcal{K}, j \in \mathcal{U}, l \in \mathcal{U}$ and $\nu \in \mathcal{T}$.

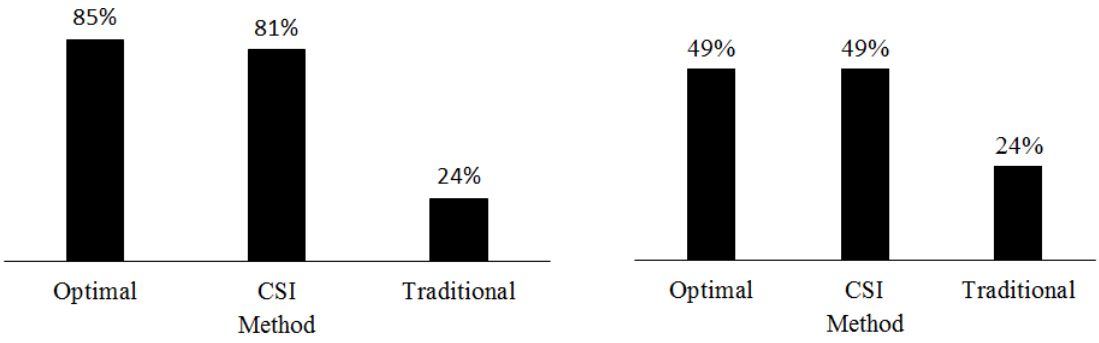
2.4.3 Analyzing the value of the SMM approach to Census Modeling

In this section, we solve the *maximum elective admission* formulation presented in Sec. 2.3.2 and Online Appendix B. The figure shows the percentage improvement in the optimal model setup, the proposed CSI method and the traditional method for two important performance metrics, viz. elective patient admissions and ward utilization. As can be seen, the results of CSI are very close to the optimum while the traditional method for patient typing and trajectory estimation provides significantly less benefit in both performance metrics. As can be seen in Fig. 6a, the improvement in elective patient admissions for the CSI was 81%, which is close to the optimum (85%), while the improvement only reached 24% with the traditional method. Also, for ward utilization, the CSI performed as well as the optimal solution with an increase of 49%, while the traditional method shows an increase of only 24% (Fig. 6b). This improvement can be attributed to the accurate estimation of the patient types and their trajectories which leads to better understanding of their flow between wards and corresponding resource requirements for the MIP model.

The results of the simulation study indicate that our proposed CSI approach not only accurately estimates the patient types and their trajectories for CM, but it also yields a schedule in RS that is very near the true optimal, and significantly outperforms existing HASC methods. This study is the first, to our knowledge, to quantify the impact of estimation approach on elective inpatient optimization solutions, and further demonstrates the importance of effective estimation techniques (e.g. our SMM method) on patient flow optimization.

2.5 Case study on real hospital data

In this section, we will study the impact of our integrated framework (CSI) on hospital resource optimization at a partner hospital, and as a holistic tool for HASC problem. In particular, we focus on validating the trajectory estimation and RS models, as forecasting arrival streams is out of the scope of this paper. Hence, we take the arrival stream as given



(a) Percentage increase in elective (EL) patients admission

(b) Percentage increase in ward utilization

Figure 6: Service level improvements for the three model setups: optimal, CSI and traditional. The improvement is measured for two metrics: number of elective patient admissions and the ward workload. The results of CSI is very close to the optimal.

in order to independently evaluate the accuracy and impact of trajectory estimation on the HASC problem.

We use historical data of patient admission and transitions in a hospital with 55 wards including surgical, ICU/CCU, medicine, neurology, oncology, obstetrics, etc. This system is a good example of a complex hospital system with general ward network structure, transfers and blocking/congestion. We obtained one year of data from 2012, with about 11,000 patients who stayed at least one night in the hospital. The data set includes the patient trajectory data, length-of-stay at each ward, and patient attribute data, for e.g. age, sex, diagnosis, etc. The ratio of elective and emergency patients in the data is almost equal. Patients have an average of 4.1 transfers before leaving the hospital through discharge/death. We compare the performance of the CSI model with that of the traditional estimation model to demonstrate the impact of our approach.

We begin with the CM step by applying SMM-based clustering on patient trajectory data to identify patient types. From Fig. 7, we can infer that there are 32 patient types. Again, no redundant clusters were found from pairwise hypothesis testing. The trajectory probability distributions for each of these patient types are computed using Eq. 12. Simultaneously a conventional partition based clustering, k-means, method is used to cluster patients based on the patient's attribute data. Since the criteria for finding the optimal number of clusters

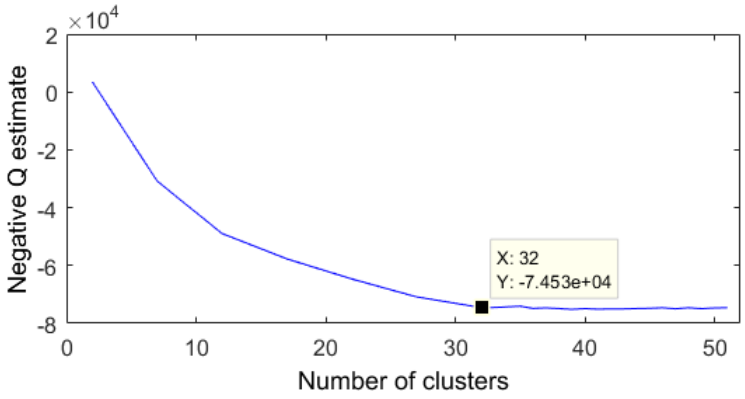
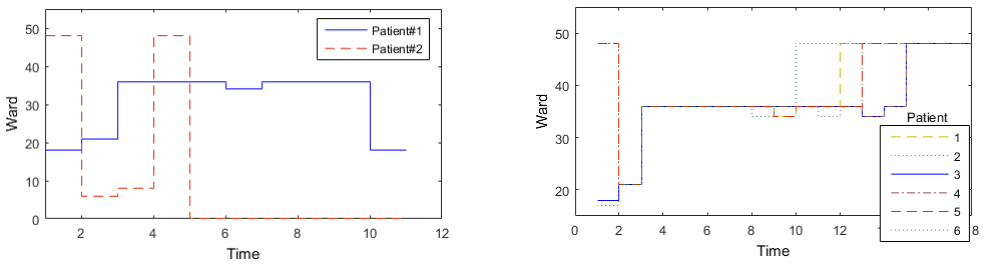


Figure 7: The estimated Q function against increasing number of clusters for the real data in Case Study. It is observed that the improvement in Q function is not significant after 32 clusters.

with k-means are rather subjective and in order to have a fair comparison, we use the same number of clusters as chosen by SMM (i.e., 32 clusters). This does not affect the optimization in RS even if we have a few redundant clusters, but prevents the risk of suboptimal results due to under-estimation of the number of clusters. Therefore, the benefits demonstrated by this case study represent a conservative estimate of the true potential benefits when compared to an application to a hospital in the real world. After performing k-means clustering, empirical trajectory distributions are estimated for each patient cluster.

To verify our claims that two patients with similar attributes may not follow the same trajectory, we observed two patients who were put into the same cluster using the k-means; they were both male, aged between 55-65 years and were diagnosed for heart disease. Their trajectories within hospital are shown in Fig. 8a. In this figure, patient#1 enters the cardiology ward, transitions to the angiography center then to the neurology ward and finally back to cardiology before leaving the hospital. Patient#2, on the other hand, begins their stay in the surgical ward, transitions to the heart clinic, then the ICU, then the operating theater, then to the ICU again and finally back to the surgical ward before getting discharged. Although the observed attributes for both patients show similar profiles and a heart disease diagnosis, the trajectories followed by these patients were very different. Observing their trajectories more closely, one can see that patient#2 might have had a severe heart condition, while patient#1 had a relatively milder heart condition only requiring angiography.

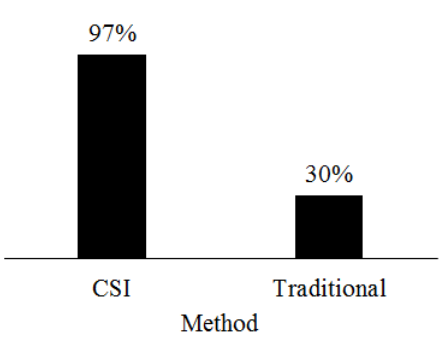


(a) Patient trajectories from a k-means cluster (b) Patient trajectories from a SMM based cluster

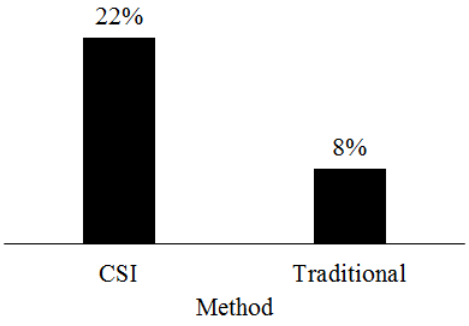
Figure 8: Trajectories of patients belonging to same cluster. It is observed that patients in SMM based clusters follow more similar trajectories than k-means.

When employing our SMM-clustering method, we do not see such dissimilarity in patient trajectories within one cluster. As an example, Fig. 8b shows trajectories of a few patients from one of the clusters identified by the SMM approach. Most of the patients in this cluster enter the hospital either in surgical or cardiology wards, then transition to the heart clinic, ICU, operating theater and finally cardiology before leaving the hospital. There is one case of patient#6 who entered the hospital in ortho and spine center, but then followed similar trajectory of going to heart clinic, ICU, operating theater and finally cardiology. This could be caused by a heart condition developing during an orthopedic admission, or possibly due to initial off-ward placement (because the cardiology ward was full). It is interesting to see that if we would have used the conventional attribute based clustering this patient would have been put into a orthopedic related cluster, while the SMM approach was able to identify the patient's "true" cluster.

To test the impact of our SMM approach for improved clustering and trajectory estimation on the RS optimization, we use the *maximum elective admission* formulation given in Sec. 2.3.2 and Online Appendix B. The goal is to increase the volume of patients served, thereby increasing revenues, while maintaining the same level of service and access. The results are shown in Fig. 9, with Fig. 9a and Fig. 9b showing the percentage increase in elective admissions and ward utilization, respectively, for the CSI and traditional methods relative to the baseline current elective admission schedule of the partner hospital. Our CSI method demonstrates a potential increase in elective admissions of 97%, while the traditional method can only achieve a 30% increase. As further validation that we are making



(a) Percentage increase in elective (EL) patients admission



(b) Percentage increase in ward utilization

Figure 9: Comparing the improvement in elective admissions and ward workload for proposed CSI and the traditional method with respect to the current service and workload levels using real hospital data.

the correct comparison, the 30% increase in elective admissions is comparable with increases reported in other elective admission optimization papers in the literature. Moreover, ward utilization improved by 22% for CSI, but only improved by 8% using the traditional method. This case study of an actual partner hospital demonstrates the importance of an accurate patient clustering and trajectory estimation method, as using our CSI not only provides a more accurate forecast of the hospital stochastic workload process, but also dramatically improves optimization solutions. Further, to the best of our knowledge, our CSI method is the only approach in the extant literature that has all the properties required for effective integration with admission scheduling optimization approaches: *scalable* to hospital of any size, considers *ward interactions*, and accounts for *patient heterogeneity*.

2.6 Appendices

2.6.1 Appendix A: Derivation of SMM-clustering update expressions for EM algorithm

In this appendix, we present the derivation of parameter update expressions for the EM algorithm in Sec. 2.3.1.2. As mentioned in the section, we have to obtain the posterior distributions of the parameters to find their optimal estimates that maximizes Eq. 8.

We use Dirichlet prior distributions, given in Eq. 10, for the parameters. The Dirichlet hyperparameters for parameters in $\mathbf{\Theta} = \{\pi^{(k)}, \boldsymbol{\rho}^{(k)}, \mathbf{P}^{(k)}, \mathbf{H}^{(k)}\}, k \in \mathcal{K}$ are denoted by $\{a_\pi^{(k)}, a_\rho^{(k)}, a_P^{(k)}, a_H^{(k)}\}, k \in \mathcal{K}$, respectively. For each model parameter, the hyperparameters

can be set to equal values, if there is no specific prior knowledge (non-informative prior). Besides, we assume the parameters are independent. Using it with the conditions on probability sums equal to 1 in Eq. 1 and parameter independence assumptions gives the following expressions for prior probabilities,

$$\begin{aligned} p(\pi) &\propto \prod_{k\in\mathcal{K}} \left(\pi^{(k)}\right)^{a_{\pi}^{(k)}-1} \\ p(\boldsymbol{\rho}) &\propto \prod_{k\in\mathcal{K}} \prod_{u\in\mathcal{U}} \left(\rho_u^{(k)}\right)^{a_{\rho,u}^{(k)}-1} \\ p(\mathbf{P}) &\propto \prod_{k\in\mathcal{K}} \prod_{u\in\mathcal{U}} \prod_{j\in\mathcal{U}} \left(P_{uj}^{(k)}\right)^{a_{P,uj}^{(k)}-1} \\ p(\mathbf{H}) &\propto \prod_{k\in\mathcal{K}} \prod_{u\in\mathcal{U}} \prod_{j\in\mathcal{U}} \left(P_{uj}^{(k)}\right)^{a_{P,uj}^{(k)}-1} \end{aligned} \tag{18}$$

Furthermore, using the parameter independence, the prior distribution for $\boldsymbol{\Theta}$ is,

$$p(\boldsymbol{\Theta}) = p(\pi)p(\boldsymbol{\rho})p(\mathbf{P})p(\mathbf{H}) \tag{19}$$

Plugging Eq. 19 and Eq. 2 into Eq. 8, and using the hyperparameters mentioned in Sec. 2.3.1.2, we get,

$$
\begin{aligned}
Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)}) &= \mathbb{E}_{\Theta^{(p)}}\left[\log(p(\mathbf{Y}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})\right] \\
&= \sum_{i=1}^{N}\sum_{k\in\mathcal{K}}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\log\left[\pi^{(k)}p_{\boldsymbol{\Theta}}(\mathbf{y}^{(i)}|Z^{(i)}=k)\right]+\log p(\boldsymbol{\Theta}) \\
&= \sum_{i=1}^{N}\sum_{k\in\mathcal{K}}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\log\left[\pi^{(k)}\rho^{(k)}_{u_1^{(i)}}\prod_{l=2}^{L}\left\{P^{(k)}_{u_{l-1}^{(i)},u_l^{(i)}}\cdot H^{(k)}_{u_{l-1}^{(i)},u_l^{(i)}}(\nu_l^{(i)})\right\}\right]+\log p(\pi)p(\boldsymbol{\rho})p(\mathbf{P})p(\mathbf{H}) \\
&= \sum_{i=1}^{N}\sum_{k\in\mathcal{K}}\log\left[\left(\pi^{(k)}\right)^{\Omega_{ik}(\boldsymbol{\Theta}^{(p)})}\left(\rho^{(k)}_{u_1^{(i)}}\right)^{\Omega_{ik}(\boldsymbol{\Theta}^{(p)})}\cdot\right. \\
&\quad \left.\prod_{l=2}^{L}\left\{\left(P^{(k)}_{u_{l-1}^{(i)},u_l^{(i)}}\right)^{\Omega_{ik}(\boldsymbol{\Theta}^{(p)})}\cdot\left(H^{(k)}_{u_{l-1}^{(i)},u_l^{(i)}}(\nu_l^{(i)})\right)^{\Omega_{ik}(\boldsymbol{\Theta}^{(p)})}\right\}\right]+\log p(\pi)p(\boldsymbol{\rho})p(\mathbf{P})p(\mathbf{H}) \\
&\propto \log\left[\prod_{k\in\mathcal{K}}\left(\pi^{(k)}\right)^{\left(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})+a_{\pi}^{(k)}-1\right)}\right]+ \\
&\quad \sum_{k\in\mathcal{K}}\log\left[\prod_{u\in\mathcal{U}}\left(\rho_u^{(k)}\right)^{\left(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\kappa(u_1^{(i)},u)+a_{\rho,u}^{(k)}-1\right)}\right]+ \\
&\quad \sum_{k\in\mathcal{K}}\sum_{u\in\mathcal{U}}\log\left[\prod_{j\in\mathcal{U}}\left(P_{uj}^{(k)}\right)^{\left(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\bar{\kappa}_{uj}(\mathbf{y}^{(i)})+a_{P,uj}^{(k)}-1\right)}\right]+ \\
&\quad \sum_{k\in\mathcal{K}}\sum_{u\in\mathcal{U}}\sum_{j\in\mathcal{U}}\log\left[\prod_{\nu\in\mathbb{N}}\left(H_{uj}^{(k)}(\nu)\right)^{\left(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\tilde{\kappa}_{uj,\nu}(\mathbf{y}^{(i)})+a_{H,uj}^{(k)}(\nu)-1\right)}\right] \\
&\propto \log\left[\pi^{(k)}\sim Dirichlet(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})+a_{\pi}^{(k)})\right]+ \\
&\quad \log\left[\rho_u^{(k)}\sim Dirichlet(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\kappa(u_1^{(i)},u)+a_{\rho,u}^{(k)})\right]+ \\
&\quad \log\left[P_{uj}^{(k)}\sim Dirichlet(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\bar{\kappa}_{uj}(\mathbf{y}^{(i)})+a_{P,uj}^{(k)})\right]+ \\
&\quad \log\left[H_{uj}^{(k)}(\nu)\sim Dirichlet(\sum_{i=1}^{N}\Omega_{ik}(\boldsymbol{\Theta}^{(p)})\tilde{\kappa}_{uj,\nu}(\mathbf{y}^{(i)})+a_{H,uj}^{(k)}(\nu))\right] \qquad (20)
\end{aligned}
$$

where, $\kappa(x,y)$ is an indicator function equal to 1 if $x=y$, $\bar{\kappa}_{uj}(\mathbf{y}^{(i)})$ is the count function equal to the number of times transition was made from state u to j in trajectory $\mathbf{y}^{(i)}$, and $\tilde{\kappa}_{uj,\nu}(\mathbf{y}^{(i)})$ is the count function equal to the number of times transition was made from state u to j, in trajectory $\mathbf{y}^{(i)}$, when length of stay at state u was ν time units.

As shown in Eq. 20, the posteriors of the model parameters are Dirichlet distributions with updated hyperparameters. The posterior of any Dirichlet variable, $x_1,\ldots,x_m\sim$

$Dirichlet(a_1, \ldots, a_m)$ is maximized at $E[x_i] = \frac{a_i}{\sum_{i'=1}^{m} a_{i'}}, \forall i$. Thus, the parameter estimates to maximize Eq. 8 are,

$$
\begin{aligned}
\pi^{(k)(p+1)} &= \frac{\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) + a_{\pi}^{(k)}}{\sum_{k' \in \mathcal{K}} \left[\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) + a_{\pi}^{(k')}\right]}, \forall k \in \mathcal{K}. \\
\rho_u^{(k)(p+1)} &= \frac{\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) \kappa(u_1^{(i)}, u) + a_{\rho,u}^{(k)}}{\sum_{u' \in \mathcal{U}} \left[\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) \kappa(u_1^{(i)}, u') + a_{\rho,u'}^{(k)}\right]}, \forall k \in \mathcal{K}, u \in \mathcal{U} \\
P_{uj}^{(k)(p+1)} &= \frac{\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) \bar{\kappa}_{uj}(\mathbf{y}^{(i)}) + a_{P,uj}^{(k)}}{\sum_{j' \in \mathcal{U}} \left[\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) \bar{\kappa}_{uj'}(\mathbf{y}^{(i)}) + a_{P,uj'}^{(k)}\right]}, \forall k \in \mathcal{K}, (u, j) \in \mathcal{U} \\
H_{uj}^{(k)}(\nu)^{(p+1)} &= \frac{\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) \tilde{\kappa}_{uj,\nu}(\mathbf{y}^{(i)}) + a_{H,uj}^{(k)}(\nu)}{\sum_{\nu' \in \mathbb{N}} \left[\sum_{i=1}^{N} \Omega_{ik}(\boldsymbol{\Theta}^{(p)}) \tilde{\kappa}_{uj,\nu'}(\mathbf{y}^{(i)}) + a_{H,uj}^{(k)}(\nu')\right]}, \forall k \in \mathcal{K}, (u, j) \in \mathcal{U}, \nu \in \mathcal{T}
\end{aligned}
$$

2.6.2 Appendix B: Elective Scheduling Optimization MIP Formulation

In this appendix we present, an optimization model from the literature (Helm and Van Oyen (2015)). that is used to demonstrate the importance of a rigorous patient trajectory estimation procedure. We designed our estimation approach to integrate with optimization and what-if scenarios, with this particular optimization being used as a proof of concept that (1) our method integrates well with current optimization approaches, and (2) our method significantly improves the outcome of the optimization when compared with traditional approaches proposed for use with these types of models. We begin by describing the model parameters and then present the optimization model with brief description of the objective and constraints. For a more detailed description of the optimization approach we refer the readers to Helm and Van Oyen (2015).

Sets

$\mathcal{K}$	set of all patient types
$\mathcal{U}$	set of hospital wards

Hospital parameters

ζ	vector of ward capacities
η	vector of total cancellations attributed for each ward
b	limit on the average number of blockages per week
$\mathbf{o}$	vector of limit on the average number of off-unit patients allowed for each ward
$\mu_d^{(k)}$	current elective admission volume of type k patients on day d
$\bar{\mu}_d^{(k)}$	maximum number of elective admissions of type k allowed on day d
$\mathbf{R}$	reward vector where R_k is the reward for admitting patient of type k

Patient trajectory and census distributions

$\gamma_u^{(k)}(d_1)$	probability that an elective patient of type k requires a bed in ward u, d_1 days after admission (trajectory distribution)
$p_{u,d}(n)$	probability that there are n emergency patients demanding a bed in ward u on day d
$\bar{p}_d(n)$	probability that there are n emergency patients demanding a bed in the hospital on day d

Decision Variables

$\Psi_d^{(k)}$	number of type $k \in \mathcal{K}$ patients scheduled on day d
$\delta_{d,n}$	number of blockages if there are n emergency patients in the hospital on day d
$\acute{o}_{d,n}^{u}$	number of ward u off-unit patients on day d if there are n emergency patients in ward u

The patient trajectory and census distribution parameters are computed offline as explained earlier in this section. Since the PALM model for emergency patient bed demand is exogenous to the decision variable, this too is calculated off-line, with the results captured as $p_{u,d}(n)$ and $\bar{p}_d(n)$. We consider a weekly planning horizon that repeats itself every week,

generating a cyclo-stationary system that varies by day of week. The objective is to maximize the throughput of the sum of elective patient admissions (over the planning horizon) of each type weighted by a "reward" vector $\mathbf{R}$ ($\mathbf{1}$ denotes a column vector of all ones). The reward vector gives flexibility to allow the model to treat one patient type differently from another, for example, the model can prioritize one patient type over another with respect to patient criticality, projected revenue generated by the admission, or other strategic priority. The formulation is as follows:

$$\max_{\Theta,\delta,\hat{\delta}} \mathbf{R} \cdot \Psi \cdot \mathbf{1} \tag{21}$$

s.t.

$$\delta_{d_1,n} \geq n - \sum_{u\in\mathcal{U}}(\zeta_u - \sum_{d_2=1}^{7}\sum_{k\in\mathcal{K}} \Psi_{d_2}^{(k)} \cdot \sum_{n'=0}^{\infty} \gamma_u^{(k)}(7n' + d_1 - d_2)), \tag{22}$$

$$d_1 = 1,\ldots,7;\ n = 1,2,\ldots$$

$$\sum_{d=1}^{7}\sum_{n=0}^{\infty} \bar{p}_d(n)\delta_{d,n} \leq b \tag{23}$$

$$\delta_{d,n+1} \geq \delta_{d,n} \qquad d = 1,\ldots,7;\ n = 1,2,\ldots \tag{24}$$

$$\acute{o}_{d_1,n}^{u} \geq n + \sum_{d_2=1}^{7}\sum_{k\in\mathcal{K}} \Psi_{d_2}^{(k)} \cdot \sum_{n'=0}^{\infty} \gamma_u^{(k)}(7n' + d_1 - d_2) - \zeta_u - \eta_u \sum_{d=0}^{7}\sum_{n'=0}^{\infty} \delta_{d,n'} \cdot \bar{p}_d(n') \tag{25}$$

$$\forall u \in \mathcal{U};\ d_1 = 1,\ldots,7;\ n = 1,2,\ldots$$

$$\sum_{n=0}^{\infty} p_{u,d}(n)\acute{o}_{d,n}^{u} \leq \mathbf{o}_u \qquad \forall u \in \mathcal{U};\ d = 1,\ldots,7 \tag{26}$$

$$\acute{o}_{d,n+1}^{u} \geq \acute{o}_{d,n}^{u} \qquad d = 1,\ldots,7;\ n = 1,2,\ldots \tag{27}$$

$$\sum_{d=1}^{7} \Psi_d^{(k)} \geq \sum_{d=1}^{7} \mu_d^{(k)} \qquad \forall k \in \mathcal{K} \tag{28}$$

$$\Psi_d^{(k)} \leq \bar{\mu}_d^{(k)} \qquad \forall k \in \mathcal{K};\ d = 1,\ldots,7 \tag{29}$$

$$\Psi_d^{(k)}, \delta_{d,n}, \acute{o}_{d,n}^{u} \in \mathbb{Z}^+$$

The constraints of this model are primarily for constraining the blockages faced by the patients, limiting off-ward placement, and respecting the hospital resource limits. Since the purpose of this work is to demonstrate how CM can be improved by developing methods that integrate with optimization, and not to provide new optimization methods, we briefly describe the optimization presented here. Greater detail regarding this approach can be found in Helm and Van Oyen (2015). Constraints 22 calculate the number of blocked patients at the hospital level if n emergency patients are in the hospital on day d_1. This sets the helper variable, $\delta_{d,n}$ which is subsequently used to calculate expected blockages according to the distribution on the emergency patient bed demand stochastic process in the left hand side (LHS) of Constraints 23 by multiplying the indicator of whether the n^{th} patient would be blocked by the probability of seeing n emergency patients in the hospital. The right hand side constrains the expected blocked patients to be less than some target level, b, which can be chosen by management. Constraint 24 is a cut that is added to the formulation that significantly improves model solution speed.

Similar to the constraints (Eq. 22-24) for blockages, we have constraints in Eq. 25-27 for approximating and limiting expected off-unit census. An additional term in Eq. 25, $\eta_u \sum_{d=0}^{7} \sum_{n'=0}^{\infty} \delta_{d,n'} \cdot \bar{p}_d(n')$, subtracted from the otherwise expected number of off-unit census gives patients who were blocked and not able to be admitted to the hospital in the first place.

Constraints in Eq. 28 ensures that the proper mix of patients is respected. Specifically, it ensures that each patient type has at least as many admissions each week as they did prior to optimization. Constraints 29 ensure that the model respects the hospital resource capacity for a day. For example, hospitals frequently avoid admitting elective patients on Sundays, which could be achieved by setting $\bar{\mu}^{(k)}_{Sunday} = 0$.

2.6.3 Appendix C: Distributions for assigning attributes to patients in simulation study

In Sec. 2.4.2.1, we assign physical attributes to patient for our simulation study. We perform a conservative assignment, in favor of traditional patient clustering method, by giving higher chance of patients within a *true* cluster having similar attributes. Table 3 below shows the

Cluster	Age
1	$N(20, 3)$
2	$N(30, 3)$
3	$N(40, 3)$
4	$N(50, 3)$

(a) Normal Distributions for patient age

	Sex	
Cluster	**M**	**F**
1	80%	20%
2	20%	80%
3	70%	30%
4	30%	70%

(b) Uniform distribution for patient sex within clusters

	Diagnosis		
Cluster	**D1**	**D2**	**D3**
1	70%	20%	10%
2	20%	70%	10%
3	10%	20%	70%
4	80%	10%	10%

(c) Uniform distribution for patient diagnosis

Table 3: Generating distributions for patient attributes within *true* clusters

generating distributions for the patient attributes within each cluster. As shown in the table, age is taken from a normal distribution with different means (Table 3a), sex and diagnosis (Table 3b-3c) are taken according to a Bernoulli random variable with different success probabilities. The distribution parameters are chosen such that there is high attribute similarity (dissimilarity) between patients within (between) clusters.

2.7 References

Abraham, G., Byrnes, G. B., & Bain, C. A. (2009). Short-term forecasting of emergency inpatient flow. IEEE Transactions on Information Technology in Biomedicine, 13(3), 380-388.

Adan, I., Bekkers, J., Dellaert, N., Vissers, J., & Yu, X. (2009). Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning. Health care management science, 12(2), 129-141.

Aiken, L. H., Clarke, S. P., Sloane, D. M., Sochalski, J., & Silber, J. H. (2002). Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. JAMA: the journal of the American Medical Association, 288(16), 1987-1993.

Armony, M., Israelit, S., Mandelbaum, A., Marmor, Y. N., Tseytlin, Y., & Yom-Tov, G. B. (2011). Patient flow in hospitals: A data-based queueing-science perspective. Stochastic Systems, 20.

Bekker, R., & Koeleman, P. M. (2011). Scheduling admissions and reducing variability in bed demand. Health care management science, 14(3), 237-249.

Billingsley, P. (1961a). Statistical methods in Markov chains. The Annals of Mathematical Statistics, 12-40.

Billingsley, P. (1961b). Statistical inference for Markov processes (Vol. 2). Chicago: University of Chicago Press.

Channouf, N., L'Ecuyer, P., Ingolfsson, A., & Avramidis, A. N. (2007). The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. Health Care Management Science, 10(1), 25-45.

Earnest, A., Chen, M. I., Ng, D., & Sin, L. Y. (2005). Using autoregressive integrated moving average (ARIMA) models to predict and monitor the number of beds occupied during a SARS outbreak in a tertiary hospital in Singapore. BMC Health Services Research, 5(1), 36.

Faddy, M. J., & McClean, S. I. (2000). Analysing data on lengths of stay of hospital patients using phase-type distributions. Applied Stochastic Models in Business and Industry, 15(4), 311-317.

Fetter, R. B., Shin, Y., Freeman, J. L., Averill, R. F., & Thompson, J. D. (1980). Case mix definition by diagnosis-related groups. Medical care, 18(2), i-53.

Green, L. (2006). Queueing analysis in healthcare. In Patient flow: reducing delay in healthcare delivery (pp. 281-307). Springer US.

Griffin, J., Xia, S., Peng, S., & Keskinocak, P. (2012). Improving patient flow in an obstetric unit. Health care management science, 15(1), 1-14.

Griffith, J. R., Hancock, W. M., & Munson, F. C. (Eds.). (1976). Cost control in hospitals. Health Administration Press.

Hall, R., Belson, D., Murali, P., & Dessouky, M. (2006). Modeling patient flows through the healthcare system. In Patient flow: Reducing delay in healthcare delivery (pp. 1-44). Springer US.

Hancock, W. M., & Walter, P. F. (1979). The use of computer simulation to develop hospital systems. ACM SIGSIM Simulation Digest, 10(4), 28-32.

Hancock, W. M., & Walter, P. F. (1983). The "ASCS": Inpatient Admission Scheduling and Control System. Health Administration Press.

Harper, P. R. (2005). A review and comparison of classification algorithms for medical decision making. Health Policy, 71(3), 315-331.

Harper, P. R., & Shahani, A. K. (2002). Modelling for the planning and management of bed capacities in hospitals. Journal of the Operational Research Society, 53(1): 11-18.

Helm, J. E., & Van Oyen, M. P. (2015). Design and optimization methods for elective hospital admissions. Operations Research, 62(6), 1265-1282.

Irvine, V., McClean, S., & Millard, P. (1994). Stochastic models for geriatric in-patient behaviour. Mathematical Medicine and Biology, 11(3), 207-216.

Jacobson, S. H., Hall, S. N., & Swisher, J. R. (2006). Discrete-event simulation of health care systems. In Patient flow: Reducing delay in healthcare delivery (pp. 211-252). Springer US.

Jones, S. A., Joy, M. P., & Pearson, J. (2002). Forecasting demand of emergency care. Health care management science, 5(4), 297-305.

Kao, E. P. (1972). A semi-Markov model to predict recovery progress of coronary patients. Health Services Research, 7(3), 191.

Kao, E. P. (1974). Modeling the movement of coronary patients within a hospital by semi-Markov processes. Operations Research, 22(4), 683-699.

Keehan, S., Sisko, A., & Truffer, C. (2007). Expenses for hospital inpatient stays: 2004. AHRQ, Statistical Brief, 164.

Konrad, R., DeSotto, K., Grocela, A., McAuley, and others (2013). Modeling the impact of changing patient flow processes in an emergency department: Insights from a computer simulation study. Operations Research for Health Care, 2(4), 66-74.

Littig, S. J., & Isken, M. W. (2007). Short term hospital occupancy prediction. Health care management science, 10(1), 47-66.

Marshall, A. H., & McClean, S. I. (2003). Conditional phase-type distributions for modelling patient length of stay in hospital. International Transactions in Operational Research, 10(6), 565-576.

Massey, W. A., & Whitt, W. (1993). Networks of infinite-server queues with nonstationary Poisson input. Queueing Systems, 13(1-3), 183-250.

McLachlan, G. J., & Krishnan, T. (2007). The EM algorithm and extensions (Vol. 382). Wiley-Interscience.

Richardson, D. B. (2006). Increase in patient mortality at 10 days associated with emergency department overcrowding. Medical Journal of Australia, 184(5), 213.

Ridley, S., Jones, S., Shahani, A., Brampton, W., Nielsen, M., & Rowan, K. (1998). Classification trees: A possible method for iso-resource grouping in intensive care. Anaesthesia, 53(9), 833-840.

Smallwood, R. D., Murray, G. R., Silva, D. D., Sondik, E. J., & Klainer, L. M. (1969). A medical service requirements model for health system design. Proceedings of the IEEE, 57(11), 1880-1887.

Taylor, G. J., McClean, S. I., & Millard, P. H. (2000). Stochastic models of geriatric patient bed occupancy behaviour. Journal of the Royal Statistical Society: Series A (Statistics in Society), 163(1), 39-48.

Thomas, W. H. (1968). A model for predicting recovery progress of coronary patients. Health services research, 3(3), 185.

Weiss, E. N., Cohen, M. A., & Hershey, J. C. (1982). An iterative estimation and validation procedure for specification of semi-Markov models with application to hospital patient flow. Operations Research, 30(6), 1082-1104.

Wu, C. J. (1983). On the convergence properties of the EM algorithm. The Annals of statistics, 95-103.

Zeltyn, S., et al. (2011). Simulation-based models of emergency departments: Operational, tactical, and strategic staffing. ACM Transactions on Modeling and Computer Simulation (TOMACS), 21(4), 24.

Zhang, B., Murali, P., Dessouky, M. M., & Belson, D. (2009). A mixed integer programming approach for allocating operating room capacity. Journal of the Operational Research Society, 60(5), 663-673.

CHAPTER III

LONGITUDINAL MRI DATA ANALYSIS IN PRESENCE OF MEASUREMENT ERROR AND ABSENCE OF REPLICATES

Longitudinal data, commonly found in healthcare and medical applications, contains a series of measurements/data collected for a subject at different points in time. Several important drug efficacy analysis such as assessment of effectiveness of a drug delivery system to cancer cells and progress of patients' condition on administering a drug, is done with the help of such longitudinal data. Another application of longitudinal data is found in the area of patient monitoring and diagnosis using magnetic resonance imaging (MRI). It is a medical imaging technique that captures the anatomy and the physiological processes of a subject.

MRI scans are generally repeated over a period of time to yield a longitudinal data. Such longitudinal data is analyzed to assess improvement or deterioration in patient's condition over time. MRI is commonly used in diagnosis of neurological cancers, dementia, cerebrovascular disease, heart diseases, oncology, etc. The analysis of MRI data and their extracted features have been widely studied in several disciplines. However, these analyses often rely on a complete sanctity of the data, meaning the collected data has no inherent error or corruptness. Contrary to this belief, it is very common that a measurement system brings inherent errors in the recorded data. This is more common when the measurement system has a multi-stage procedure. For example, MRI scanning entails a visual scan by an MRI machine followed by steps for post-processing the image to obtain levels of various biological markers. In each stage, the measurements/data can be contaminated by, for example, patient movement while scanning, incorrect image processing, etc.

The presence of measurement errors in longitudinal data hampers an adequate assessment of patients. As also stated by Carroll et al. (2006), it results in inflated variance of signal estimates — which in patient diagnosis is translated to treatment effect on progress of patient condition. This may sometimes mask the true effect or indicate a false effect.

Despite its importance, the existence of measurement errors in longitudinal data analysis is often ignored, leading to misguided conclusions. The impact of such inadequate assessments can be extremely severe.

For example, consider a longitudinal MRI data of older patients to detect symptoms of any cognitive decline (which leads to Alzheimer's disease). Presence of measurement errors in the collected data will inflate its variance. This can mask a true declining trend in a patient's condition, leading to delayed treatment, and possibly to Alzheimer disease, which is the sixth leading cause of death among senior citizens in the U.S. Besides, in the other scenarios, it may cause false signal detection on patients' condition resulting in unnecessary medications and cost burden.

To address this issue, it is imperative to estimate and isolate measurement errors for an accurate data analysis. In the presence of replicates, this is rather straightforward — the within-replicates variation can be easily estimated and considered as the result of measurement errors. However, it becomes particularly challenging when there are no replicates.

The scenario of unreplicated data is quite common in longitudinal processes found in medical applications. This can be attributed to high cost or potential health hazards (to patients or operators) in acquisition of data, or to the limited availability of patients. For the same reasons, the time intervals between two measurements are long. For example, the time intervals between MRI scans of potential Alzheimer patients (used in this chapter as a case study) is about 6 months or more depending on the patient's availability. In such situations, any observed variations can be due to either an actual change in the condition of the subject of interest, or inherent measurement errors in the system. In addition, another challenge in medical data analysis is missing values due to patients unavailability or dropouts. Although several existing methods have tackled this issue, they all assume either measurement errors do not exist or they are so small that can be neglected.

In this chapter, we propose a new methodology to accurately model a longitudinal process in the presence of measurement errors in the data and the absence of replicates. The outcomes and salient features of the proposed methodology are (1) the estimation of the

measurement system variation and its separation from the process variation, (2) being robust to missing values, (3) the accurate pattern analysis of longitudinal data by removing measurement variations, and (4) providing precise confidence interval for model parameters leading to more powerful statistical testing. To achieve this, we utilize the fact that longitudinal data are autocorrelated to decompose the overall error variance into measurement errors, autocorrelation and random noises. This is done developing a new estimation method that integrates a variogram estimation with an EM framework for mixed-effect regression.

The remainder of this chapter is organized as follows: In Section 3.1, we review the relevant methods in the literature and discuss their drawbacks. Section 3.2 begins with formulating the problem using mixed-effect regression. Then, we present the proposed EM-Variogram technique for model parameter estimation under the absence of replication. Further, we validate the methodology using extensive experimental simulations in Section 3.3. As most longitudinal studies have missing values due to subject drop-outs, mistimed visits, premature study termination, etc., we will also preform a sensitivity analysis on missing data in this section. In Section 3.4, a case study of analyzing progression in Alzheimer's disease using longitudinal MRI is presented to illustrate the efficacy of the proposed methodology in real applications.

3.1 Related Work

Longitudinal data analysis is a classical problem on which extensive research has been done over past decades. Among this research, the commonly used methods can be listed as, mixed-effect models, generalized estimating equations (GEE), and transitional models. Mixed-effect models incorporate between-individual variations and within-individual correlations in longitudinal data. GEE models the mean structure and correlation structure separately without distributional assumptions for the data, while transitional models considers Markov structures for within-individual correlations.

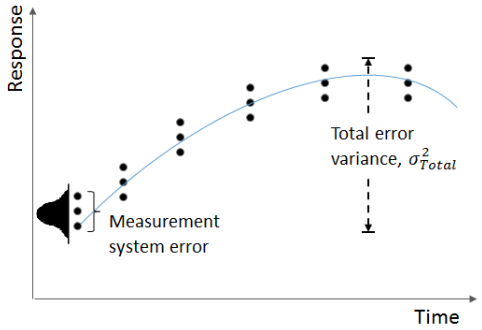
As Chaganty and Joe (2004) indicated the choices of valid working correlation matrices can be very limited in GEE approach, restricting its applicability for modeling measurement errors. Besides, the parameters in GEE have population average interpretations, and

thus, does not capture the influence of explanatory variables on responses for heterogeneous units. Moreover, transitional models can be mathematically and computationally challenging to incorporate measurement errors (Carroll et al., 2006). Mixed-effect models, however, have been widely used for modeling longitudinal data with measurement errors. Carroll et al. (2006) provided an overview of these models. They also pointed out the possibility of unreplicated data that makes a direct estimation of the measurement error variance challenging. For illustration, a simple example of a data series with replicates is shown in Fig. 10a. The total variation of the data, σ^2_{Total}, can be decomposed into the variation caused by the functional mean and the gauge variation. In this case, one can easily use a traditional mixed-effect regression model (Laird and Ware, 1982) to estimate all variation components including the gauge variation (caused by measurement system errors) as well as the functional mean. However, as we can see in Fig. 10b, when replicated measurements are not available, these variations are confounded and cannot be separated. To address this issue, Carroll et al. (2006) proposed to use an instrumental variable (IV). An IV can be seen as an additional covariate with some properties such as having no measurement errors or being uncorrelated with other variables. However, in practice, such IVs may not be present and a falsely assumed IV can lead to erroneous inferences.

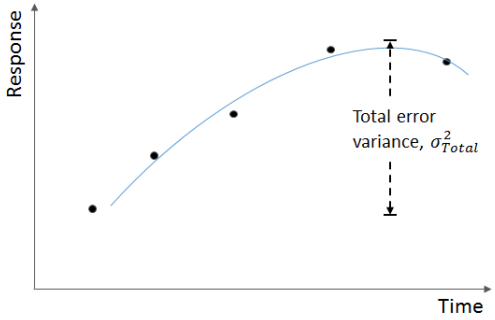
Another significant body of literature focuses on measurement errors in the regressors (covariates) in a longitudinal study. For example, Wang, et al. (1998) developed a generalized linear mixed measurement error model that assumes a classical additive error on the regressors. Higgins et al. (1997) and Wu (2002) describe a mixed-effects model where the random effects depend on covariates measured with error. Contrary to this, in our problem the variable of interest, i.e. the response variable, potentially has measurement errors. Although Abrevaya and Hausman (2004) stated that measurement errors in the response variable can be ignored because it gets absorbed into the model residuals, it adversely affects the signal detection power of the model. Carroll et al., 2006, studied the effect of these errors on response variables and concluded that it increases the variability of the estimates, and thus decreases the detection power. Buonaccorsi (1991, 1996) and Buonaccorsi and Tosteson (1993) discussed some methods to address this problem. These methods relied on a

error-free validation data and/or at least two independent unbiased replicate measurements of response, which may not be available in practice.

Carroll et al. (2006), also mentions the use of repeated and reproduced measurements for gauging the measurement error in the data modeling. This approach is part of a broader gauge repeatability and reproducibility (GRR) analysis methods in the area of Measurement System Analysis (MSA). Repeatability is referred to as the variation in measurements when a characteristic is measured multiple times by the same measurement instrument (gauge), appraiser and procedure, while reproducibility is defined as the variation of measurements when a characteristic is measured by multiple measurement instrument (gauges), appraisers, and/or procedures. The measurement system variability can be estimated using either or both repeatability and reproducibility variations. There exists extensive research on GRR analysis and several methods have been developed for estimating the measurement system variability. Statistical control charts, e.g., $\bar{X}$ and R charts, are commonly used for estimating repeatability and reproducibility, in which $\bar{X}$ chart provides a means for estimating the reproducibility and R chart measures the consistency of each appraiser that pertains to repeatability (MSA Manual, 2010). Analysis of variance (ANOVA) with random effects is another GRR method widely used to estimate the components of measurement systems variability (see Montgomery, 2001, Montgomery and Runger, 1993a, b, Borror et al., 1997, Burdick and Larson, 1997, and Burdick et al., 2007 for more detail). Other GRR methods include correlation coefficients analysis (Halligan, 2002), intra-class correlation (Bland and Altman, 1986), measure of agreements (Barnhart et al., 2007) and repeatability coefficient (Lexell and Downham, 2005, and Bland and Altman, 2003). GRR requires replicates of data under stationary conditions, that is, when the distribution of the measured characteristic is constant during the measurements. However, in longitudinal data, where the measurements are taken over time, often the mean and variance of the measured characteristic are not necessarily constant. Besides, as mentioned before, measurements may not be repeated or reproduced at the same observation time (or in a very short interval) due to cost and/or risk considerations. Thus, the existing GRR methods cannot be applied on this problem.



(a) Replicated measurements at stationary intervals. Measurement system error can be computed from replicated measurements and decoupled from total error variance.



(b) Unreplicated measurements at stationary intervals. Computation of measurement system error is challenging in such cases.

Figure 10: Longitudinal data with replicated and unreplicated measurements, respectively, at stationary intervals.

3.2 EM-Variogram Variance Decomposition for Measurement Error Analysis in Longitudinal Data

As discussed earlier, this chapter focuses on measurement error analysis for longitudinal data. In longitudinal studies, the variable of interest is measured over time on a randomly selected sample of subjects. We formulate the longitudinal data analysis problem by using linear mixed-effect models in which the total variation of a longitudinal dataset is comprised of the functional mean (fixed effects), subject-specific (random) effects, measurement system errors and autocorrelated noises. Although (restricted) maximum likelihood methods can be used to fit a linear mixed-effect model and estimate fixed and random effects, these methods only provide a confounded variance estimate and cannot decouple the measurement system variance from the noise variance in case of unreplicated measurements (to be discussed in the following subsection). To cope with this problem, we develop a new estimation approach by integrating EM algorithm and Variogram. Our main assumptions in the proposed approach are, 1) the modeling error caused by model mis-specification is negligible, and 2) noises are autocorrelated, which is a valid assumption for most longitudinal data.

3.2.1 Problem formulation

Suppose we have a sample of longitudinal data measured over a random sample of units of size N. The measurements are made at discrete observation times. The length of the

longitudinal data and observation times can vary across units. This is because of missing values resulting from unavailability or different availability times of a unit. We denote the set of all discrete measurement times for all units in the sample as $\mathcal{T}$, and $T_i \subset \mathcal{T}$, $i = 1, \ldots, N$ will denote the measurement times of unit i. We use a linear mixed effect (LME) model to model the sample of longitudinal data as follows:

$$\mathbf{y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\epsilon}_i, \tag{30}$$

where $\mathbf{y}_i$ is the response vector for subject i consisting of $|T_i|$ observations; β is a $p \times 1$ vector of unknown fixed-effect parameters; and $\mathbf{X}_i$ is a $|T_i| \times p$ known design matrix for subject i. The random effect parameters are unit specific and denoted by $\mathbf{b}_i$ for subject i where $\mathbf{b}_i$ is a $q \times 1$ vector of unknown random individual effects, assumed to be normally distributed with zero mean and covariance Q, i.e., $\mathbf{b}_i \sim N(\mathbf{0}, Q)$. Corresponding to $\mathbf{b}_i$, $\mathbf{Z}_i \subseteq \mathbf{X}_i$ is an $|T_i| \times q$ known design matrix. $\boldsymbol{\epsilon}_i$ is an $|T_i| \times 1$ vector of error terms, assumed to be normally distributed with $N(\mathbf{0}, \boldsymbol{\Omega}_i)$ where Ω_i is a $|T_i| \times |T_i|$ positive-definite covariance matrix. Therefore, the conditional distribution of $\mathbf{y}_i$ can be written as

$$\mathbf{y}_i|\beta, \mathbf{b}_i \sim N(X_i\beta + Z_i\mathbf{b}_i, \Omega_i) \tag{31}$$

In the LME model in Eq. 30, the first term describes the overall trend over time, the second term represents the deviations from the mean due to unit-to-unit differences, and the error term, $\boldsymbol{\epsilon}_i$, includes both measurement system error and autocorrelated noises. Although in theory, it is possible to assume a general (unconstrained) structure for the covariance matrix of $\boldsymbol{\epsilon}_i$, in practice, the number of parameters are too large to estimate, and hence one should assume a constrained structure such as autoregressive, exponential, gaussian, etc. The elements of Ω_i lump together the variance caused by measurement system error and the variance of autocorrelated noises. We, therefore, use the following decomposed covariance structure to decouple them (Diggle et al., 2002).

$$\Omega_i = \tau^2\mathbf{I}_i + \sigma^2\Phi_\phi(\mathbf{U}_i), \tag{32}$$

where τ^2 is the variance of the measurement system error; σ^2 is the variance of random noises; and $\Phi_\phi(\mathbf{U}_i)$ is a $|T_i| \times |T_i|$ autocorrelation matrix for the lag matrix $\mathbf{U}$. Typically an exponential or gaussian autocorrelation function, represented by $\Phi_\phi(u) = e^{-\phi u}$ and $\Phi_\phi(u) = e^{-\phi u^2}$ respectively, are used (see Diggle et al. (2002) for further detail). This autocorrelation function, along with the factor of σ^2, and the measurement system error gives the covariance of error for subject i.

To simplify the model representation, we denote the vector of variance parameters by $\theta = (\boldsymbol{\alpha}, Q)$, where, $\boldsymbol{\alpha} = \{\tau^2, \sigma^2, \phi\}$, and the vector of fixed and random effects by $B = (\beta, \mathbf{b})$. A restricted maximum likelihood (REML) method is traditionally used for estimating the parameters of an LME (Laird and Ware, 1982). However, due to the decomposed structure of the error covariance matrix in Eq. 30, REML cannot be used directly because, a) it is easy to show that there is no analytical (closed-form) update expressions for the variance parameters, $\boldsymbol{\alpha}$, for the REML's EM estimation approach, b) numerically solving the REML likelihood function with the decomposed variance structure can give inconsistent results due to a complex objective function, and c) computing an unconstrained variance and then fitting Eq. 32 to estimate $\boldsymbol{\alpha}$ is not possible, as it will require estimating an Ω_i for each unit $i = 1, \ldots, N$, which is very difficult with typically few data for each unit. Other estimation methods, that solves for a linear mixed-effect model using either maximum likelihood or REML objective function, for example, parameter expanded EM, will also suffer from the same problems.

Another class of methods uses a *full* Bayesian approach with Gibbs sampling for estimation, see for example, Zeger and Karim (1991) and Gilks et al.(1993). But a full Bayesian formulation would require computation of posteriors or conditional posterior functions. This is difficult if the underlying parametric model does not have a linear or log-linear structure, as in Eq. 32. The conditional dependency of $\boldsymbol{\epsilon}_i$ and its non-linear covariance structure, makes development of a full Bayesian formulation unclear.

Therefore, in the following section (Â§3.2.2), we propose a new estimation technique, called EM-Variogram, that can model the decomposed covariance structure and estimate both the mean and variance parameters.

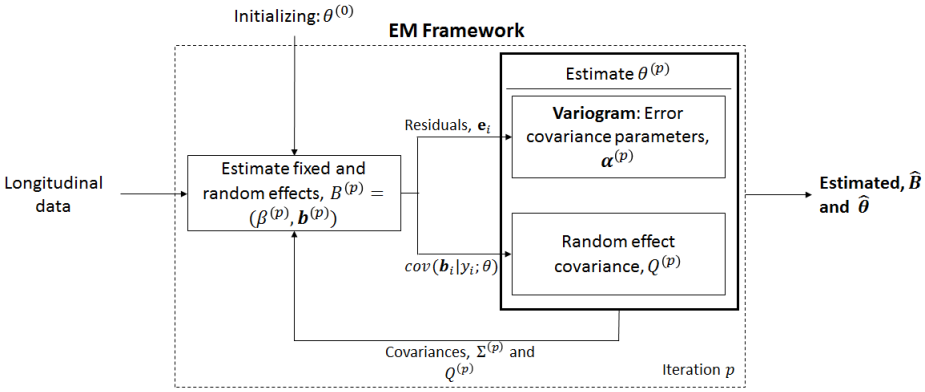


Figure 11: The overview of estimation procedure

3.2.2 EM-Variogram for Estimating LME Model Parameters

A high level overview of the procedure is given in Figure 11. As shown in the figure, the ultimate goal is to estimate $\theta = (\boldsymbol{\alpha}, Q)$ and $B = \{\beta, \mathbf{b}\}$. We develop an EM framework, that combines EM algorithm with Variogram estimation technique to solve for Model 30, yielding $\hat{\theta}$ and $\hat{B}$. The developed method is an iterative approach that takes in a longitudinal data, and an initial value of variance parameters in θ as inputs. Within each iteration, update rules derived from REML and Variogram estimation are used for updating θ and B, until convergence. In the following, we will explain each part of the procedure in detail.

Unlike MLE, the restricted maximum likelihood (REML) approach yields smaller bias, and hence, typically, used for LME parameter estimation. In REML, we construct a likelihood that depends only on θ, to remove the effect of degrees of freedom lost in the estimation of β. We use a Bayesian formulation where the fixed effect, β, and random effect, $\mathbf{b}$'s, are considered as random variables. We take a flat prior distribution for β, $p(\beta) \sim N(\beta^*, \Gamma), \Gamma \to \infty$, which is a non-informative density and the choice of β^* is immaterial. The prior for $\mathbf{b}$ is chosen as $p(\mathbf{b}) \sim N(0, Q)$.

Using the prior densities and assuming priors are independent, we can express the joint log-likelihood as,

$$\begin{aligned}\log p_\theta(Y,B) &= \log p_\theta(Y|B)p(\beta)p(\mathbf{b}) \\ &= \log p_{\Omega(\boldsymbol{\alpha})}(Y|B) + \log p_Q(\mathbf{b}) \\ &= G_1(\Omega(\boldsymbol{\alpha})) + G_2(Q)\end{aligned} \tag{33}$$

where, $G_1(\Omega(\boldsymbol{\alpha})) = \log p_{\Omega(\boldsymbol{\alpha})}(Y|B)$ and $G_2(Q) = \log p_Q(\mathbf{b})$.

Using the distributional assumptions, we can further express G_1 and G_2 as,

$$G_1(\Omega(\boldsymbol{\alpha})) = -\frac{1}{2}\sum_{i=1}^{N}\log\det(\Omega_i(\boldsymbol{\alpha})) - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i))^T\Omega_i^{-1}(\boldsymbol{\alpha})(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i)) \tag{34}$$

$$G_2(Q) = -\frac{1}{2}N\log\det(Q) - \frac{1}{2}\sum_{i=1}^{N}\mathbf{b}_i^T Q^{-1}\mathbf{b}_i. \tag{35}$$

Our objective is to solve,

$$\arg\max_{\theta}\log p_\theta(Y,B). \tag{36}$$

Since the objective function in Eq. 36 is non-convex with respect to θ, an expectation-maximization (EM) algorithm is used to learn the parameters. The EM algorithm is an iterative approach, where in each iteration, the expectation of the (log)likelihood is computed (E-step), and parameters that maximize the expectation are estimated (M-step). By working on the expectation of the likelihood, an EM algorithm becomes robust to the presence of missing values, a common phenomenon in longitudinal studies.

In the following, we develop an EM approach for parameter estimation and show that the EM approach alone is not suitable to estimate and decouple the components of $\Omega(\boldsymbol{\alpha})$. We, thus, use a Variogram estimation approach to estimate $\boldsymbol{\alpha}$ and integrate it with the EM algorithm to develop the proposed EM-Variogram estimation technique.

E-step

In the expectation step we find the expected value of the (log)likelihood in Eq. 33 with respect to the "current" parameter estimate, $\theta^{(p)} = \{\Omega(\boldsymbol{\alpha}^{(p)}), Q^{(p)}\}$, for any iteration index

p, denoted by $O(\theta|\theta^{(p)})$ and is given by $O(\theta|\theta^{(p)}) = \mathbb{E}_{\theta^{(p)}}[\log p_\theta(Y,B)] = \mathbb{E}_{\theta^{(p)}}[G_1(\Omega(\boldsymbol{\alpha}))] + \mathbb{E}_{\theta^{(p)}}[G_2(Q)]$.

For a simpler expression of the expectations, we first obtain parameters in $B = \{\beta, \mathbf{b}\}$. For a given variance parameters, $\theta^{(p)}$, the marginal distribution of $\mathbf{y}_i$ can be found by $\mathbf{y}_i = X_i\beta + \boldsymbol{\epsilon}_i^*$, where, $\boldsymbol{\epsilon}_i^* = Z_i\mathbf{b}_i + \boldsymbol{\epsilon}_i$ are normally distributed as $\boldsymbol{\epsilon}_i^* \sim N(\mathbf{0}, \Sigma_i(\theta))$ with $\Sigma_i(\theta) = \Omega_i(\boldsymbol{\alpha}) + Z_iQZ_i^T = \tau^2\mathbf{I}_i + \sigma^2\Phi_\phi(\mathbf{U}_i) + Z_iQZ_i^T$.

Therefore, marginally, $\mathbf{y}_i \sim N(X_i\beta, \Sigma_i(\theta))$. Consequently, the current estimate of β given $\theta^{(p)}$ can be computed as,

$$\beta^{(p)} = \left(\sum_{i=1}^{N} X_i^T\Sigma_i^{-1}(\theta^{(p)})X_i\right)^{-1}\sum_{i=1}^{N} X_i^T\Sigma_i^{-1}(\theta^{(p)})\mathbf{y}_i \tag{37}$$

Also, the estimate for random effects, $\mathbf{b}_i$, will be the posterior mean, given below (see Appendix A).

$$\mathbf{b}_i^{(p)} = \mathbb{E}[\mathbf{b}_i|\mathbf{y}_i] = Q^{(p)}Z_i^T\Sigma_i^{-1}(\theta^{(p)})\left(\mathbf{y}_i - X_i\beta^{(p)}\right) \tag{38}$$

The expectations of G_1 and G_2, derived in Appendix B, is thus expressed using $\beta^{(p)}$ and $\mathbf{b}^{(p)}$ as

$$\begin{aligned}\mathbb{E}_{\theta^{(p)}}[G_1(\Omega(\boldsymbol{\alpha}))] &= -\frac{1}{2}\sum_{i=1}^{N}\log\det(\Omega_i(\boldsymbol{\alpha})) - \\ &\quad \frac{1}{2}\sum_{i=1}^{N}\left[(\mathbf{e}_i^{(p)})^T\Omega_i^{-1}(\boldsymbol{\alpha})\mathbf{e}_i^{(p)} + tr\left(\Omega_i^{-1}(\boldsymbol{\alpha})cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)})\right)\right], \qquad (39)\\ \mathbb{E}_{\theta^{(p)}}[G_2(Q)] &= -\frac{1}{2}N\log\det(Q) - \frac{1}{2}\sum_{i=1}^{N}\left[(\mathbf{b}_i^{(p)})^TQ^{-1}\mathbf{b}_i^{(p)} + tr\left(Q^{-1}cov(\mathbf{b}_i|\mathbf{y}_i;\theta^{(p)})\right)\right], \qquad (40)\end{aligned}$$

where, $\mathbf{e}_i^{(p)} = \mathbf{y}_i - (X_i\beta^{(p)} + Z_i\mathbf{b}_i^{(p)}), i = 1, \ldots, N$.

The above equations, Eq. 39-40, give the objective function $O(\theta|\theta^{(p)})$ to be maximized in the next step.

M-step

In the maximization step, our objective is to estimate the parameters that maximize $O(\theta|\theta^{(p)})$. As shown earlier, this function is separated into two independent components, $\mathbb{E}_{\theta^{(p)}}[G_1(\Omega(\boldsymbol{\alpha}))]$ and $\mathbb{E}_{\theta^{(p)}}[G_2(Q)]$. Therefore, we can maximize them independently to obtain the updated parameter estimates. Direct maximization of Eq. 39 and Eq. 40 gives the following parameter estimates (see Appendix C), also called the REML robust estimates,

$$\Omega_i^{(p+1)} = \mathbf{e}_i^{(p)}(\mathbf{e}_i^{(p)})^T + cov(\epsilon_i|\mathbf{y}_i;\theta^{(p)}),\ i = 1,\ldots,N \tag{41}$$

$$Q^{(p+1)} = \frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{b}_i^{(p)}(\mathbf{b}_i^{(p)})^T + cov(\mathbf{b}_i|\mathbf{y}_i;\theta^{(p)})\right] \tag{42}$$

We use the update expression for Q in Eq. 42. However, as explained in Sec. 3.2.1, we require a parametric structure of Ω to isolate the measurement error, rendering the unconstrained estimate in Eq. 41 not useful. Although, one approach would be to estimate Ω_i's from Eq. 41, plug the estimates in Eq. 32, and solve for the variance components, but from a computational point of view, estimation of unconstrained covariance is difficult because, a) given set of T_i measurements for a unit i, we require $\frac{1}{2}|T_i|(|T_i|+1)$ parameters that should be estimated from the data, and b) the presence of many missing values. Therefore, REML robust estimates for Ω is not applicable in our problem.

We, thus, use a *Variogram* approach for estimating the covariance parameters in $\boldsymbol{\alpha}$. A Variogram represents the difference variance of a stochastic process at two spatially or temporally separated locations and can effectively model various covariance structures (Diggle et. al., 2002). Variogram methods are popularly used in the field of Geostatistics to fit a model with temporal or spatial correlations (Dutter, 2012). In the following, Variogram is used to estimate the variance parameters from given residuals in the M-step of any iteration p .

Variogram

For a given longitudinal data, we assume, $\mathbb{E}[\boldsymbol{\epsilon}(t)] = 0, var(\boldsymbol{\epsilon}(t)) = \sigma_\epsilon^2 < \infty,$ and$cov(\boldsymbol{\epsilon}(t_j), \boldsymbol{\epsilon}(t_k)) \propto |t_j - t_k|$. Thus, $\boldsymbol{\epsilon}$ is weakly or second order stationary. A variogram for such processes is defined by $2\gamma(\nu_{jk}) = var(\boldsymbol{\epsilon}(t_j) - \boldsymbol{\epsilon}(t_k)) = \mathbb{E}[(\boldsymbol{\epsilon}(t_j) - \boldsymbol{\epsilon}(t_k))^2]$, where, $\nu_{jk} = |t_j - t_k|$ and $(t_j, t_k) \in V(\nu_{jk})$. Here $V(\nu_{jk}) = \{(t_j, t_k) : |t_j - t_k| = \nu_{jk}; t_j, t_k \in \mathcal{T}\}$, which is the

set of all combinations of observation times (t_j, t_k) such that they are apart by v_{jk} (or lag difference). A sample estimate of variogram for a lag ν is obtained from residuals as $2\hat{\gamma}(\nu_{jk}) = \frac{1}{|V(\nu_{jk})|}\sum_{V(\nu_{jk})} [\mathbf{e}(t_j) - \mathbf{e}(t_k)]^2$. For the decomposed covariance structure, defined in this chapter, a theoretical expression for the variogram (Diggle et. al., 2002) is

$$\gamma_{\boldsymbol{\alpha}}(\nu) \quad = \quad \tau^2 + \sigma^2(1 - \Phi_\phi(\nu)), \tag{43}$$

where, τ^2, σ^2 and ϕ are the measurement error variance, autocorrelated noise variance and autocorrelation effect parameters, respectively. In the context of a variogram, τ^2 is also known as nugget, σ^2 as sill, and the autocorrelation effect corresponds to the *range*. Figure 12 shows example of a variogram and illustrates the notations. From Eq. 43, it can be seen that the theoretical variogram can be used to model the covariance parameters in $\boldsymbol{\alpha}$ used in Eq. 32. Therefore, estimating the theoretical variogram by sample variogram will yield the set of estimates for covariance parameters. Two types of fitting techniques are used for variogram: maximum likelihood methods and least squares methods. The least squares methods are more suitable due to their non-parametric nature, geometric interpretation, and simple framework. More specifically, we use weighted least squares method, which is also more robust to outliers and missing values.

Estimation of $\boldsymbol{\alpha}$

For a given sample variogram estimates, $\hat{\gamma}_{\boldsymbol{\alpha}^{(p)}}(\nu)$, and the theoretical variogram, $\gamma_{\boldsymbol{\alpha}}(\nu)$, we minimize the weighted least squares function, $H_{\boldsymbol{\alpha}^{(p)}}(\boldsymbol{\alpha}) = l^T_{\boldsymbol{\alpha}^{(p)}}(\boldsymbol{\alpha}) W(\boldsymbol{\alpha}) l_{\boldsymbol{\alpha}^{(p)}}(\boldsymbol{\alpha})$, where $l_{\boldsymbol{\alpha}^{(p)}}(\boldsymbol{\alpha}) = [f(2\hat{\gamma}_{\boldsymbol{\alpha}^{(p)}}(\nu_{jk})) - f(2\gamma_{\boldsymbol{\alpha}}(\nu_{jk}))]$; $\forall\{(j,k);\, t_j, t_k \in \mathcal{T}\}$, f is a transformation function and $W(\boldsymbol{\alpha})$ is a weight matrix. As shown in Cressie (1985), and Das et. al. (2012), the variance of logarithmic transformation of $2\hat{\gamma}(\nu)$ is proportional to $\frac{2}{|V(\nu)|}$. Thus, we use this transformation function and $\frac{2}{|V(\nu)|}$ as the inverse weight. This will result in

$$H_{\boldsymbol{\alpha}^{(p)}}(\boldsymbol{\alpha}) \quad = \quad \sum_{\forall\{(j,k);\, t_j, t_k \in \mathcal{T}\}} [\log(2\hat{\gamma}_{\boldsymbol{\alpha}^{(p)}}(\nu_{jk})) - \log(2\gamma_{\boldsymbol{\alpha}}(\nu_{jk}))]^2 \frac{|V(\nu_{jk})|}{2} \tag{44}$$

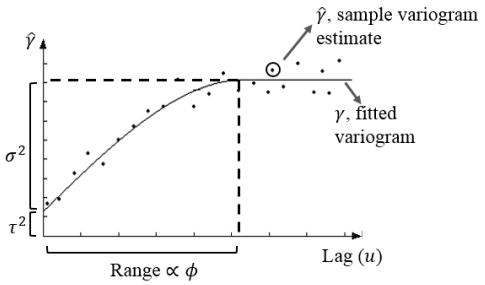


Figure 12: Illustration of components of a Variogram

As shown in Das et. al. (2012) the estimates obtained from Eq. 44 is asymptotically unbiased and has a smaller asymptotic variance. To summarize the variogram estimation, as part of the EM-Variogram integrated approach, for any iteration p, we compute the sample variogram, $\hat{\gamma}_{\boldsymbol{\alpha}^{(p)}}$, from the given residuals $\mathbf{e}^{(p)}$ and to estimate covariance parameters $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha}^{(p+1)} = \arg\min_{\boldsymbol{\alpha}} H_{\boldsymbol{\alpha}^{(p)}}(\boldsymbol{\alpha})$. We use interior point method (Mehrotra, 1992) to obtain the minima of Eq. 44. Other methods, like Newton-Raphson, Nelder Mead, etc. can also be used for the minimization.

As can be seen, the variogram approach can effectively estimate the covariance of errors, and decouple the measurement system variance, τ^2, from the noise variance, σ^2. In the next subsection, we will summarize the overall algorithm for the EM-Variogram estimation method.

3.2.3 Algorithm

The proposed EM-Variogram algorithm is given in Table 4. The estimation procedure is initiated with initial values for variance parameters in θ, given as $\theta^{(0)}$. Thereafter, for any iteration, p, we first estimate the fixed and random effects in $B^{(p)} = (\beta^{(p)}, \mathbf{b}^{(p)})$ using the variance parameters in $\theta^{(p)}$. The residuals and conditional covariance of random effects (computed using $B^{(p)}$) are used to estimate the model covariance parameters in $\theta = \{\boldsymbol{\alpha}, Q\}$ to yield $\theta^{(p+1)}$. For estimating $\theta^{(p+1)}$, a Variogram approach for $\boldsymbol{\alpha}^{(p+1)}$ and a robust REML estimate for $Q^{(p+1)}$ are used. Using the estimated $\theta^{(p+1)}$, we re-estimate the fixed and random in the next iteration, $B^{(p+1)}$. This process is repeated until convergence.

Table 4: EM-Variogram integrated algorithm

1 Initialization, $\theta^{(0)} = \{\boldsymbol{\alpha}^{(0)}, Q^{(0)}\}$

For iteration, $p = 0, 1, 2, \ldots$

2.1 Compute, $B^{(p)} = (\beta^{(p)}, \mathbf{b}_1^{(p)}, \ldots, \mathbf{b}_N^{(p)})$ from Eq. 37 and 38 using the current estimates of variance-covariance components $\theta^{(p)} = (\boldsymbol{\alpha}^{(p)}, Q^{(p)})$.

2.2 Get model residuals, $\mathbf{e}_i^{(p)} = y_i - X_i\beta^{(p)} - Z_i\mathbf{b}_i^{(p)}$ and posterior covariances $cov(\mathbf{b}_i|y_i; \theta^{(p)}) = Q^{(p)} - Q^{(p)}Z_i^T(\Omega_i(\boldsymbol{\alpha}^{(p)}) + Z_iQ^{(p)}Z_i^T)^{-1}Z_iQ^{(p)}$ where, $\Omega_i(\boldsymbol{\alpha}^{(p)})$ is given in Eq. 32.

3.1 Compute $Q^{(p+1)}$ as $Q^{(p+1)} = \frac{1}{N}\sum_{i=1}^{N}[\mathbf{b}_i^{(p)}(\mathbf{b}_i^{(p)})^T + cov(\mathbf{b}_i|y_i; \theta^{(p)})]$

3.2 Compute $\{\boldsymbol{\alpha}^{(p+1)}\}$ using variogram by solving Eq. 44 as explained in Â§3.2.2

4 Repeat 2-3 until convergence.

3.2.4 Missing values

The proposed methodology is robust to missing values in data. This property is due to the EM-Variogram technique used for estimation. Suppose we partition $\mathbf{y}_i$ into two vectors, $\mathbf{y}_i = (W_1^{(i)}, W_2^{(i)})$, where $W_1^{(i)}$ is observed and $W_2^{(i)}$ is missing. For parameter estimation in such situations, maximum likelihood method on $\mathbf{y}_i$ cannot be used. Although one alternative is to maximize the likelihood of the observed data, W_1, the likelihood becomes intractable. This is

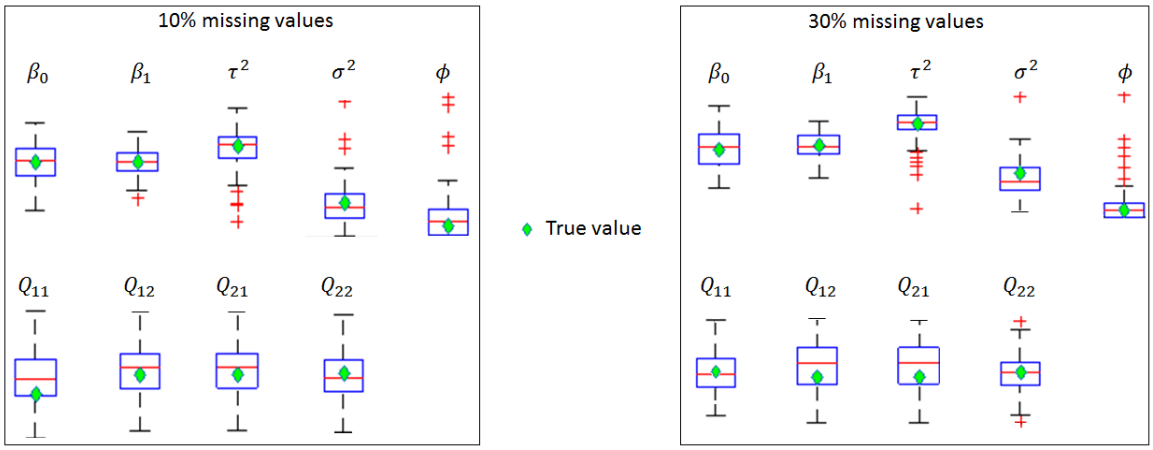
because, the likelihood estimation requires integration (or summation, if discrete variable) over the range of W_2, which becomes complex in many cases due to multidimensionality (Dempster et al., 1977). The EM approach addresses this issue by maximizing a lower bound on the likelihood, estimating the latent model parameter and repeatedly constructing a closer lower bound (See Wu, 1983).

Besides, in the proposed methodology, as part of the EM algorithm, we utilize a Variogram approach for estimating the variance components in θ. We use a weighted least squares (WLS) method for the Variogram fitting, which is also robust to missing values. Therefore, the proposed EM-Variogram estimation technique proves to be effective, accurate and extremely robust to missing values in the data. In the next section, Sec. 3.3, we will show the efficacy of the methodology and it's robustness to missing values.

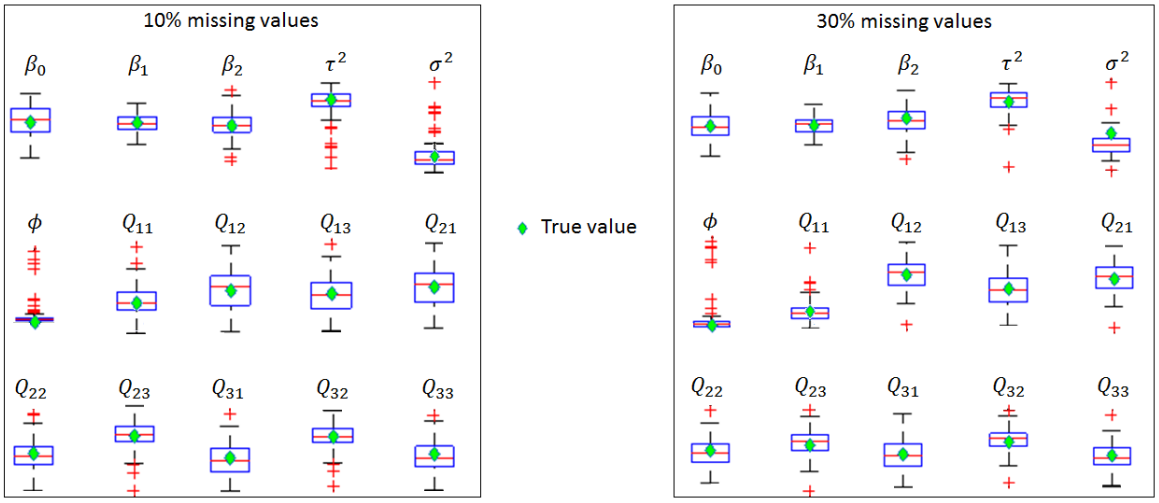
3.3 Experimental Validation

In this section, we will validate the proposed methodology using simulated data. The accuracy of the model estimates is compared with the known *true values* for the corresponding simulated scenarios. Several scenarios are simulated to study the proposed methodology's accuracy and sensitivity with respect to the following parameters: (a) different underlying models, namely, linear and quadratic models, (b) number of subjects (units), $N \in \{30, 50, 100\}$, (c) length of observations, $|T| \in \{10, 30, 60\}$, (d) the ratio of measurement system error variance, τ^2, and random noise variances, σ^2, in $\{0.1, 0.2, 0.4, 0.6\}$ and (e) the percentage of missing values in the data, $M \in \{0, 10, 20, 30\}$. This results in 288 simulation scenarios. Moreover, to mimic the real world in simulating the longitudinal data, we place missing values, a) towards the end of patients' study representing patients dropouts or premature study termination, and b) in between few observations corresponding to missed tests or mis-timed observations. In addition to computing the point estimates of the LME parameters using the EM-Variogram method, we use bootstrapping to study the dispersion of these estimates by resampling the simulated data 100 times for each scenario.

As example the results of two simulation scenarios corresponding to one linear and one quadratic model with 10% and 30% missing values for sample size of $N = 50$ and the



(a) Linear trend



(b) Quadratic trend

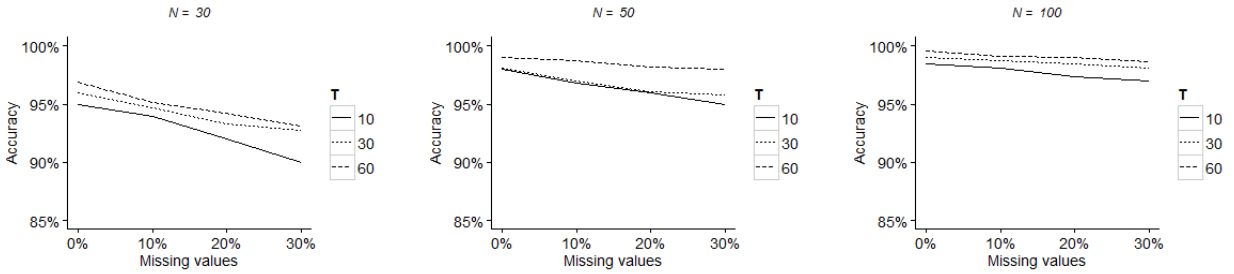
$$\beta_0, \beta_1, (\beta_2) = 10, 25, (2)$$

$$\tau^2, \sigma^2, \phi = 20, 2, 0.2$$

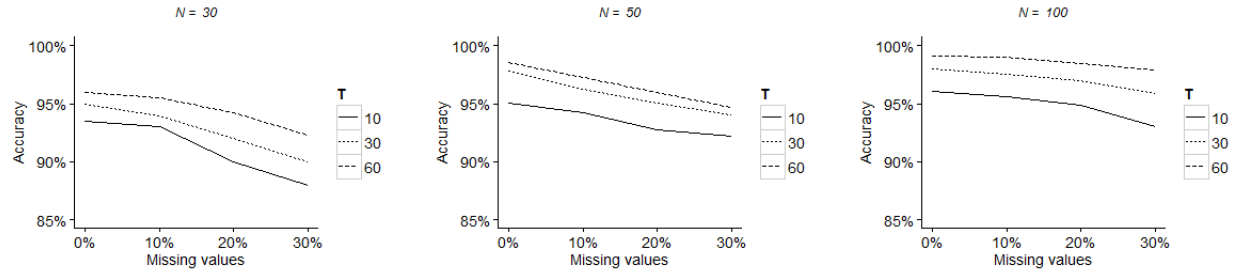
$$Q_{2(3)\times 2(3)} = \begin{bmatrix} 10 & -5 & (1) \\ 5 & 10 & (-2) \\ (1) & (-2) & (1) \end{bmatrix}$$

(c) True parameter values for above linear and quadratic (in parentheses) simulation

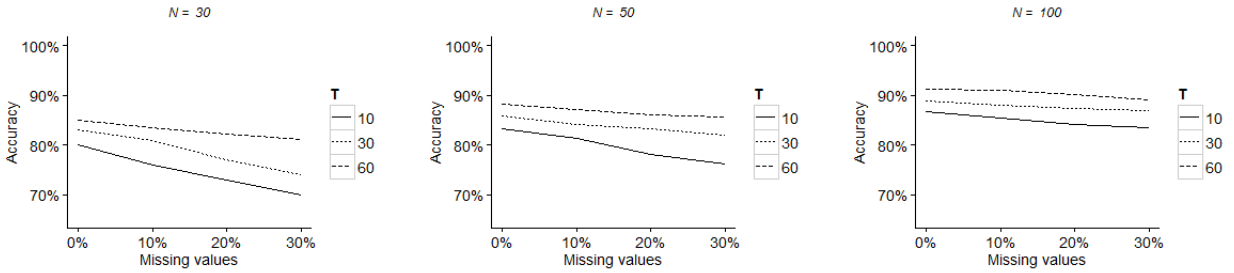
Figure 13: Estimation results for experimental evaluation of the proposed EM-Variogram method. The box plot shows the parameter estimates over bootstrapped samples, and its *true value* is indicated by a green diamond. The *true value* falls within the confidence limits of each parameter estimations – indicating the accuracy of the methodology.



(a) Slope parameter, β_1



(b) Measurement system error variance, τ^2



(c) Autocorrelation noise variance, σ^2

Figure 14: Sensitivity analysis of the methodology using various simulated scenarios. Each plot shows the average relative accuracy of the estimated parameter over various simulations. The results indicate high accuracy and the methodology's robustness to missing values.

length of observation times $|T| = 30$, are shown in Figure 13a and Figure 13b. In these figures, the box plots of estimated parameters by using the EM-Variogram method from the bootstrapped simulated data along with the true value of each parameter, marked with a diamond, are plotted. As can be seen from the figures, in all cases (except for one) the interquartile range (IQR) of the estimates includes the true value of the corresponding parameters, indicating the high accuracy. The IQR of estimates for β and τ^2 are smaller than that of other parameters, which implies more precise estimates. The box plot results for other scenarios also indicated similar estimation accuracy and precision.

Finally, we summarize the EM-Variogram's accuracy and sensitivity for the critical parameters, viz. slope coefficient, β_1, the measurement system variance, τ^2, and the noise variance, σ^2, in Figures 14a - 14c, respectively. Each subplot corresponding to a parameter has different plots for different sample sizes, N, and legends correspond to different length of observations, T . The relative accuracy is plotted against increasing amounts of missing values. Besides, the several sets of true values for the fixed effect and variance parameters were used for this analysis (and not limited to the set given in Fig. 13c).

From these figures, as expected, the accuracy decreases with increasing amount of missing values, however, the rate of the accuracy decline in most cases is low — showing robustness of the proposed method. With regard to the slope parameter, β_1, the accuracy is always high. For the adverse scenario with small sample size of $N = 30$, shorter length of observations, and higher amount of missing values, its accuracy goes below 95% but stays above 90%. In all other scenarios, its accuracy is always greater than 95%, irrespective of the length of observation and amount of missing values. Similarly, the accuracy of τ^2 goes slightly less than 90% under the same adverse scenario mentioned above. However, its accuracy is close to 95% or higher in other situations. The accuracy of σ^2, in general, is lower than β_1 and τ^2 (between ~70-90%), but follows similar behavior in sensitivity.

In short, the simulation results show the validity of the proposed EM-Variogram in accurately estimating the LME parameters fitted on longitudinal data with inherent measurement system error in various scenarios. They also indicate the robustness of the estimation method amount of missing values. In the next section, we demonstrate how the applications of the proposed methodology can impact the real world.

3.4 Case study

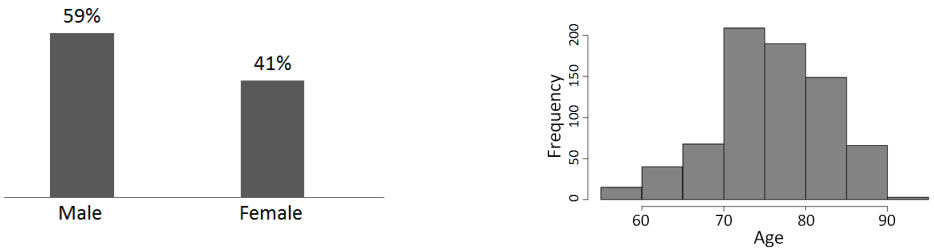
3.4.1 Problem Statement

In this case study, we used the data obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial MRI, positron emission tomography

(PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). For up-to-date information, see www.adni-info.org. MCI is a neurological condition that causes cognition decline. MCI can eventually convert to AD, a neurodegenerative disease. Both MCI and AD affect the brain structure and function. Hippocampus is an important neuroanatomical structure that is affected by both. It is associated mainly with memory, in particular long-term memory, and the decline in its volume is associated with disease progression. Thus, longitudinal hippocampal volume analysis is important in studying MCI and AD.

MRI devices can detect some of the structural changes, including hippocampal volume, caused by the diseases by scanning patients' brain. In this study, we use such MRI data, collected by ADNI. Given a set of longitudinal scans, we process the data using the FreeSurfer (Fischl, 2002, Dale, 1999, Fischl et. al., 1999), Longitudinal Stream (Reuter et. al., 2012, Reuter and Fischl, 2011) to automatically obtain hippocampal volume estimates for each hemisphere. The longitudinal stream creates an unbiased within-subject template space and image (Reuter et. al., 2012) using an inverse consistent registration method (Reuter et. al. 2010). This template is a robust representation of the average subject anatomy. Several processing steps of the FreeSurfer pipeline are then initialized for each time point with common information from the subject template to increase reliability of the automated measurements.

In addition, as subjects have different head sizes, we normalized the hippocampal volumes to account for different intra-cranial volume (ICV), where the ICV estimate is kept constant across time. The entire process involves several image processing steps that may contribute to the measurement system error. In addition, the image acquisition step may also introduce some variability (due to patient motion, hydration, etc.) and contribute to the measurement system error. This increases the variance of the observations, which can make it challenging to find statistically significant trends. Better statistical estimation and powerful hypothesis testing on the hippocampal atrophy can be made by isolating the variance of measurement system error. Therefore, in this study, we use the proposed methodology on the longitudinal



(a) Male-female ratio

(b) Age distribution

Figure 15: Patient demographics

hippocampus volumes, and compare the results with the traditional LME method.

3.4.2 Data

A set of 741 patients (59% males, 41% female, age 75±6.9) with MCI and 215 control subjects (OC) are selected for this study. A subset of MCI patients ($N = 140$) eventually developed AD, which are called MCI converters (MCI-C). The rest of MCI subjects ($N = 217$) are called MCI non-converters (MCI-NC). The patient demographics are shown in Figure 15. As can be seen in Figure 15b, the majority of patients are above 70 years old. The length of observations on the patients goes up to 4 years, with observations made at approximately 6-month intervals. There is about 35% missing observations. Normalized left and right hippocampal volumes are generated and used to distinguish between MCI-C, MCI-NC, and OC.

3.4.3 Results

Hippocampal volume reduction

Hippocampal volume reduction rates (slopes) are estimated for the OC, MCI-NC and MCI-C groups separately. We, then, test the statistical significance of the slopes and report the (log) p-values in Figure 16. When computing the p-values using the traditional LME method, the confounded variance estimates are used, while the proposed EM-Variogram method allows us to decouple and remove the measurement system variance from the confounded variance, resulting in a more accurate (smaller) estimate for the noise variance. As can be seen in Figure 16, the (log) p-values of proposed methodology is significantly smaller than the traditional LME method. Since multiple comparisons are frequently performed

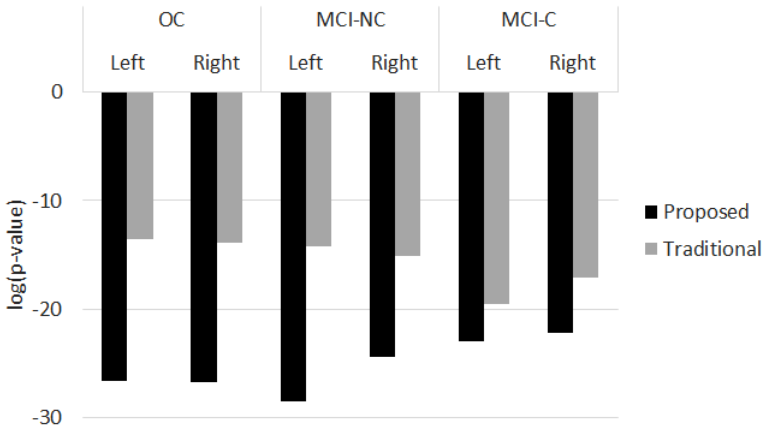


Figure 16: Comparison of p-values of degeneration rate estimation from proposed and traditional LME method.

in such studies, the significance level of the hypothesis testing is accordingly adjusted (reduced) in practice. In such cases, the proposed methodology will be more capable in finding statistically significant estimates.

This increased sensitivity is due to the isolation of measurement system variance. Another benefit of such increased sensitivity is that it allows to reduce the sample size and, thus, the costs of longitudinal studies. Furthermore, it enables to detect more subtle effects, which is necessary during treatment assessment or to quantify early changes during pre-symptomatic phases, where disease modifying interventions may be most effective.

Comparison of degeneration effects

Next, we analyze differences in degeneration rates of MCI-NC patients with MCI-C, and OC patients. Figure 17 shows the (log) p-values of their differences. We can see that the proposed method detects the difference with much higher confidence compared to the traditional LME method. These results demonstrate that hippocampal volume loss is an excellent marker for disease progression, as it is capable of differentiating the various groups at different stages. Finding these markers is important as they can then be used to quantify effects of disease modifying therapies or for computer-aided diagnosis. Overall, these results are quite useful for medical studies, like, differentiating various groups at different stages.

Degeneration estimation

Finally, we take a subset of the data containing only the first five observations. We, then,

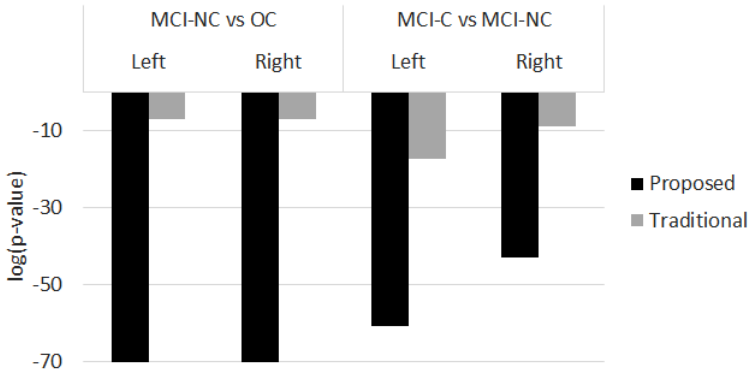
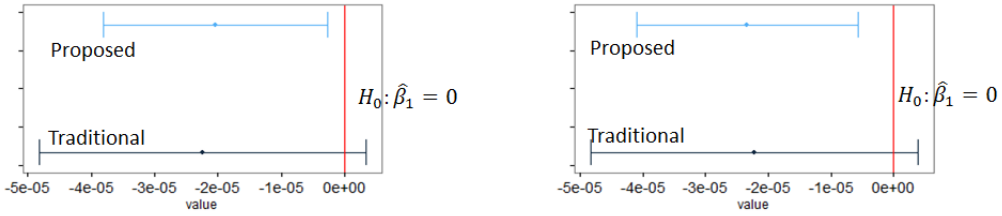


Figure 17: Comparison of degeneration effects



(a) Left-Hippocampus

(b) Right-Hippocampus

Figure 18: Degeneration rate estimates with confidence intervals

filter patients with MCI-NC who are less than 65 years in age ($N = 28$). In a typical real world scenario, we have such smaller data samples to detect degeneration. We want to see if the proposed methodology is capable of early detection of the degeneration. Figures 18a-18b show the degeneration rate estimates with its 95% confidence intervals for left and right hippocampus. The confidence intervals are obtained by using the bootstrap method with 100 resampling. We can see that the traditional method gives a wider confidence interval, thus, there is no statistically significant degeneration (null hypothesis, H_0, cannot be rejected). On the other hand, the proposed methodology isolates and remove the measurement error, and hence, gives narrower confidence intervals indicating that hippocampal atrophy is statistically significant (H_0 is rejected).

This implies that the methodology can do early detection of Alzeihmer disease. This early detection of disease effects can potentially lead to initiation of timely treatment and, thus, help patients to postpone dementia effects or remain functioning for a longer period.

In this case study, we performed a real data analysis to demonstrate the superior performance of the proposed method over traditional LME in detecting a statistically significant effect, and detecting differences between disease conditions. We also show that it allows us to make accurate inferences from a smaller dataset, which can help in reducing costs of study and early detection.

3.5 Discussion and Conclusion

Existing methods rely on a complete sanctity of data for analyzing them and drawing inferences. This can be problematic if the available data has measurement errors, especially in medical decision making. This research focused on analysis of longitudinal data in a challenging scenario, when there are measurement errors but replicates are not available. Our major contribution was development of a new EM-Variogram technique for estimating an extended linear mixed effect model with a parametric covariance structure. The parametric covariance expression decouples the measurement error variance from the overall variance. The developed estimation technique, thus, isolates the measurement error, and hence, provides a more accurate effect estimation and statistical inferences.

The performance of the model was experimentally validated via simulations for various scenarios. We find that the methodology is effective and accurate in modeling, and is robust to missing values, commonly found in a longitudinal data. The methodology was also applied to a longitudinal MRI dataset for evaluation of hippocampal volume in (potential) Alzheimer disease patients with mild cognitive impairment (MCI). The measurement error variance was accurately decoupled from the noise variance. After isolating the measurement error, we were able to obtain more precise (narrower) confidence interval for the effect estimates, leading to more powerful statistical tests. Moreover, it can also detect more subtle effects with less data and, thus, can a) do an early detection of a patient's condition allowing necessary treatment assessment and diagnosis at early stages of the disease, and b) reduce costs of longitudinal studies.

Besides, the numerical experiments show that the proposed method may have relatively poor performance in random effect variance estimations if the random noise in the model is

relatively high. However, this situation typically indicates that the selected model is much different from the true underlying model, and thus, can be resolved with a better model selection. In practical applications, one can find a proper choice for the underlying model by trying a wide range of model and comparing them using a model selection criterion, like, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), etc. Moreover, an effective model selection is also necessary to keep the modeling error minimal, an underlying assumption of the developed approach.

In conclusion, the experimental and the case study results indicate that the proposed methodology can effectively address the analysis problem for longitudinal data where replicated measurement are very costly or difficult to take, for example drug efficacy tests, destructive tests, etc. Besides, the proposed method is flexible and can be used in cases where other covariates (e.g., sex, age, etc.) besides time are included in the model.

In this chapter, we make assumptions on the homogeneity of measurement device conditions over time, which makes the reproducibility analysis irrelevant. However, the extension of the proposed methodology for a simultaneous gauge repeatability and reproducibility (GRR) analysis would be a topic of interest for future research. Another potential direction for future research, is to develop methods for GRR study of multivariate longitudinal data streams.

3.5.1 Acknowledgement

This work was supported in part by the National Institutes of Health under Advanced Multimodal Neuroimaging Training Program under Grant R90-DA023427.

Support for this research was also provided in part by the National Cancer Institute (1K25-CA181632-01), the Genentech Foundation, the National Science Foundation (NSF-CMMI 1451088), and the NVIDIA corporation. Further support was provided by the A.A. Martinos Center for Biomedical Imaging (P41RR014075, P41EB015896, U24RR021382), and was made possible by the resources provided by Shared Instrumentation Grants 1S10RR023401, 1S10RR019307, and 1S10RR023043.

The collection and sharing of the MRI data used in the group study based on ADNI was

funded by the Alzheimer's Disease Neuroimaging Initiative (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: Alzheimer's Association; Alzheimer's Drug Discovery Foundation; BioClinica, Inc.; Biogen Idec Inc.; Bristol-Myers Squibb Company; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; GE Healthcare; Innogenetics, N.V.; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Medpace, Inc.; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Synarc Inc.; and Takeda Pharmaceutical Company. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

3.6 APPENDICES

3.6.1 Appendix A: Posterior mean and covariances

Here we will derive the posterior mean and covariance of $\mathbf{b}_i$ and posterior covariance of $\boldsymbol{\epsilon}_i$. We will require the following identity for the derivation.

For any random variables, w_1 and w_2, such that their joint distribution is normal as Eq. 45,

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \tag{45}$$

then, the conditional distribution is given as,

$$w_2|w_1 \quad \sim \quad N\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(w_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{1}\Sigma_{12}\right) \tag{46}$$

We use Eq. 30 and drop the index i for a simpler expression, as,

$$\mathbf{y} \quad = \quad X\beta + Z\mathbf{b} + \boldsymbol{\epsilon} \tag{47}$$

Note that we can jointly express $\mathbf{y}$ and $\mathbf{b}$ as,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{b} \end{bmatrix} \quad = \quad \begin{bmatrix} X\beta \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} Z & I \\ I & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{48}$$

where I and $\mathbf{0}$ are identity and zero matrices, respectively.

Also,

$$cov\left(\begin{bmatrix} \mathbf{b} \\ \boldsymbol{\epsilon} \end{bmatrix}\right) \quad = \quad \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \Omega \end{bmatrix} \tag{49}$$

Therefore, from Eqs. 48-49,

$$\begin{aligned} \begin{bmatrix} \mathbf{y} \\ \mathbf{b} \end{bmatrix} \quad &\sim \quad N\left(\begin{bmatrix} X\beta \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} Z & I \\ I & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \Omega \end{bmatrix} \begin{bmatrix} Z & I \\ I & \mathbf{0} \end{bmatrix}^T\right) \\ &=^d \quad N\left(\begin{bmatrix} X\beta \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} ZQZ^T + \Omega & ZQ \\ QZ^T & Q \end{bmatrix}\right) \end{aligned} \tag{50}$$

Thus, from the identity in Eq. 46

$$\begin{aligned} \mathbb{E}[\mathbf{b}|\mathbf{y}] \quad &= \quad \mathbf{0} + QZ^T(ZQZ^T + \Omega)^{-1}(\mathbf{y} - X\beta) \\ &= \quad QZ^T\Sigma^{-1}(\mathbf{y} - X\beta) \end{aligned} \tag{51}$$

where, $\Sigma = \Omega + ZQZ^T$, also used in E-Step of Sec. 3.2.2.

The covariance is similarly found using the identity as,

$$cov(\mathbf{b}|\mathbf{y}) \quad = \quad Q - QZ^T(ZQZ^T + \Omega)^{-1}ZQ \tag{52}$$

The covariance of $\boldsymbol{\epsilon}$ can be similarly found by jointly expressing $\mathbf{y}$ and $\boldsymbol{\epsilon}$ as,

$$\begin{bmatrix} \mathbf{y} \\ \boldsymbol{\epsilon} \end{bmatrix} \quad = \quad \begin{bmatrix} X\beta \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} Z & I \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{53}$$

$$\begin{aligned} \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\epsilon} \end{bmatrix} \quad &\sim \quad N\left(\begin{bmatrix} X\beta \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} Z & I \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \Omega \end{bmatrix} \begin{bmatrix} Z & I \\ \mathbf{0} & I \end{bmatrix}^T \right) \\ &=^d \quad N\left(\begin{bmatrix} X\beta \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} ZQZ^T + \Omega & \Omega \\ \Omega & \Omega \end{bmatrix} \right) \end{aligned} \tag{54}$$

Thus, using identity Eq. 46 again on Eq. 54, we get,

$$cov(\boldsymbol{\epsilon}|\mathbf{y}) \quad = \quad \Omega - \Omega(ZQZ^T + \Omega)^{-1}\Omega \tag{55}$$

3.6.2 Appendix B: Expectations for EM algorithm

In this appendix, we will derive the expectations of G_1 and G_2 for the E-step in Sec. 3.2.2. For deriving the expectations of G_1 and G_2, we will require the following identity:

$$\mathbb{E}\left[\mathbf{x}^T\mathbf{A}\mathbf{x}\right] \quad = \quad \mathbf{m}^T\mathbf{A}\mathbf{m} + tr(\mathbf{A}\Sigma) \tag{56}$$

where, $\mathbf{x}$ is a stochastic vector with $\mathbb{E}[\mathbf{x}] = \mathbf{m}$ and $cov(\mathbf{x}) = \Sigma$, and $\mathbf{A}$ is a symmetric matrix.

$$\begin{aligned}
\mathbb{E}_{\theta^{(p)}}\left[G_1(\Omega(\boldsymbol{\alpha}))\right] &= \mathbb{E}_{\theta^{(p)}}\left[-\frac{1}{2}\sum_{i=1}^{N}\log\det(\Omega_i(\boldsymbol{\alpha})) - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i))^T\Omega_i^{-1}(\boldsymbol{\alpha})(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i))\right. \\
&= -\frac{1}{2}\sum_{i=1}^{N}\mathbb{E}_{\theta^{(p)}}\left[\log\det(\Omega_i(\boldsymbol{\alpha}))\right] - \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}_{\theta^{(p)}}\left[(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i))^T\Omega_i^{-1}(\boldsymbol{\alpha})(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i))\right] \\
&= -\frac{1}{2}\sum_{i=1}^{N}\log\det(\Omega_i(\boldsymbol{\alpha})) - \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}_{\theta^{(p)}}\left[(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i))^T\Omega_i^{-1}(\boldsymbol{\alpha})(\mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i))\right]
\end{aligned}$$

We have,

$$\boldsymbol{\epsilon}_i = \mathbf{y}_i - (X_i\beta + Z_i\mathbf{b}_i) \tag{58}$$

with,

$$\mathbb{E}_{\theta^{(p)}}[\boldsymbol{\epsilon}_i] = \mathbf{y}_i - (X_i\beta^{(p)} + Z_i\mathbf{b}_i^{(p)}) = \mathbf{e}_i^{(p)} \tag{59}$$

Using Eq. 56 and Eq. 58-59 in Eq. 57, we get,

$$\begin{aligned}
\mathbb{E}_{\theta^{(p)}}\left[G_1(\Omega(\boldsymbol{\alpha}))\right] &= -\frac{1}{2}\sum_{i=1}^{N}\log\det(\Omega_i(\boldsymbol{\alpha})) - \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}_{\theta^{(p)}}\left[\boldsymbol{\epsilon}^T\Omega_i^{-1}(\boldsymbol{\alpha})\boldsymbol{\epsilon}\right] \\
&= -\frac{1}{2}\sum_{i=1}^{N}\log\det(\Omega_i(\boldsymbol{\alpha})) - \\
&\quad \frac{1}{2}\sum_{i=1}^{N}\left[(\mathbf{e}_i^{(p)})^T\Omega_i^{-1}(\boldsymbol{\alpha})\mathbf{e}_i^{(p)} + tr\left(\Omega_i^{-1}(\boldsymbol{\alpha})cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)})\right)\right]
\end{aligned} \tag{60}$$

Similarly, for finding expectation of G_2we have,

$$\mathbb{E}_{\theta^{(p)}}[\mathbf{b}_i] = \mathbf{b}_i^{(p)} \tag{61}$$

where $\mathbf{b}_i^{(p)} = Q^{(p)}Z_i^T\Sigma_i^{-1}(\theta^{(p)})(\mathbf{y}_i - X_i\beta^{(p)})$ as given in Sec. 3.2.2.

The expectation of G_2 is, thus,

$$\mathbb{E}_{\theta^{(p)}}\left[G_2(Q)\right] = -\frac{1}{2}N\log\det(Q) - \frac{1}{2}\sum_{i=1}^{N}\left[(\mathbf{b}_i^{(p)})^T Q^{-1}\mathbf{b}_i^{(p)} + tr\left(Q^{-1}cov(\mathbf{b}_i|\mathbf{y}_i;\theta^{(p)})\right)\right] \quad (62)$$

3.6.3 Appendix C: REML robust estimates

In this appendix, we will derive the REML robust estimates for EM algorithm shown in Eq. 41-42. We will require the following identities for the derivation.

For any vector **a**,**b**, a square and invertible matrix **X**, and any matrix **B**,

$$\frac{\partial}{\partial\mathbf{X}}\log|\det(\mathbf{X})| = (\mathbf{X}^{-1})^T = (\mathbf{X}^T)^{-1} \quad (63)$$

$$\frac{\partial}{\partial\mathbf{X}}\mathbf{a}^T\mathbf{X}^{-1}\mathbf{b} = -(\mathbf{X}^{-1})^T\mathbf{a}\mathbf{b}^T(\mathbf{X}^{-1})^T \quad (64)$$

$$\frac{\partial}{\partial\mathbf{X}}tr(\mathbf{X}^{-1}\mathbf{B}) = -(\mathbf{X}^{-1}\mathbf{B}\mathbf{X}^{-1})^T \quad (65)$$

We maximize Eq. 39 in Sec. 3.2.2 to obtain the REML robust estimate for covariance of unit i. This is done by differentiating Eq. 39 and equating it to 0. Moreover, the covariance here is unconstrained, thus, we do not use the parameter $\boldsymbol{\alpha}$ for representing Ω.

$$\frac{\partial}{\partial\Omega_i}\mathbb{E}_{\theta^{(p)}}\left[G_1(\Omega)\right] = \frac{\partial}{\partial\Omega_i}\left(-\frac{1}{2}\sum_{i=1}^{N}\log\det(\Omega_i) - \frac{1}{2}\sum_{i=1}^{N}\left[(\mathbf{e}_i^{(p)})^T\Omega_i^{-1}\mathbf{e}_i^{(p)} + tr\left(\Omega_i^{-1}cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)})\right)\right]\right) \quad (66)$$

Using identity in Eq. 63 and Ω_i being square, invertible and symmetric matrix, we get,

$$\frac{\partial}{\partial\Omega_i}\log\det(\Omega_i) = \Omega_i^{-1} \quad (67)$$

Furthermore, using identity in Eq. 64, we get,

$$\frac{\partial}{\partial \Omega_i}\left((\mathbf{e}_i^{(p)})^T \Omega_i^{-1} \mathbf{e}_i^{(p)}\right) = -\Omega^{-1}\mathbf{e}_i^{(p)}(\mathbf{e}_i^{(p)})^T\Omega^{-1} \tag{68}$$

Finally, we use identity in Eq. 65 to get,

$$\frac{\partial}{\partial \Omega_i} tr\left(\Omega_i^{-1} cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)})\right) = -\left(\Omega_i^{-1} cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)})\Omega_i^{-1}\right)^T = -\Omega_i^{-1} cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)})\Omega_i^{-1} \tag{69}$$

Plugging in results in Eq. 67-69 into Eq. 66, and equating it to 0,

$$-\frac{1}{2}\Omega_i^{-1} + \frac{1}{2}\left[\Omega_i^{-1}\left(\mathbf{e}_i^{(p)}(\mathbf{e}_i^{(p)})^T + cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)})\right)\Omega_i^{-1}\right] = 0$$

we obtain the estimate for Ω_i as,

$$\hat{\Omega}_i = \mathbf{e}_i^{(p)}(\mathbf{e}_i^{(p)})^T + cov(\boldsymbol{\epsilon}_i|\mathbf{y}_i;\theta^{(p)}) \tag{70}$$

Moreover, it is straightforward to show $\frac{\partial^2}{\partial \Omega_i^2}\mathbb{E}_{\theta^{(p)}}\left[G_1(\Omega)\right] < 0$ to prove $\hat{\Omega}_i$ gives the maxima.

Similar to above, we maximize Eq. 40 to obtain the estimate for Q.

$$\begin{aligned}\frac{\partial}{\partial Q}\mathbb{E}_{\theta^{(p)}}\left[G_2(Q)\right] &= \frac{\partial}{\partial Q}\left(-\frac{1}{2}N\log\det(Q) - \frac{1}{2}\sum_{i=1}^{N}\left[(\mathbf{b}_i^{(p)})^T Q^{-1}\mathbf{b}_i^{(p)} + tr\left(Q^{-1}cov(\mathbf{b}_i|\mathbf{y}_i;\theta^{(p)})\right)\right]\right) \\ &= -\frac{1}{2}NQ^{-1} + \frac{1}{2}Q^{-1}\left[\sum_{i=1}^{N}\left(\mathbf{b}_i^{(p)}(\mathbf{b}_i^{(p)})^T + cov(\mathbf{b}_i|\mathbf{y}_i;\theta^{(p)})\right)\right]Q^{-1} = 0\end{aligned}$$

Thus,

$$\hat{Q} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{b}_i^{(p)}(\mathbf{b}_i^{(p)})^T + cov(\mathbf{b}_i|\mathbf{y}_i;\theta^{(p)})\right) \tag{71}$$

3.7 References

Abrevaya, J., & Hausman, J. A. (2004). Response error in a transformation model with an application to earnings-equation estimation. The Econometrics Journal, 7(2), 366-388.

Ballou, D., Madnick, S., & Wang, R. (2003). Special section: assuring information quality. Journal of Management Information Systems, 9-11.

Barnhart, H. X., Haber, M. J., & Lin, L. I. (2007). An overview on assessing agreement with continuous measurements. Journal of biopharmaceutical statistics, 17(4), 529-569.

Bland, J. M., & Altman, D. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. The lancet, 327(8476), 307-310.

Bland, J. M., & Altman, D. G. (2003). Applying the right statistics: analyses of measurement studies. Ultrasound in obstetrics & gynecology, 22(1), 85-93.

Borror, C. M., Montgomery, D. C., & Runger, G. C. (1997). Confidence intervals for variance components from gauge capability studies. Quality and Reliability Engineering International, 13(6), 361-369.

Burdick, R. K., Borror, C. M., & Montgomery, D. C. (2003). A review of methods for measurement systems capability analysis. Journal of Quality Technology, 35(4), 342.

Burdick, R. K., & Larsen, G. A. (1997). Confidence intervals on measures of variability in R&R studies. Journal of Quality Technology, 29(3), 261.

Carroll, R. J., Ruppert, D., Stefanski, L. A., & Crainiceanu, C. M. (2006). Measurement error in nonlinear models: a modern perspective. CRC press.

Collette, L., Sylvester, R. J., Stenning, S. P., Fossa, S. D., Mead, G. M., de Wit, R., ... & Kaye, S. B. (1999). Impact of the treating institution on survival of patients with "poor-prognosis" metastatic nonseminoma. Journal of the National Cancer Institute, 91(10), 839-846.

Cressie, N. (1985). Fitting variogram models by weighted least squares. Journal of the International Association for Mathematical Geology, 17(5), 563-586.

Dale, A. M., Fischl, B., & Sereno, M. I. (1999). Cortical surface-based analysis: I. Segmentation and surface reconstruction. Neuroimage, 9(2), 179-194.

Das, S., Rao, T. S., & Boshnakov, G. N. (2012). On the estimation of parameters of

variograms of spatial stationary isotropic random processes. Technical report, School of Mathematics. The University of Manchester.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological), 1-38.

Diggle, P., Heagerty, P., Liang, K. Y., & Zeger, S. (2002). *Analysis of longitudinal data*. Chapter 5. Oxford University Press.

Dutter, R. (2012). Variograms in Geostatistics. Robust Statistics, Data Analysis, and Computer Intensive Methods: In Honor of Peter Huber's 60th Birthday, 109, 153.

Dominici, F., Zeger, S. L., & Samet, J. M. (2000). A measurement error model for time-series studies of air pollution and mortality. Biostatistics, 1(2), 157-175.

Fahrmeir, L., & Tutz, G. (2013). Multivariate statistical modelling based on generalized linear models. Springer Science & Business Media.

Faraway, J. J. (2005). *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press.

Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., and others. (2002). Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. Neuron, 33(3), 341-355.

Fischl, B., Sereno, M. I., & Dale, A. M. (1999). Cortical surface-based analysis: II: inflation, flattening, and a surface-based coordinate system. Neuroimage, 9(2), 195-207.

Fu, Q., Li, Y., Liu, G., & Li, H. (2012). Low cycle fatigue behavior of AZ91D magnesium alloy containing rare-earth Ce element. Procedia Engineering, 27, 1794-1800.

Gilks, W. R., Wang, C. C., Yvonnet, B., & Coursaget, P. (1993). Random-effects models for longitudinal data using Gibbs sampling. Biometrics, 441-453.

Halligan, S. (2002). Reproducibility, repeatability, correlation and measurement error. The British journal of radiology, 75(890), 193-194.

Higgins, K. M., Davidian, M., & Giltinan, D. M. (1997). A two-step approach to measurement error in time-dependent covariates in nonlinear mixed-effects models, with application to IGF-I pharmacokinetics. Journal of the American Statistical association, 92(438),

436-448.

Hyslop, D. R., & Imbens, G. W. (2001). Bias from classical and other forms of measurement error. Journal of Business & Economic Statistics, 19(4), 475-481.

Kahn, B. K., Strong, D. M., & Wang, R. Y. (2002). Information quality benchmarks: product and service performance. Communications of the ACM, 45(4), 184-192.

Laird, N. M., & Ware, J. H. (1982). Random-effects models for longitudinal data. Biometrics, 963-974.

Lexell, J. E., & Downham, D. Y. (2005). How to assess the reliability of measurements in rehabilitation. American journal of physical medicine & rehabilitation, 84(9), 719-723.

Llovet, J. M., Bustamante, J., Castells, A., Vilana, R., Ayuso, M. D. C., Sala, M., ... & Bruix, J. (1999). Natural history of untreated nonsurgical hepatocellular carcinoma: rationale for the design and evaluation of therapeutic trials. Hepatology, 29(1), 62-67.

McCullagh P, Nelder JA. *Generalized linear models*, 2nd ed. London: Chapman & Hall; 1989. 511 p.

Mehrotra, S. (1992). On the implementation of a primal-dual interior point method. SIAM Journal on optimization, 2(4), 575-601.

Montgomery, D.C. (2001). *Design and Analysis of Experiments*, 5th ed., Wiley, New York.

Montgomery, D. C. (2007). *Introduction to statistical quality control.* John Wiley & Sons.

Montgomery, D. C., & Runger, G. C. (1993a). Gauge capability and designed experiments. Part I: basic methods. Quality Engineering, 6(1), 115-135.

Montgomery, D. C., & Runger, G. C. (1993b). Gauge capability analysis and designed experiments. Part II: experimental design models and variance component estimation. Quality Engineering, 6(2), 289-305.

MSA Manual (2010). *Measurement System Analysis (MSA) manual.* 4th Edition. AIAG.

Nelder JA, Wedderburn RWM. *Generalized linear models.* J R Stat Soc [A] 1972;135:370–384.

Padhya, K.T., Marrero, J.A., Singal, A.G. (2013). Recent advances in the treatment of hepatocellular carcinoma. Curr Opin Gastroenterol 29: 285–292.

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. Communications of the ACM, 45(4), 211-218.

Rabe-Hesketh, S., & Skrondal, A. (2008). Multilevel and longitudinal modeling using Stata. STATA press.

Rao Chaganty, N., & Joe, H. (2004). Efficiency of generalized estimating equations for binary responses. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66(4), 851-860.

Redman, T. C. (1998). The impact of poor data quality on the typical enterprise. Communications of the ACM, 41(2), 79-82.

Reuter, M., & Fischl, B. (2011). Avoiding asymmetry-induced bias in longitudinal image processing. Neuroimage, 57(1), 19-21.

Reuter, M., Rosas, H. D., & Fischl, B. (2010). Highly accurate inverse consistent registration: a robust approach. Neuroimage, 53(4), 1181-1196.

Reuter, M., Schmansky, N. J., Rosas, H. D., & Fischl, B. (2012). Within-subject template estimation for unbiased longitudinal image analysis. Neuroimage, 61(4), 1402-1418.

Singal, A. G., Pillai, A., & Tiro, J. (2014). Early detection, curative treatment, and survival rates for hepatocellular carcinoma surveillance in patients with cirrhosis: a meta-analysis. PLoS medicine, 11(4), e1001624.

Verbeke, G., & Molenberghs, G. (2009). *Linear mixed models for longitudinal data*. Springer Science & Business Media.

Wang, L. (2004). Estimation of nonlinear models with Berkson measurement errors. Annals of Statistics, 2559-2579.

Wang, N., Lin, X., Gutierrez, R. G., & Carroll, R. J. (1998). Bias analysis and SIMEX approach in generalized linear mixed measurement error models. Journal of the American Statistical Association, 93(441), 249-261.

Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. Journal of management information systems, 5-33.

Wu, C. J. (1983). On the convergence properties of the EM algorithm. The Annals of statistics, 95-103.

Wu, L. (2002). A joint model for nonlinear mixed-effects models with censoring and covariates measured with error, with application to AIDS studies. Journal of the American Statistical Association, 97, 955–964.

Zeger, S. L., & Karim, M. R. (1991). Generalized linear models with random effects; a Gibbs sampling approach. Journal of the American statistical association, 86(413), 79-86.

(

CHAPTER IV

SEQUENCE GRAPH TRANSFORM (SGT): A FEATURE EXTRACTION FUNCTION FOR SEQUENCE DATA MINING

4.1 Introduction

A sequence can be defined as a contiguous chain of discrete *alphabets*, where an alphabet can be an event, a value or a symbol, sequentially tied together in a certain order, for eg., `BAABCCADECDBBA`. Sequences are one of the most common data types found in diverse fields like social science, web, healthcare, bioinformatics, marketing, text mining etc. Some examples of sequences are, web logs, music listening history, patient movements through hospital wards, DNA, RNA and protein sequences in bioinformatics.

This ubiquitous presence of sequence data has made development of new sequence analysis methods important. Few of its motivating applications are, a) understanding users behavior from their web-surfing and buying sequences data, to serve them better advertisements, product placements, promotions, and so on, b) assessing process flows (sequences) in a hospital to find the expected patient movement based on her diagnostic profile, to better optimize the hospital resource and service, c) analysis of biological sequences to understand human evolution, physiology and diseases, etc.

However, the existing sequence data mining methods lack in their effectiveness due to absence of a good measure of (dis)similarity between sequences. For a (dis)similarity measure, almost every mainstream data mining methods use a *distance* between objects in a Euclidean space. For example, in k-means clustering the *distance* between objects (data points) within a cluster are minimized while *distance* between clusters are maximized, in classification models, like SVM or logistic regression, the *distance* of a boundary is minimized or maximized from the objects. Moreover, in deep learning realm as well, objects are transformed into euclidean vectors for comparisons. It has been commonly accepted that a euclidean space *distance* between objects is one of the best measures for (dis)similarity.

But, since sequences are an unstructured data — made of arbitrarily placed alphabets for any arbitrary length —, their representation in a euclidean space is not obvious. This devoid sequence data analysis from the mainstream data mining methods, and causing sequence mining methods to be less effective, in terms of accuracy, complexity and interpretability.

To that end, feature extraction of sequences in a finite-dimensional euclidean feature space is necessary. This can also be viewed as an embedding space, where the objective is to transform a sequence into a feature vector, such that the features capture the sequence characteristics. Besides, by definition, the embedding space has the same dimension for all sequences in a data corpus. This will facilitate computation of (dis)similarity between two sequences by measuring the *distance* between their embeddings.

Several researchers proposed some feature set or function (see references in Kumar et al., 2012) based on a fundamental premise: *a sequence is characterized by the patterns formed due to the alphabets' positions relative to others*. However, to work with this premise some of them make strong assumptions, which are not always valid. For example, Markovian models (Ranjan et al., 2015) typically make a first-order Markov assumption on sequence generation process. On the other hand, some methods perform information abstraction by taking n-grams or substrings, that can potentially lead to loss of information or inclusion of noise. The existing methods and their shortcomings are elaborated further in the subsequent subsection (Sec. 4.1.1).

We, therefore, developed a new feature extraction approach, which we call Sequence Graph Transform (SGT), that works on the same fundamental premise but without any restrictive assumption. Importantly, it effectively captures the sequence features and embeds it into a finite-dimensional euclidean space that leads to an accurate comparison of sequences by measuring the *distance* between them in the feature space.

In the following, we discuss the related work (Sec. 4.1.1) and then give the problem specifications, viz. the sequence mining challenges, types of problems and the scope of proposed approach, in Sec. 4.1.2.

4.1.1 Related Work

There are several literature in the sequence mining realm; a detailed survey can be found in Kumar et al. (2012), Gaber (2009), Dong and Pei (2007), and references therein. In this section, we will briefly go through the broad categories of the literature and mention their shortcomings.

Early research works used to find *edit*-distances between sequences after alignment. Methods for global alignment, local alignment, with or without overlapping were developed by Needleman and Wunsch, 1970, Smith and Waterman, 1981. Subsequently, a few heuristic approaches were proposed based on alignment techniques that can work on a larger dataset (BLAST, Altschul et al., 1997, and FASTA, Pearson, 1990). More recently, multiple sequence alignment techniques were developed (UCLUST, Edgar, 2010; CD-HIT, Fu et al., 2012; and MUSCLE, Edgar, 2004). These methods were developed with a focus on bioinformatics sequence problems, and severely lack in their general applicability (*generality*) due to difficulty in tuning, high computational complexity, and inability to work on sequences with significantly varying lengths. Besides, these methods do not provide any feature representation of sequences.

Few researchers worked on sequence features extraction for an embedding in the Euclidean space (Linial et al., 1997, Ding and Dubchak, 2001). But their methods are *ad-hoc* feature spaces developed for protein sequences with no theoretical support, thus, difficult to extend to a general problem.

More universally applicable and relatively powerful methods work on one of the following broad assumptions, i) sequence process has an underlying parametric distribution, ii) similar sequences have common substrings, and iii) sequence evolves from hidden strings.

The parametric methods typically make a Markovian distribution assumptions, more specifically a first-order Markov property, on the sequence process (Cadez et al., 2003 and Ranjan et al., 2015). However, such distributional assumption is not always valid. A general n-order Markov model were also proposed but not popular in practise due to high computation. Beyond Markov models, Hidden Markov model based approaches are popular in both bioinformatics and general sequence problems (HHblits: Remmert et al., 2012; Helske and

Helske, 2016). It assumes a hidden layer of latent states which results into the observed sequence. These hidden states have a first-order Markov transition assumption, but due to the multi-layer setting the first-order assumption is not transmitted to the observed sequence. However, tuning HMM (finding optimal hidden states) is difficult and it is computationally intensive, thus, effecting its *generality* and *scalability*.

N-gram methods (also known as k-mer methods in bioinformatics realm) are the most popular approaches that work on the second assumption (Tomović et al., 2006; Hauser et al., 2013). Although the pretext of this assumption seems appropriate, the optimal selection of substring length, i.e. n in n-gram or k in k-mer, is difficult. In sequence mining, selection of a small value for n can lead to inclusion of noise but increasing it severely increases the computation. Some other variants, like spaced-words and adaptive n, is more difficult to optimize (Didier et al., 2012; Comin and Verzotto, 2012).

Another class of methods hypothesize that sequences are generated from some evolutionary process where a sequence is produced by reproducing complex strings from simpler substrings (Siyari et al., 2016, and references therein). This method solves a NP-hard optimization problem to identify the underlying evolution hierarchy and the corresponding substrings. These substrings can also be used as features for sequence data mining. However, the estimation algorithms for this, and similar, methods are heuristics that usually do not guarantee optimality. The algorithms can also lead to several solutions which will cause identifiability and ambiguity issues. Moreover, the evolutionary assumption may not be always true.

Besides these methods, sequence mining problem have also been given attention by the deep learning research community. Embedding spaces for sequences have been proposed using Recurrent Neural Networks (RNN) and Long Short Term Memory (Graves, 2013). However, the dimension of these embeddings are typically large, and is a rigorous tuning problem in a deep learning network. Training such model is computationally intensive, sometimes not interpretable and requires large amount of training data.

4.1.2 Problem Specification

As discussed above, the related methods fail to address at least one of the following challenges, a) Feature mapping: Effective extraction of sequence characteristics into a finite-dimensional euclidean space (a vector), b) Universal applicability: This mainly requires absence of any distributional or a domain specific assumption, and a small number of tuning hyperparameters, and c) Scalability: It relies on the computational complexity, which should be small with respect to sequence length, size of the database and alphabets set.

We propose a new sequence feature extraction function, called Sequence Graph Transform (SGT), that addresses all of the above challenges and is shown to outperform existing state-of-the-art methods in sequence data mining. SGT works by quantifying the pattern in a sequence by scanning the positions of all alphabets relative to each other. We call it a *graph* transform because of its inherent property of interpretation as a graph, where the alphabets form the nodes and a directed connection between two nodes shows their "association". These "associations" between all alphabets represent the characteristic features of a sequence. A Markov model transition matrix can be compared analogously with the SGT's feature space, however, among other differences (explored further in the paper), the associations (graph edges) do not represent a probability and SGT is non-parametric. The non-parametric property also makes it robust to any underlying sequence generation distribution.

In addition, sequence analysis problems can be broadly divided into: i) *length-sensitive*: the inherent patterns as well as the sequence lengths should match to render two sequences as similar, for eg., in protein sequence clustering, and ii) *length-insensitive*: the inherent patterns should be similar, irrespective of the lengths, for eg., weblog comparisons. In contrast with the existing literature, SGT provides a solution for both scenarios. Advantage of this property becomes more pronounced when we have to perform both types of analysis on the same data, and implementing different methods for each becomes cumbersome.

In this chapter, our major contributions are, a) development of a new feature extraction function, SGT, for sequences, b) a theoretical and experimental evaluation of SGT, and c) illustration through real data examples that SGT bridges the gap between sequence mining

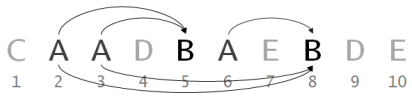


Figure 19: Illustration of effect of elements on each other. In this example, we show effect of presence of A on B.

and mainstream data mining through implementation of fundamental methods, viz. PCA, k-means, SVM and graph visualization on sequence data analysis via SGTs.

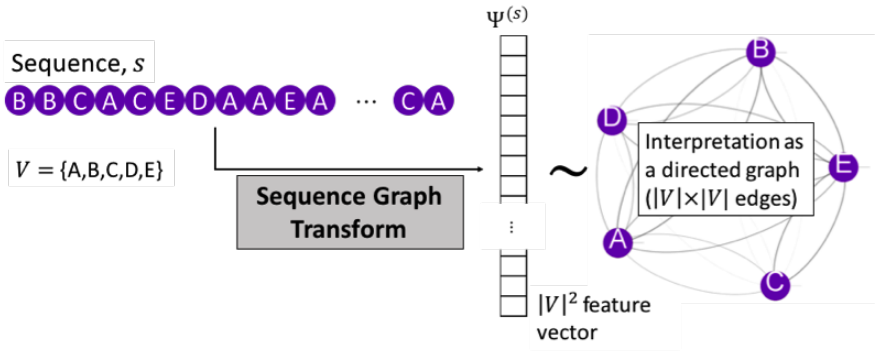
In the following, we first present the overview and intuition behind SGT, its formal definition and development, in Sec. 4.2. We also discuss its characteristics, theoretical properties, extensions, and implementation algorithms in this section. Thereafter, we experimentally validate the efficacy of SGT via clustering, and compare it with other state-of-the-art methods. We also show its capability of performing alphabet clustering as an extension. After the validation, we illustrate some real world applications, viz. clustering, visualization, classification and search, using four different datasets in Sec. 4.5. Finally, we discuss the results and performance aspects of SGT in Sec. 4.6.

4.2 Sequence Graph Transform (SGT)

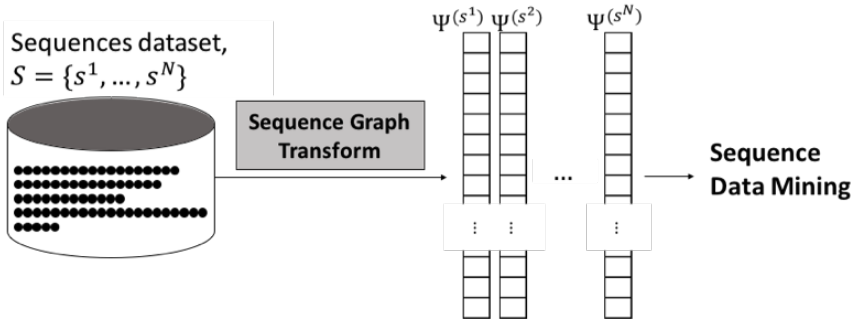
4.2.1 Overview and Intuition

Sequence Graph Transform works on the same fundamental premise — relative positions of alphabets in a sequence characterizes it — to extract the sequence characteristic pattern features. This premise holds true for any sequence problem, because similarity in sequences are measured based on the similarities in their pattern. For commonly occurring feed forward sequences, this premise is equivalent to: the relative position of an alphabet instance is a result of its interactions with all other alphabet instances prior to it. For a simpler terminology, we will call an alphabet instance as an event. A feed-forward sequence example can be a clickstream sequence for a user, where any subsequent click event depends on the prior links she clicked. In the following, we will illustrate and develop the feature extraction approach for feed-forward sequences, and later show its extension to "undirected" (no forward or reverse directional relationship between consecutive events) sequences.

In an illustrative example in Fig. 19, showing a feed-forward sequence, the presence of



(a) Feature extracted as a vector with a graph interpretation.



(b) Use of sequences' SGT features for data mining

Figure 20: Overview of SGT feature extraction and data mining procedure.

alphabet `B` at positions 5 and 8 should be seen in context with or as a result of all other predecessors. To extract the sequence features, we take relative positions of an alphabet pair at a time. For example, the relative positions for pair (`A`,`B`) are {(2,3),5} and {(2,3,6),8}, where the position set for `A` are the ones preceding `B`. In the SGT procedure defined and developed in the following sections (Sec. 4.2.3-4.2.4), we assay these position informations to extract the sequence features.

These extracted features are "association" between `A` and `B`, which can be interpreted as a connection feature representing "`A` leading to `B`". We should note that "`A` leading to `B`" will be different from "`B` leading to `A`". This is similar to the Markov probabilistic models, where transition probabilities of going from `A` to `B` is estimated. However, it is different because the connection feature , 1) is not a probability, and 2) takes into account all orders of relationship without any increase in computation.

The extracted association between `A` and `B` can also be interpreted as a measure of separation (or closeness) between `A` to `B`. Again, the separation going from `A` to `B` will be

different from B to A. The associations between all alphabets in alphabet set, denoted as $\mathcal{V}$, can be extracted similarly, to obtain sequence features in a $|\mathcal{V}|^2$-dimensional feature space. Besides, in contrast to the evolutionary or hidden layer models, SGT will not require search for any hidden states or strings.

The SGT features also make it easy to visualize the sequence as a directed graph, with sequence alphabets in $\mathcal{V}$ as graph nodes and the edge's weights equal to the directional association between nodes. Hence, we call it a sequence *graph* transform. Moreover, we show that under certain conditions the SGT also allows node clustering, thus, the alphabet clustering.

A high level overview of our approach is given in Fig. 20a-20b. In Fig. 20a we show that applying SGT on a sequence, s, yields a finite-dimensional feature vector, $\Psi^{(s)}$, for the sequence, also interpreted and visualized as a directed graph. For a general sequence data analysis, SGT can be applied on each sequence in a data corpus, as shown in Fig. 20b, to yield a finite- and equal-dimensional representation corresponding to each sequence. This provides a direct distance-based comparison between sequences, and thus, makes application of mainstream data mining methods for sequence analysis rather straightforward.

4.2.2 Notations

Suppose we have a dataset of sequences denoted by $\mathcal{S}$. Any sequence in the dataset, denoted by s, are made of alphabets in set $\mathcal{V}$. A sequence can have instances of one or many alphabets from $\mathcal{V}$. For example, sequences from a dataset, $\mathcal{S}$, made of alphabets in, $\mathcal{V} = \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}, \texttt{E}\}$(suppose), can be $\mathcal{S} = \{\texttt{AABAAABCC}, \texttt{DEEDE}, \texttt{ABBDECCABB}, \ldots\}$. As we can see, the sequences in the set have instances of $\{\texttt{A}, \texttt{B}, \texttt{C}\}$, $\{\texttt{D}, \texttt{E}\}$, $\{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}, \texttt{E}\} \subseteq \mathcal{V}$, respectively. The length of a sequence, s, denoted by $L^{(s)}$, is equal to the number of events in it. In the sequence, s_l will denote the alphabet at position l, where $l = 1, \ldots, L^{(s)}$ and $s_l \in \mathcal{V}$.

As mentioned in the previous section, we extract sequence features in the form of "associations" between the alphabets, represented as $\psi_{uv}^{(s)}$, where $(u, v) \in \mathcal{V}$, are the corresponding alphabets. Note that $\psi_{uv}^{(s)} \neq \psi_{vu}^{(s)}$ for feed-forward sequences. The connection features, $\Psi^{(s)} = [\psi_{uv}^{(s)}], (u, v) \in \mathcal{V}$, can be interpreted as a directed "graph", with edge weights, ψ, and

nodes in $\mathcal{V}$, or is vectorized to a $|\mathcal{V}|^2$-vector denoting the sequence s in the feature space. Each sequence, $s \in \mathcal{S}$, is transformed to this feature space to get $\Psi^{(s)}, \forall s \in \mathcal{S}$.

Finally, we use a developer function, $\phi_\kappa(d)$, where d is a "distance" variable, and κ is a tuning parameter to regulate ϕ. The developed function is used to obtain ψ, i.e. $\psi = f(\phi)$. The function f will be developed in the following subsection.

4.2.3 SGT Definition

As also explained in Sec. 4.2.1, the sequence graph transform extracts the features from the relative positions of events (alphabet instances). A quantification for an "effect" from relative positions of two events in a sequence is given by $\phi(d(l, m))$, where l, m are the positions of the events and $d(l, m)$ is a distance (or gap) measure. This quantification is a directional effect of the preceding event on the later event. For example, see Fig. 21a, where `A` and `B` are at positions l and m, and the directed arc denotes the effect of `A` on `B`.

For developing SGT, we require following conditions on ϕ, 1) Strictly greater than 0: $\phi_\kappa(d) > 0$; $\forall\, \kappa > 0$, $d > 0$; 2) Strictly decreasing with d: $\dfrac{\partial}{\partial d}\phi_\kappa(d) < 0$; and 3) Strictly decreasing with κ: $\dfrac{\partial}{\partial \kappa}\phi_\kappa(d) < 0$

The first condition is to keep the extracted feature, $\psi = f(\phi)$, easy to analyse and interpret. The second condition strengthens the effect of closer neighbors. The last condition helps in tuning the procedure, allowing us to change the effect of neighbors.

There are several functions that satisfies the above conditions, for eg., Gaussian, Inverse, and Exponential. Also, $d(l, m)$ can be chosen as absolute: $|m - l|$, quadratic: $(m - l)^2$, lagged gap: $(|m - l| - \text{constant})$, etc. In this paper, we choose $d(l, m) = |m - l|$ and ϕ as an exponential function as they will yield interpretable results for the SGT properties (Sec. 4.2.4). Thus,

$$\phi_\kappa(d(l, m)) = e^{-\kappa d(l,m)} = e^{-\kappa|m-l|},\ \forall\, \kappa > 0,\ d > 0 \tag{72}$$

In a general sequence, we will have several instances of an alphabet pair. For example, see Fig. 21b, where there are five $(\mathtt{A}, \mathtt{B})$ pairs, and an arc for each pair shows effect of `A` on

(a) (b)

Figure 21: Visual illustration of effect of alphabets' relative positions.

B. Therefore, the first step is to find the number of instances of each alphabet pair. The instances of alphabet pairs are stored in a $|\mathcal{V}| \times |\mathcal{V}|$ asymmetric matrix, Λ. Here, Λ_{uv} will have all instances of alphabet pairs (u, v), such that in each pair instance, v's position is after u.

Thus,

$$\Lambda_{uv}(s) = \{(l, m) : s_l = u, s_m = v, l < m, (l, m) \in 1, \ldots, L^{(s)}\} \tag{73}$$

In the sequence, ϕ from each (u, v) pair instance will contribute to the "association" feature, ψ_{uv}. Thus, we aggregate individual contributions from each pair instance and normalize it, as shown below in Eq. 74a-74b, to give ψ_{uv}. Here, $|\Lambda_{uv}|$ is the size of the set Λ_{uv}, which is equal to the number of (u, v) pair instances. Eq. 74a gives the feature expression for a *length-sensitive* sequence analysis problem because it also contains the sequence length information within it (proved with a closed-form expression under certain conditions in the following, Sec. 4.2.4). In Eq. 74b, the length effect is removed by standardizing $|\Lambda_{uv}|$ with the sequence length $L^{(s)}$ for *length-insensitive* problems.

$$\psi_{uv}(s) = \frac{\sum_{\forall (l,m) \in \Lambda_{uv}(s)} e^{-\kappa |m-l|}}{|\Lambda_{uv}(s)|}; \text{ length sensitive} \tag{74a}$$

$$\psi_{uv}(s) = \frac{\sum_{\forall (l,m) \in \Lambda_{uv}(s)} e^{-\kappa |m-l|}}{|\Lambda_{uv}(s)|/L}; \text{ length insensitive} \tag{74b}$$

and $\Psi(s) = [\psi_{uv}(s)]$, $(u, v) \in \mathcal{V}$ is the SGT feature representation of sequence s.

The developed SGT effectively extracts the pattern features of a sequence leading to an accurate comparison of different sequences. In the next section, we will prove this efficacy.

4.2.4 SGT properties

In this section, we derive closed-form expressions for the SGT feature, ψ_{uv}, under some mild assumptions. These closed-form expressions help in better understanding of the SGT's

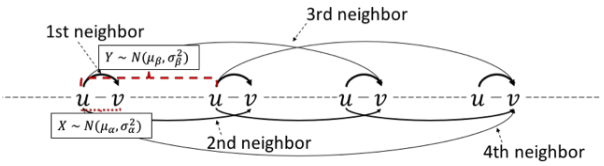


Figure 22: Illustration of notations used for SGT properties derivation

properties and proving its efficacy. However, in practice, SGT will work even without the assumptions made here.

Suppose we have a sequence, as shown in Fig. 22, in which the inherent pattern is "alphabet v occurs closely after u". Assuming the relative gap-distances between alphabet instances follow a normal distribution, the relative distances between u and its *first-neighboring* v is $X \sim N(\mu_\alpha, \sigma_\alpha^2)$, and between two consecutive u's is $Y \sim N(\mu_\beta, \sigma_\beta^2)$. Since v is supposed to occur closer to u, we will have $\mu_\alpha < \mu_\beta$. Besides, although the relative gap distances between alphabets are positive integers, it is safe to assume them to follow a continuous normal distribution.

As also mentioned before, and can be seen in Fig. 22, there will be several (u, v) pairs. To easily denote them, we use a term, m-th neighboring pair, a generalization of first-neighbor used above, where the m-th neighbor pair for (u, v) will have $m-1$ other u's in between. A first neighbor, is thus, the immediate (u, v) neighboring pairs, while the 2nd-neighbor has one other u in between, and so on.

We further assume that the marginal probability of the number of occurrences of each element in the sequence is uniform, i.e., $P_s[u] = P_s[v] = p; u, v \in \mathcal{V}$. Therefore, the expected number of first neighboring pairs (u, v), denoted by M, is

$$M = pL \tag{75}$$

where L is the sequence length. Consequently, it is easy to show that the expected number of m-th neighboring (u, v) pairs is $(M - m + 1)$, i.e., second neighboring (u, v) pairs will be $(M-1)$, $(M-2)$ for the third, so on and so forth, till one instance for the M^{th} neighbor (see Fig. 22 for an example). The gap distance for an m^{th} neighbor is given as,

$$Z_1 = X \quad ; \quad Z_m = X + \sum_{i=2}^{m} Y_i, m = 2, \ldots, M \tag{76}$$

Besides, the total number of (u, v) pair instances will be $\sum_{m=1}^{M} m = \frac{M(M+1)}{2} (= |\Lambda_{uv}|$, by definition). Suppose, we define a set that contains distances for each possible (u, v) pairs, as $\mathcal{D} = \{Z_m^i, i = 1, \ldots, m; m = 1, \ldots, M\}$. We can note that $|\mathcal{D}| = \frac{M(M+1)}{2}$.

Thus, putting these into Eq. 74a, the feature, ψ_{uv} can also be expressed as, $\psi_{uv} = \frac{\sum_{d_i \in \mathcal{D}} \phi_\kappa(d_i)}{|\Lambda_{uv}|}$.

Also, if $d \sim N(\mu_d, \sigma_d^2)$, the developer function $\phi_\kappa(d)$ becomes a lognormal distribution.

$$\phi_\kappa(d) \quad \sim \quad \text{lognormal}(\kappa\mu_\kappa, \kappa^2\sigma_\kappa^2) \tag{77}$$

On deriving the expected value (see Appendix B) for the length-sensitive feature, ψ_{uv}, we get,

$$E_p[\psi_{uv}] \quad \approx \quad \frac{2}{Lp+1}\gamma \tag{78}$$

where,

$$\gamma \quad = \quad \left| \frac{e^{-\tilde{\mu}_\alpha}}{1 - e^{-\tilde{\mu}_\beta}} \right| \tag{79}$$

and, $\tilde{\mu}_\alpha = \kappa\mu_\alpha - \frac{\kappa^2}{2}\sigma_\alpha^2; \tilde{\mu}_\beta = \kappa\mu_\beta - \frac{\kappa^2}{2}\sigma_\beta^2$.

As we can see in Eq. 78, the expected value has the length, L, of the sequence. Besides, within γ, both short and long pattern information of the sequence is embedded. The numerator of γ embeds the information on the short distance relative positions of the alphabets (u, v), while the denominator embeds the distant relative positions.

This property allows SGT to effectively capture the overall patterns from the relative positions of alphabets, and sets it apart from other methods. Besides, if the variances, $\sigma_\cdot^2$, are small, a higher value of κ will reduce the effect of distantly separated alphabets and vice-versa. Thus, it can easily regulate the effect of preceding events without any increase in computation.

Moreover, the variance of ψ_{uv} in this case approaches to 0 with L (see Appendix B), making the SGT features more precise.

Next, for the length-insensitive SGT, the expected value for ψ_{uv} is,

$$E_p[\psi_{uv}] \approx \frac{2L}{Lp+1}\gamma \tag{80}$$

Therefore, as sequence length, L, increases, the (u, v) "association" feature approaches a constant,

$$\lim_{L\to\infty} E_p[\psi_{uv}] = \frac{2}{p}\left|\frac{e^{-\tilde{\mu}_\alpha}}{1-e^{-\tilde{\mu}_\beta}}\right| = \frac{2}{p}\gamma \tag{81}$$

Again, the variance of ψ_{uv} in length-insensitive case also goes to 0 with L (see Appendix B). Thus,

$$\lim_{L\to\infty} \Pr\{\psi_{uv} = \frac{2}{p}\gamma\} \to 1 \tag{82}$$

Therefore, the effect of sequence length is removed in this SGT variant. This makes the resulting features invariant to the sequence length, and thus, enables a length insensitive sequence analysis.

4.2.5 Extensions of SGT

4.2.5.1 Undirected sequences

SGT can be further extended to work on undirected sequences. In such sequences, the directional pattern or directional relationships (like in feed-forward) is not important. In other words, it is immaterial whether `B` occurs before or after `A`, occurring closely (or farther) is important. Here we are interested in overall proximity of events in either direction to characterize a sequence's pattern. From SGT operation standpoint, we have to remove the condition, $l < m$, from Eq. 73, giving us,

$$\tilde{\Lambda}_{uv}(s) = \{(l, m) : s_l = u, s_m = v, (l, m) \in 1, \ldots, L^{(s)}\} \tag{83}$$

Thus, the SGT for undirected sequences can be computed, and denoted as $\tilde{\Psi}$.

It is easy to show that.

$$\tilde{\Lambda} = \Lambda + \Lambda^T \tag{84}$$

and (see Appendix C for proof),

$$\tilde{\Psi} = \frac{|\Lambda|\Psi + |\Lambda^T|\Psi^T}{|\Lambda| + |\Lambda^T|} \tag{85}$$

where, Λ and Ψ are given in Eq. 73 and Eq. 74a-74b, respectively.

Moreover, for sequences with uniform marginal distribution of occurrence of elements, $v \in \mathcal{V}$, Λ will be close to symmetric, thus, the undirected sequence graph can be approximated as,

$$\tilde{\Psi} \approx \frac{\Psi + \Psi^T}{2} \tag{86}$$

In practice, this approximation is useful in most cases.

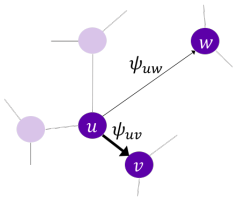
4.2.5.2 Alphabet clustering

Node clustering in graphs is a classical problem, with various techniques, like, spectral clustering, graph partitioning, and others. SGT's graph interpretation facilitates grouping of alphabets that occur closely via any of these node clustering methods.

This will effectively require the SGT to give larger weights to the edges, ψ_{uv}, corresponding to alphabet pairs that occur closely. For instance, consider a sequence in Fig. 23a, in which v occurs closer to u than w, also implying $E[X] < E[Y]$. Therefore, in this sequence's SGT representation, edge weight for $u \rightarrow v$ should be greater than for $u \rightarrow w$, i.e. $\psi_{uv} > \psi_{uw}$. Note that the feature ψ_{uv} is same as the edge weight for arc $u \rightarrow v$ in the graph interpretation.

Using same assumption of uniform marginal probability of alphabet occurrences (as in, Sec. 4.2.4), we will have, $E[|\Lambda_{uv}|] = E[|\Lambda_{uw}|]$. Therefore, $\psi_{uv} \propto E[\phi(X)]$ and $\psi_{uw} \propto E[\phi(Y)]$, and due to Condition 2 on ϕ given in Sec. 4.2.3,

(a) u, v are closer than u, w.



(b) Corresponding SGT's Graph view.

Figure 23: Illustrative sequence example for alphabet clustering.

$$
\begin{aligned}
if \quad & E[X] < E[Y] \\
\implies \quad & E[\phi(X)] > E[\phi(Y)] \\
\implies \quad & E[\psi_{uv}] > E[\psi_{uw}]
\end{aligned}
$$

Moreover, for an easier clustering, it is important to bring the "closer" alphabets more close in the euclidean space and vice-versa. In the SGT's graph interpretation for the above example, it implies ψ_{uv} should go as high as possible to bring v closer to u in the graph and vice-versa for (u, w). Thus, effectively, $\Delta = E[\psi_{uv} - \psi_{uw}]$ should be increased.

As proved in Appendix D, Δ will increase with the hyperparameter κ, if we select κ such that $\kappa d > 1$ holds for any value of d. Since, d in this case is the relative gap-distance between sequence events, it is always a positive integer. Therefore, the criteria on κ should hold for any $d \in \mathbb{N}$.

Thus, a SGT can represent a sequence as a graph with its alphabets connected with weighted edges, which enables clustering of closely occurring alphabets using graph node clustering methods.

4.3 SGT Algorithm

In this section, we provide two algorithms for SGT computation. Both algorithms are designed for a feed-forward sequence defined in Eq. 74a-74b and developed with an exponential function for ϕ given in Eq. 72. The two algorithms are for the following two cases to achieve faster computation: 1) Algorithm-1 when the sequence lengths are smaller than the feature

Algorithm 1 Parsing a sequence to extract the SGT features

Input: A sequence, $s \in \mathcal{S}$, alphabet set, $\mathcal{V}$, and hyperparameter, κ.

Initialize:
    $\mathbf{W}^{(0)}, \mathbf{W}^{(\kappa)} \leftarrow \mathbf{0}_{\mathcal{V}\times\mathcal{V}}$, and sequence length, $L \leftarrow 1$
for $i \in \{1, \ldots, (\text{length}(s) - 1)\}$ **do**
    for $j \in \{(i+1), \ldots, \text{length}(s)\}$ **do**
        $\mathbf{W}^{(0)}_{s_i,s_j} \leftarrow \mathbf{W}^{(0)}_{s_i,s_j} + 1$
        $\mathbf{W}^{(\kappa)}_{s_i,s_j} \leftarrow \mathbf{W}^{(\kappa)}_{s_i,s_j} + \exp(-\kappa|j-i|)$
        where, $s_i, s_j \in \mathcal{V}$
    end for
    $L \leftarrow L + 1$
end for
if length-sensitive is True **then**
    $\mathbf{W}^{(0)} \leftarrow \mathbf{W}^{(0)}/L$
end if

Output $\psi_{uv}(s) \leftarrow \left(\frac{W^{(\kappa)}_{u,v}}{W^{(0)}_{u,v}}\right)^{\frac{1}{\kappa}}$;$\Psi(s) = [\psi_{uv}(s)]$, $(u, v) \in \mathcal{V}$

space, i.e. $L < |\mathcal{V}|^2$, and 2) Algorithm-2 when $L > |\mathcal{V}|^2$. However, the second algorithm will require an additional pre-processing of a sequence.

The algorithms take in a sequence, s, from the corpus of sequence database, $\mathcal{S}$, alphabet set, $\mathcal{V}$, and the SGT hyperparameter, κ. We initialize two $\mathcal{V} \times \mathcal{V}$ matrices, $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(\kappa)}$, with zero values, where a $\mathcal{V} \times \mathcal{V}$ matrix is a square matrix of dimension $|\mathcal{V}|$ and the row and column index names are same as the set $\mathcal{V}$. Besides, the sequence length, L, is initialized as 1 in Algorithm-1, while 0 for Algorithm-2. Post computation, $\mathbf{W}^{(0)}$ will correspond to the denominator and $\mathbf{W}^{(\kappa)}$ to the numerator in Eq. 74a-74b.

Algorithm-1 parses the input sequence across its length using a nested loop and updates $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(\kappa)}$. Thereafter, if the problem is *length insensitive,* each cell in $\mathbf{W}^{(0)}$ is normalized by dividing with L. Finally, an element-wise division of $\mathbf{W}^{(\kappa)}$ by $\mathbf{W}^{(0)}$ gives the SGT defined in Eq. 74a-74b. However, we output the κ^{th} root of it as the final SGT features, because although the κ^{th} root is not necessary theoretically, it keeps the SGTs easy-to-interpret and comparable for any value of κ.

In Algorithm-2, instead of parsing the sequence length, we perform a nested loop on all alphabets in $\mathcal{V}$ to update $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(\kappa)}$. For that, we pre-process the sequence to obtain

a list containing the positions for each alphabet, by using the defined function GetAlphabetPositions. Thereafter, for any pair of alphabets (u, v) we take the set of their positions in the sequence, U and V. We, then, assign a set C as the cartesian product of sets U and V with a condition that in any resulting pair the position value from set V should be higher than the one from set U. Thus, the cell (u, v) for $\mathbf{W}^{(0)}$ will be equal to the number of tuples in set C, and for $\mathbf{W}^{(\kappa)}$ will be the sum over all results of function ϕ_κ on the difference between elements of each tuple in C. The remaining steps are same as Algorithm-1.

For the SGT extension to undirected sequences in Sec. 4.2.5.1, the loop range in line 3 in Algorithm-1 should be changed to $j \in \{1, \ldots, \text{length}(s)\}$ and the set assignment in line 13 in Algorithm-2 should change to $C \leftarrow U \times V = \{(i, j) | i \in U, j \in V\}$.

The outputted SGT for the sequence, s, will be a $|\mathcal{V}| \times |\mathcal{V}|$ matrix, which can be vectorized (size, $|\mathcal{V}|^2$) for use in distance-based data mining methods, or can be used as is for visualization and interpretation purposes. Moreover, for a sequence dataset, $\mathcal{S}$, the algorithm should be repeated for all sequences to obtain their feature representations.

The time complexity of Algorithm-1 is $O(NL^2)$, where N is the number of sequences $(= |\mathcal{S}|)$ and L is the average sequence length, and $O(N|\mathcal{V}|^2)$ for Algorithm-2, if the preprocessing step is excluded. The space complexity for both is $O(N|\mathcal{V}|^2)$. However, in most datasets, not all alphabets in $\mathcal{V}$ are present in a sequence, resulting into a sparse SGT features representation. In such cases, the complexities reduce by factor of the sparsity level. Moreover, as also evident from Fig. 20b, the SGT operation on any sequence in a dataset is independent of other. This means, we can easily parallelize the SGT operation on the sequences in $\mathcal{S}$ to significantly reduce the runtime.

The optimal selection of the hyperparameter κ will depend on the problem in hand. If the end objective is building a supervised learning model, methods like cross-validation can be used. For unsupervised learning, any goodness-of-fit criteria can be used for the selection. In cases of multiple parameter optimization, for eg. the number of clusters (say, n_c) and κ together in an unsupervised learning, we can use random search procedure. In such a procedure, we randomly initialize n_c, compute the best κ based on some goodness-of-fit measure, then fix κ to find the best n_c, and repeat until there is no change. From our

Algorithm 2 Extract SGT features by scanning alphabet positions of a sequence

Input: A sequence, $s \in \mathcal{S}$, alphabet set, $\mathcal{V}$, and hyperparameter, κ.

function GetAlphabetPositions($s, \mathcal{V}$)
    positions $\leftarrow \{\emptyset\}$
    for $v \in \mathcal{V}$ **do**
        positions(v) $\leftarrow \{i : s_i = v, i = 1, \ldots, \text{length}(s)\}$
    end for
    return positions
end function

Initialize:
    $\mathbf{W}^{(0)}, \mathbf{W}^{(\kappa)} \leftarrow \mathbf{0}_{\mathcal{V} \times \mathcal{V}}$, and sequence length, $L \leftarrow 0$
    positions $\leftarrow$ GetAlphabetPositions($s, \mathcal{V}$)
for $u \in \mathcal{V}$ **do**
    $U \leftarrow$ positions(u)
    for $v \in \mathcal{V}$ **do**
        $V \leftarrow$ positions(v)
        $C \leftarrow (U \times V)^{+} = \{(i, j) | i \in U, j \in V, \& \, j > i\}$
        $\mathbf{W}^{(0)}_{u,v} \leftarrow \text{length}(C)$
        $\mathbf{W}^{(\kappa)}_{u,v} \leftarrow \text{sum}(\exp(-\kappa | C_{:,u} - C_{:,v}|))$
    end for
    $L \leftarrow L + \text{length}(U)$
end for
if length-sensitive is True **then**
    $\mathbf{W}^{(0)} \leftarrow \mathbf{W}^{(0)} / L$
end if

Output SGT: $\psi_{uv}(s) \leftarrow \left(\frac{W^{(\kappa)}_{u,v}}{W^{(0)}_{u,v}} \right)^{\frac{1}{\kappa}}$; $\Psi(s) = [\psi_{uv}(s)]$, $(u, v) \in \mathcal{V}$

experiments on real and synthesized data, the results of SGT based data mining are not sensitive to minor differences in κ. In our implementations, we typically selected κ from $\{1, 5, 10\}$.

4.4 Experimental Analysis

In this section, we perform an experimental analysis to assess the performance of the proposed SGT. The most important motivation behind SGT is the need for an accurate method to find (dis)similarity between sequences. Therefore, to test SGT's efficacy in finding sequence (dis)similarities, we built sequence clustering experimental setup. Clustering operation requires accurate computation of (dis)similarity between objects, thus, is a good choice for efficacy assessment.

Table 5: Experimentation settings

	Sequence length, μ, σ	**Noise level**	**#clusters,** n_c
Exp-1	424.6, 130.6	45-50%	5
Exp-2	116.4, 47.7	35-65%	5
Exp-3	98.2, 108.3	–	5
Exp-4	103.9, 33.6	30-50%	3

We perform four types of experiment, a) Exp-1: length-sensitive, b) Exp-2: length-insensitive with non-parametric sequence pattern, c) Exp-3: length-insensitive with parametric sequence pattern, and d) Exp-4: alphabet clustering. The settings for each of them are given in Table 5. Alphabet set is, $\mathcal{V} = \{\mathtt{A}, \mathtt{B} \ldots, \mathtt{Z}\}$, for all sequences. Besides, except for Exp-3, clustered sequences were generated, such that sequences within a cluster share common patterns. Here two sequences having a common pattern primarily means the sequences have some common subsequences of any length, and these subsequences can be present anywhere in the sequence. The sequences also comprise of other events, which can be either *noise* or some other pattern. This setting is non-parametric, however, the subsequences can also bring some inherent parametric properties, like a mixture of Markov distribution of different orders. In Exp-3, clustered sequences were generated from a mixture of parametric distributions, like Markov and Hidden Markov models. In all the experiments, k-means with manhattan distance was applied on SGT representations of the sequences.

In Exp-1, we compare SGT with length-sensitive algorithms, viz. MUSCLE, UCLUST and CD-HIT, which are popular in bioinformatics. These methods are hierarchical in nature, and thus, itself finds the optimal number of clusters. For SGT-clustering, the number of clusters are found using the random search procedure recommended in Sec. 4.3. Besides, other methods, like Optimal Matching (for eg. Needleman-Wunsch) or sequence alignment (for eg. Smith-Waterman), are not considered due to their high time complexity, which makes them inappropriate for most large sequence datasets.

Fig. 24 shows the results, where the y-axis is the ratio of the estimated optimal number of clusters, $\hat{n}_c$, and the true number of clusters, n_c. On the x-axis, it shows the clustering accuracy, i.e. the proportion of sequences assigned to a same cluster given that they were

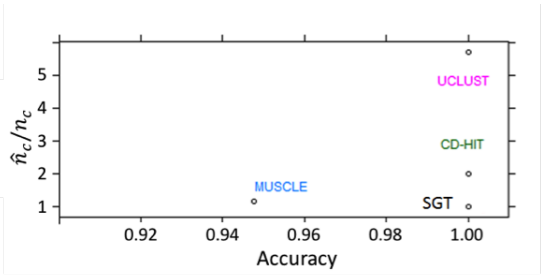


Figure 24: Exp-1 on length-sensitive sequence problem.

actually from the same cluster. For a best performing algorithm, both metrics should be close to 1. As shown in the figure, CD-HIT overestimated the number of clusters by about twice, while UCLUST severely overestimated by 5 times, but both had a 100% accuracy. MUSCLE accurately estimated n_c but had about 95% accuracy. On the other hand, SGT could accurately estimate n_c, as well as, it has a 100% accuracy.

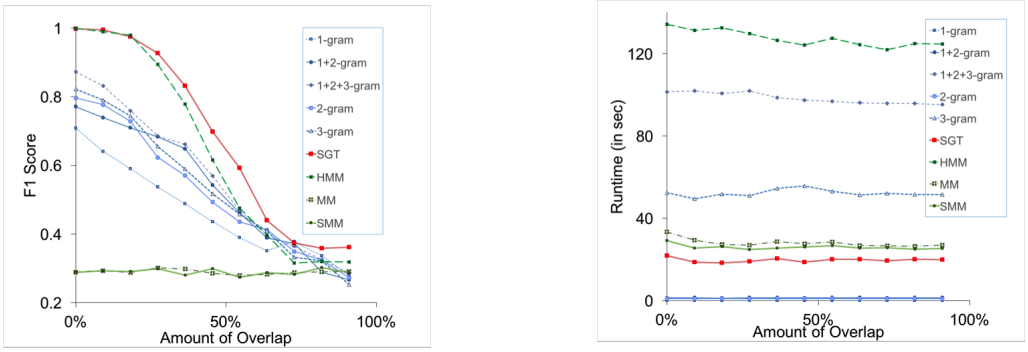
In Exp-2, we compare SGT with commonly used sequence or string analysis techniques, viz. n-gram, mixture Hidden Markov model (HMM), Markov model (MM) and semi-Markov model (SMM) based clustering. For n-gram, we take $n = \{1, 2, 3\}$, and their combinations. Note that 1-gram is equivalent to bag-of-words method. For these methods, we provided the known n_c to the algorithms. We use F1-score as the accuracy metric. It considers both the *precision*[1] and the *recall*[2] of the test to compute the score by taking a weighted average of both, where its best value is 1 and worst at 0.

Besides, in this experiment, we set different scenarios such that the overlap of clusters' "centroid" are increased. A high overlap between clusters implies the sequences, belonging to these clusters, have higher amount of common patterns. Thus, separating them for clustering becomes difficult, and clustering accuracy is expected to be lower.

Fig. 25a shows the accuracy results for the above experimentation. As shown in the figure, SGT-clustering always has a higher accuracy (F1-score) and is significantly higher than MM and SMM. This is due to the fact that both MM and SMM work on a first-order Markov distribution assumption on sequences. On the other hand, although HMM has a

[1]Precision is the number of correct positive results divided by the number of all positive results

[2]Recall is the number of correct positive results divided by the number of positive results that should have been returned



(a) (b)

Figure 25: Experimentation results for general sequence datasets to compare the efficiency of Sequence clustering methods. Higher overlap implies the sequences, belonging to different clusters, share many patterns and, therefore, are harder to separate.

first-order Markov assumption on the hidden states, the observed sequence events do not require such condition. Therefore, in comparison to the MM and SMM, HMM's accuracy is relatively more robust to the strict distribution assumptions on a sequence. As a result, we see its F1-score to be comparable to SGT. The n-gram methods lie in between. An interesting finding is, while higher order n-grams performed better in general, the 1-gram method is better when overlapping was high. This shows higher order n-grams inability to distinguish between sequences when patterns are very similar.

Besides, in Fig. 25b the runtimes of the methods are compared. The smaller order n-grams have very low runtime. Among others, HMM has significantly higher runtime while SGT has lowest.

Furthermore, we did Exp-3 to see the performance of SGT in sequence datasets having an underlying mixture of parametric distributions, viz. mixture of HMM, MM and SMM. The objective of this experimentation is to test SGT's efficacy on parametric datasets against parametric methods. In addition to obtaining datasets from mixed HMM, and first-order mixed MM and SMM distributions, we also get second-order Markov (MM2) and third-order Markov (MM3) datasets. As expected, the mixture clustering method corresponding to the underlying distribution is performing the best. Note that SMM is slightly better than MM in the MM setting because of its over-representative formulation, i.e. a higher dimensional model to include a variable time distribution. However, the proposed SGT's accuracy is

(a) (b)

Figure 26: Efficacy validation of proposed SGT method on experimental datasets synthesized from different parametric distributions. For each parametric dataset, all parametric and proposed SGT methods are compared.

always close to the best. This shows SGT's robustness to underlying distribution and its universal applicability. Besides, again its runtime is smaller than all others.

Finally, we validate the efficacy of the extensions of SGT given in Sec. 4.2.5 in Exp-4. Our main aim in this validation is to perform alphabet clustering (Sec. 4.2.5.2). We setup a test experiment such that across different sequence clusters some alphabets occur closer to each other. We create a dataset which has sequences from three clusters and alphabets belonging to two clusters (alphabets, A-H in one cluster and I-P in another). There are common patterns between sequences in a cluster and the patterns are such that alphabets from same alphabet cluster will be closer. For eg. a cluster can have patterns like $\{\texttt{EFEACDAA}, \texttt{FGGCA}, \texttt{NNKLI}, \texttt{KJJOOLLPM}, \ldots\}$, where a pattern maintains the closeness of alphabets imposed from their underlying true group.

This emulates a biclustering scenario where sequences in different clusters will have distinct patterns, however, the pattern of closely occurring alphabets is common across all sequences. This is a complex scenario where clustering both sequences and alphabets can be challenging. In a typical real world problem, the alphabet clusters can be expected to be distinct in different sequence clusters, where the bi-clustering can be done in two stages, 1) cluster sequences, 2) for each sequence cluster, cluster the alphabets. Nevertheless, in this experiment we show that the developed SGT can be used to perform the bi-clustering together on a complex dataset.

In this experiment, we assume that it is required to group the alphabets by, a) finding

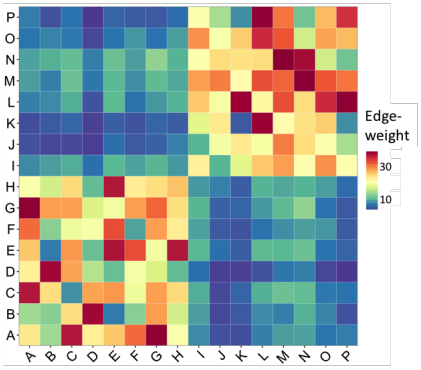


Figure 27: Heat-map showing alphabets' association via SGT edge-weights.

closeness between *distinct* alphabets, and b) irrespective of their directional pattern. For the former, we include SGT extension in Sec. 4.2.5.2 and the extension in Sec. 4.2.5.1 for the latter. This setup will validate the SGT extensions together. However, the accuracy will remain unaffected even if the analysis is done without the above two assumptions and, in turn, extensions.

Upon clustering the sequences, the F1-score is found as 1.0. . For alphabets clustering, we applied spectral clustering on the aggregated SGT's of all sequences, which yielded a accurate result with only one alphabet as mis-clustered. Moreover, a heat-map in Fig. 27 clearly shows that alphabets within same underlying clusters have significantly higher associations. Thus, it validates that SGT can accurately cluster alphabets together with clustering the sequences.

In this section, we showed that SGT outperforms existing methods in terms of both accuracy and runtime. This validates the premise that SGT is an effective representation of sequences, that precisely characterizes a sequence's patterns, and thus, provides accurate distance-based (dis)similarities for sequence data mining.

4.5 Applications

In this section we show few applications of SGT based sequence mining on real world datasets. We use four datasets, two from protein databases, one computer network log data and a user web navigation data. We show that SGT facilitates use of mainstream data mining for sequences and, indeed, outperforms the commonly used sequence mining

methods.

4.5.1 Clustering

Sequence clustering is an important application area across various fields. One important problem in this area is clustering user activity on web (web log sequences) to understand their behavior. This analysis helps in better service, design, advertisements and promotions.

We take a user navigation data[3] on msnbc.com collected during a 24-hour period. The navigation data are weblogs that correspond to page views of each user. The alphabets of these sequences are the events corresponding to a user's page request. These requests are recorded at a higher abstract level of page category, which are representative of the structure of the website. The categories are: `frontpage, news, tech, local, opinion, on-air, misc, weather, health, living, business, sports, summary, BBS (bulletin board service), travel, MSN-news,` and `MSN-sports`. The dataset comprises of 989,818 weblogs (sequences), of which we use a random sample of 100,000 sequences for our analysis. As expected, the distribution of sequence lengths (shown in Fig. 28, the distribution of log of sequence lengths) is found to be skewed and multi-modal. Their average length is 6.9 and standard deviation is 27.3, with range between 2 and 7440.

Our objective is to group the users with similar navigation patterns, irrespective of differences in their session lengths, into clusters. We, therefore, take the *length-insensitive* SGT version and use the random search procedure suggested in Sec. 4.3 to determine the optimal number of user clusters. In the procedure, we used k-means clustering with manhattan distance, and the goodness-of-fit criterion as db-index (Davies and Bouldin, 1979). The optimal point is found for the tuning parameter, $\kappa = 9$, and the number of clusters, $n_c = 104$.

The frequency distribution of number of members in each cluster is shown in Fig. 29. Again, as expected, the frequency distribution is like Pareto — majority of users belong to a small set of clusters. Nevertheless, it is important to understand distinct behaviors of both majority and minority users for better personalized services. Although most clusters have

[3] archive.ics.uci.edu

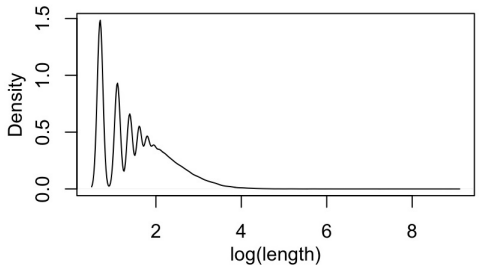


Figure 28: Sequences log-length distribution on msnbc.com

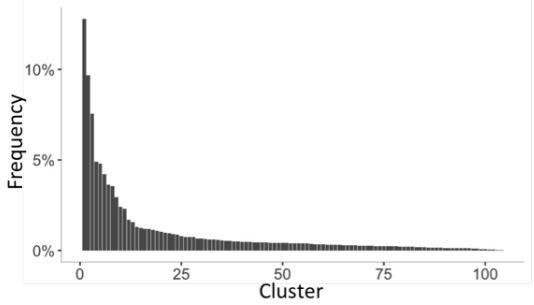


Figure 29: Frequency distribution of number of members in msnbc.com user clusters.

small memberships, but with huge amount of user sessions everyday, optimizing advertisements, marketing or design strategies, personalized to each group of users types (by their behavior) can bring significant improvement in returns.

In Cadez et al. (2000), the optimal number of clusters for a sample of same size was found to be 100, close to our finding. However, the overall clustering results from Cadez et al. may have inconsistencies due to a first-order Markovian assumption. We ran hypothesis tests on the sequence data for verifying the assumption. The Markov tests result show majority of the data, about 67.9%, could not be tested, either due to the sequence's short length or presence of single event throughout the sequence. Of the remaining, about 27% follows first order Markov property, and the rest has second or higher orders. Thus, applying Markov model based clustering on such data can be ineffective, on the other hand, proposed SGT based clustering does not depend on any distributional assumptions, hence, is better capable in identifying similar sequences.

4.5.2 Visualization

Effective visualization is a critical need for easy interpretation of a data and underlying properties. For example, in the above msnbc.com navigation data analysis, interpreting behavior of different user clusters is as important. The SGT's graph equivalence provides an appropriate visualization technique.

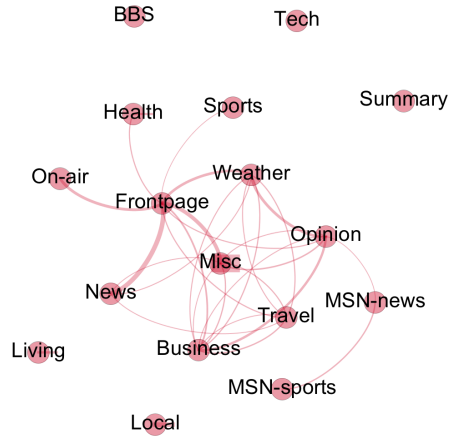
In the following (Fig. 30a-30f), we show graph visualization of some clusters' centroids, because a centroid represents the behavior of users present in the cluster. We have filtered edges with small weights for a better visualization. Thus, weakly connected nodes, i.e. infrequently visited pages, are seen as isolated.

The representative SGT for the first cluster is shown in Fig. 30a. This cluster contains the highest membership (∼12%), thus, indicates the "most" general behavior. The behavior graph in this group is centered around `frontage` and `misc`, with users tendency to navigate between `frontpage, misc, weather, opinion, news, travel` and `business`. Besides, users also tend to go to `on-air, health, sports` and `msn-news/sports` before leaving the website.

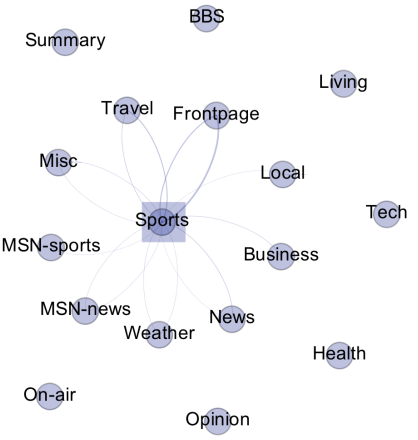
Fig. 30b shows another majority cluster with about 7.5% membership. This group of users seem to have a liking for sports. They primarily visit `sports` related pages (the box around `sports` node indicate a self-visiting edge), and also move back-and-forth from `sports` to `frontpage, travel, misc, msn-sports/news, weather, news, business,` and `local`.

Further, in Fig. 30c and 30d, we show two clusters with memberships of about 1%. The user behavior in both clusters are slightly similar yet significantly different. In cluster-25, the users primary interest is centered around `living` and `business`, while in cluster-31, it is centered around `living` and `travel`. Thus, it can be interpreted that users having the former behavior are involved in more business and visit `living` pages for general daily and local living information, while the latter behavior indicates the users are interested in travel and living information of different places (not local).

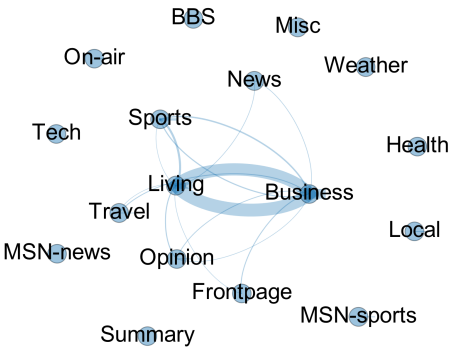
The clusters shown in Fig. 30e and 30f have memberships smaller than 1%. The users navigation behaviors in these clusters are centered around `frontpage`. The users in cluster-73 navigate frequently between the `frontpage, msn-sports` and `living`; while the users



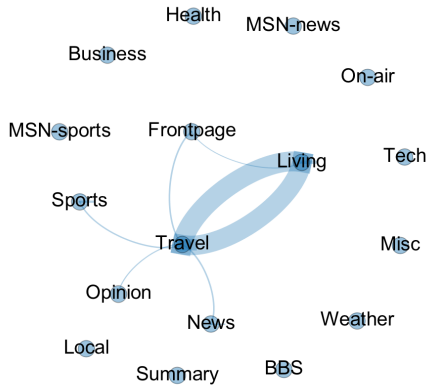
(a) Cluster# 1



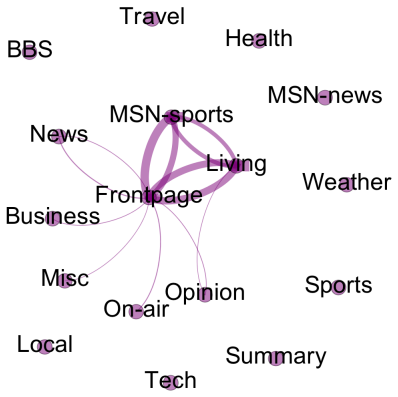
(b) Cluster# 3



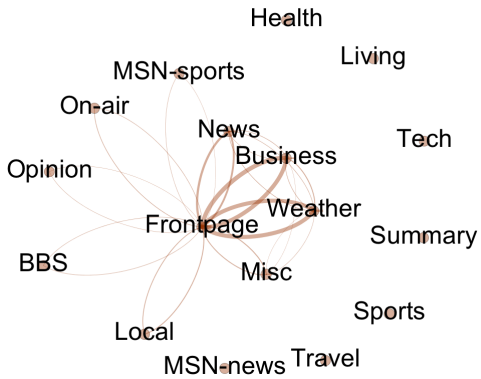
(c) Cluster# 25



(d) Cluster# 31



(e) Cluster# 73



(f) Cluster# 85

Figure 30: Graphical visualization of cluster representatives, showing general behavior of the cluster's members.

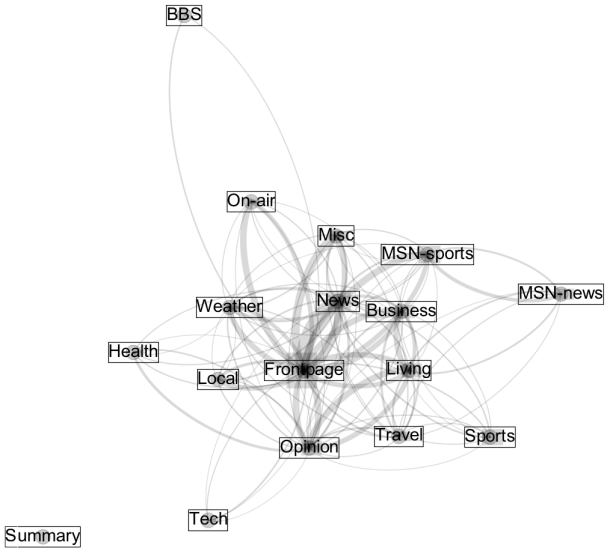


Figure 31: General navigation behavior of users on msnbc.com.

in cluster-85 go between `frontpage, news, business` and `weather`. This indicates these users are either casual browsers or new users of msnbc.com, who navigate around the frontpage more often than surfing any specific category.

Thus, while the clustering approach provided us with cluster of users, the visualization aided in understanding the user behavior in a group. Besides, we did another analysis on the data, by considering the entire sample as one cluster (unit). This analysis will give us a general picture of user navigation behavior (see Fig. 31). It is observed that `summary` is the least visited category followed by `BBS` and `tech`. The `tech` category is mostly visited from `frontpage` and `news`, and counter-intuitively significantly often from `opinion` than `business`.

4.5.3 Classification

At many occasions we have labeled sequence data where it is required to build a classification model that can accurately assign a new sequence to the right class. Here we show that SGT representation can be used for building sequence classifiers. We use two datasets, a) protein sequences and its function as labels, and b) network intrusion data containing audit logs and attack labels.

The protein dataset is a sample of 2113 sequences taken from www.uniprot.org, where a sequence is from 20 alphabets (amino acids). Each sequence has one of two functions, viz. "Might take part in the signal recognition particle (SRP) pathway" and "Binds to DNA and alters its conformation". These functions are treated as the sequence labels, and have almost a balanced distribution (∼46.4% for the first function). Besides, the sequence lengths in this dataset are similar, in the range of (289, 300). For this problem we use the *length-insensitive* SGT.

Another real world problem considered here is detection of intrusion in a computer network from the audit data. We use a sample of audit data with labels for an attack (positive class) or a normal event from MIT Lincoln Laboratory[4]. The dataset contains a BSM log file with about 400,000 lines, after post-processing gives event sequences for 115 sessions. There are 49 event types, corresponding to the alphabets set. The session lengths vary over a wide range of (12, 1773) with mean and standard deviation of (179, 192). Besides, since network intrusions are a rare event, the class distribution is significantly unbalanced with just 11.3% positive class datapoints.

In this problem, we consider the sequence lengths as important (a *length-sensitive* problem), because sequences with similar pattern but different lengths can have different labels. Take a simple example of following two sessions: {`login, password, login, password, mail,...`} and {`login, password,...(repeated several times)..., login, password`}, while the first session can be a regular user mistyping the password once, the other session is possibly an attack to guess the password. Thus, the sequence lengths are as important as the patterns.

For both datasets, we first transformed the sequences to vectors using SGT, length-insensitive and -sensitive for protein ($\kappa = 1$) and network intrusion ($\kappa = 10$) data, respectively. For the network intrusion data, the sparsity of SGTs were high. Therefore, we performed principal component analysis (PCA) on it, and kept the top 10 PCs as sequence features, we call it SGT-PC, for further modeling.

After obtaining the SGT (-PC) features, we trained a SVM classifier on them. For

[4]https://www.ll.mit.edu/ideval/data/1998data.html

Table 6: Classification accuracy (F1-score) based on 10-fold cross validation results. # Features is dimension of input data to SVM, and SVs is the number of support vectors selected.

	F1-score (#Features/SVs)	
SVM on- $\{\gamma_{protein}; \gamma_{network}\}$, $c = 1$	**Protein data**	**Network intrusion**
SGT $\{0.0014; 0.1\}$	**99.61%** (400/120)	**89.65%**(2,401/28)
Bag-of-words $\{0.05; 0.02\}$	88.45% (20/35)	48.32% (49/48)
2-gram $\{0.0025; 0.00041\}$	93.87% (400/45)	63.12% (2,401/25)
3-gram $\{0.00012; 8.4e-06\}$	95.12% (8,000/202)	49.09% (117,649/29)
1+2-gram $\{0.0012; 0.0004\}$	94.34% (820/28)	64.39% (2,450/26)
1+2+3-gram $\{4.02e-05; 8.32e-06\}$	96.89% (24,820/15)	49.74% (120,099/27)

comparison with commonly used sequence classifier techniques, we implemented bag-of-words (1-gram), 2-, 3-, 1+2-, 1+2+3-gram methods. The SVM was built with a RBF kernel. The cost parameter, c, is equal to 1, while the value for kernel parameter, γ, is shown within braces in Table 6. F1-score is used as a measure of accuracy, especially due to the network intrusion dataset where both precision and recall are important. The average test accuracy (F1-score) from a 10-fold cross validation is reported in the Table 6. Besides, the dimension of the data fed into SVM and the number of support vectors selected during training are shown as, "#Features" and "SVs", respectively.

As we can see in Table 6, the F1-scores are high for all methods in protein data, with SGT based SVM surpassing all others, followed by 1+2+3-gram. On the other hand, the accuracies are small for the network intrusion data. This is primarily because of, a) a small dataset but high dimension (related to the alphabets size), leading to a weak predictive ability of models, and b) a few positive class examples (unbalanced data) causing a poor recall rate. Still, SGT outperformed other methods with a significant margin. Although the accuracies of the methods can be further increased using other classifiers, like Boosting, Random Forest, etc., it is beyond the scope of this paper. Here our purpose is to make a simplistic comparison to highlight the superiority of SGT features in building a supervised learning model.

Table 7: Protein search query result from a sample dataset of size 1000.

Query, Q9ZIM1

Protein	**SGT-PC**	**Identity**
S9A4Q5	33.02	46.3%
S8TYW5	34.78	46.3%
A0A029UVD9	39.21	45.1%
A0A030I738	39.34	45.1%
A0A029UWE3	39.41	45.1%
A0A029TT10	39.90	45.1%
G6N1A7	44.29	45.8%
J1GZY3	44.49	45.8%
F9M004	44.58	45.8%
M2CL02	44.61	44.3%

4.5.4 Search

Most sequence databases found in real world are very large. For example, protein databases have billions of sequences and increasing. This increasing size has made it even more challenging to search for similar sequences (homologous) for structure or function predictions. Here we show that SGT sequence features can lead to a fast and accurate sequence search.

We collected a random sample of 1000 protein sequences from UniProtKB database on www.uniprot.org. To incorporate the protein sequence lengths for finding similarities, we transform them to euclidean space using *length-sensitive* SGT (with $\kappa = 1$). Thereafter, to reduce the dimension we applied principal component analysis, and preserved the first 40 principal components (explaining ∼83% of variance), denoted by SGT-PC. We arbitrarily chose a protein sequence, Q9ZIM1[5], as the search query. Note that here we denote a protein by its commonly used entry IDs (for eg. Q9ZIM1 used before) in the UniProtKB database.

We compute the euclidean distance (specifically, the manhattan distance) between the SGT-PCs of the query and each sequence in the dataset. The top 10 closest sequences are shown with their SGT-PC distances in Table 7. As a reference, we also show the *identity* between the closely found sequences and the query. An identity between two protein sequences is the edit distance between them after alignment. Here we find identities after a global alignment, and cost of gap-opening as 0 and gap-extension as 1. Note that alignment

[5]The protein sequence of Q9ZIM1 is, `MSYQQQQCKQPCQPPPVCPTPKCPEPC PPPKCPEPYLPPPCPPEHCPPPPCQDKCPPVQPYPPCQQKYPPKSK`

algorithms are approximate heuristics, thus its results should be used only as a guideline, and not ground truth.

We find that the maximum pairwise identity (=46.3%) corresponds to the smallest SGT-PC distance (33.02) for {Q9ZIM1 (query), S9A4Q5}. Also, the identity level decreases with increasing SGT-PC differences, with some minor inconsistencies. Importantly, the computation time for finding SGT-PC differences between query and the entire dataset was 0.0014 sec on a 2.2GHz Intel Core i7 processor, while identity computations took 194.4 sec. Although, the currently in-use methods for protein databases, like BLAST, have a faster alignment and identity computation procedure than a pairwise, it will still be higher than finding vector differences.

If SGT-PC based search is implemented simply like above — find distance from each datapoint in the database —, the runtime increases linearly. We saw runtime increase of about 1000 times to 1.3 sec (from 0.0014 sec) on performing the distance computation on 1 million datapoints. In practice, advanced techniques can be employed, for eg. similar to k-clust or HH-blits, where the sequences are clustered and a cluster has a representative sequence or profile with which the query sequence is compared hierarchically. This reduces the search space – significantly reducing the runtime. Moreover, a parallel architecture can be added to divide the computation across several computing nodes. We leave these advancements for future research.

This concludes our case study, where we showed the application of SGT based sequence mining on real world problems and its compatibility with various distance and graph based data mining tools. In the next section we will discuss our results from this section and Sec. 4.4.

4.6 Discussion

As we showed in Sec. 4.2.4, SGT's ability to capture the overall pattern — short and long range structures — of a sequence into a fixed finite-dimensional space makes it stand out. The n-gram models had lower performance than SGT because of this reason. N-grams cannot explicitly capture long-range dependencies, unless n is large, in which case the feature space

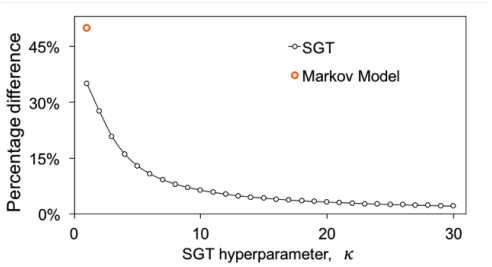


Figure 32: Percentage change in SGT feature for (A,B) with κ in presence of a noise.

becomes extremely large to handle.

Besides, the Markov models were outperformed due to their restrictive Markovian assumption, which was not always true for the data. Moreover, due to this assumption, these Markov models cannot effectively handle inherent stochasticities in a sequence and assess the long-range patterns. For illustration, suppose we have sequences in which "B occurs *closely* after A". Consider one such example, **ABCDEAB**: a first-order Markov model will give a high transition probability (a feature), equal to one, that will correspond to the inherent pattern. But in presence of noise, for eg. a random alphabet, X, appearing in between A and B, **ABCDEAXB**, the transition probability will decrease by 50% (from 1.0 to 0.5). On the other hand, SGT is robust to such noises. As shown in Fig.32, the percentage change in the SGT feature for (A,B), in the above case, is smaller than the Markov and decreases with increasing κ. It also shows that we can easily regulate the effect of such stochasticity, with a caution that sometimes the interspersed alphabets may not be noise but part of the sequence's pattern (thus, we should not set κ as a high value without a validation).

Furthermore, a Markov model cannot easily distinguish between these two sequences: **ABCDEAB** and **ABCDEFGHIJAB**, from the (A,B) transition probabilities. In both sequences, the transition probability will be equal to 1, while the SGT feature for (A,B) changes from 1.72 to 2.94 ($\kappa = 1$), showing SGT's capability in capturing the overall pattern. On another note, although deep learning methods can capture such overall patterns, their representations are in an arbitrary and usually very high dimension.

Thus, SGT proves to be more proficient in capturing sequence patterns. This, aided with the new possibility of using mainstream data mining techniques for sequence analysis,

led to SGT outperforming other state-of-the-art sequence mining methods (see Sec. 4.4-4.5). Moreover, SGT also performs better in runtime. This is because, although SGT's runtime upper bound is proportional to the square of sequence length or alphabet set size, the actual runtime can be significantly lowered by few pre-processing. For example, on implementing Algorithm-2, we can first pre-process a sequence to obtain the list of positions of each alphabet, and then run a nested loop *only* for the alphabets present in the sequence.

In addition, SGT's unique property of including or excluding the sequence length effect makes it compatible for both length sensitive and insensitive sequence problems. We show its efficacy by performing real world data analysis for both cases.

4.7 APPENDICES

4.7.1 Appendix A: Arithmetico-Geometric series

The sum of a series, where the k^{th} term for $k \geq 1$ can be expressed as,

$$t_k = (a + (k-1)d)\, br^{k-1}$$

is called as an arithmetico-geometric because of a combination of arithmetic series term $(a + (k-1)d)$, where a is the initial term and common difference d, and geometric br^{k-1}, where b is the initial value and common ratio being r.

Suppose the sum of the series till n terms is denoted as,

$$S_n = \sum_{k=1}^{n} (a + (k-1)d)\, br^{k-1} \tag{87}$$

Without loss of generality we can assume $b = 1$ for deriving the expression for S_n (the sum for any other value of b can be easily obtained by multiplying the expression for S_n with b). Expanding Eq. 87,

$$S_n = a + (a+d)r + (a+2d)r^2 + \ldots + (a + (n-1)d)r^{n-1} \tag{88}$$

Now multiplying S_n with r,

$$rS_n = ar + (a+d)r^2 + (a+2d)r^3 + \ldots + (a + (n-1)d)r^n \tag{89}$$

Subtracting Eq. 89 from 88, if $|r| < 1$, else we subtract the latter from the former, we get,

$$\begin{aligned}
|(1-r)S_n| &= \Big|\big[a + (\cancel{a} + d)r + (\cancel{a} + \cancel{2}d)r^2 + \ldots + (\cancel{a} + \cancel{(n-1)}d)r^{n-1}\big] \\
&\quad - \big[\cancel{ar} + (\cancel{a} + \cancel{d})r^2 + (\cancel{a} + \cancel{2}d)r^3 + \ldots + (a + (n-1)d)r^n\big]\Big| \\
&= \Big|a + d(r + r^2 + \ldots + r^{n-1}) - (a + (n-1)d)r^n\Big| \\
&= \left|a + \frac{dr(1-r^{n-1})}{1-r} - (a + (n-1)d)\, r^n\right|.
\end{aligned}$$

Therefore,

$$S_n = \left|\frac{1}{1-r}\left[a + \frac{dr(1-r^{n-1})}{1-r} - (a + (n-1)d)\, r^n\right]\right| \tag{90}$$

or, for any value of b,

$$S_n = b\left|\frac{1}{1-r}\left[a + \frac{dr(1-r^{n-1})}{1-r} - (a + (n-1)d)\, r^n\right]\right| \tag{91}$$

4.7.2 Appendix B: Mean and Variance of a graph tranform feature, ψ_{uv}.

Mean

Length-sensitive SGT

We first derive the expected value of ψ_{uv} for a length-sensitive SGT as,

$$\begin{aligned}
E_p[\psi_{uv}] &= \frac{\sum_{d_i \in \mathcal{D}} E_p[\phi_\kappa(d_i)]}{M(M+1)/2} \\
&= \frac{ME_p[\phi_\kappa(Z_1)] + (M-1)E_p[\phi_\kappa(Z_2)] + \ldots + E_p[\phi_\kappa(Z_M)]}{M(M+1)/2} \\
&= \frac{\sum_{m=1}^{M}(M-(m-1))E_p[\phi_\kappa(Z_m)]}{M(M+1)/2}
\end{aligned} \tag{92}$$

Since, X and Y are normally distributed, the pair gap-distance variable, Z in Eq. 76 will be,

$$Z_m \sim N(\mu_\alpha + (m-1)\mu_\beta, \sigma_\alpha^2 + (m-1)\sigma_\beta^2) \tag{93}$$

Therefore, $\phi_\kappa(Z_m) \sim$ lognormal (see Eq. 77), and

$$\begin{aligned} E_p[\phi_\kappa(Z_m)] &= e^{-\kappa(\mu_\alpha+(m-1)\mu_\beta)+\frac{1}{2}\kappa^2(\sigma_\alpha^2+(m-1)\sigma_\beta^2)} \\ &= e^{-\tilde{\mu}_\alpha-(m-1)\tilde{\mu}_\beta} \qquad (94) \\ \mathrm{var}_p[\phi_\kappa(Z_m)] &= \left(e^{\kappa^2(\sigma_\alpha^2+(m-1)\sigma_\beta^2)}-1\right) e^{-2\kappa(\mu_\alpha+(m-1)\mu_\beta)+\kappa^2(\sigma_\alpha^2+(m-1)\sigma_\beta^2)} \\ &= e^{-2\tilde{\mu}'_\alpha-2(m-1)\tilde{\mu}'_\beta} - e^{-2\tilde{\mu}_\alpha-2(m-1)\tilde{\mu}_\beta} \qquad (95) \end{aligned}$$

where,

$$\begin{aligned} \tilde{\mu}_\alpha &= \kappa\mu_\alpha - \frac{\kappa^2}{2}\sigma_\alpha^2 \quad ; \quad \tilde{\mu}'_\alpha = \kappa\mu_\alpha - \kappa^2\sigma_\alpha^2 \\ \tilde{\mu}_\beta &= \kappa\mu_\beta - \frac{\kappa^2}{2}\sigma_\beta^2 \quad ; \quad \tilde{\mu}'_\beta = \kappa\mu_\beta - \kappa^2\sigma_\beta^2 \end{aligned} \tag{96}$$

Substituting the results in Eq. 94 to Eq. 92 and Eq. 90 from Appendix A, we get,

$$\begin{aligned} E_p[\psi_{uv}] &= \frac{\sum_{m=1}^{M}(M-(m-1))e^{-\tilde{\mu}_\alpha-(m-1)\tilde{\mu}_\beta}}{M(M+1)/2} \\ &= \left(\frac{2}{M+1}\right)\left|\left(\frac{e^{-\tilde{\mu}_\alpha}}{1-e^{-\tilde{\mu}_\beta}}\right)\left[1-\frac{1}{M(e^{\tilde{\mu}_\beta}-1)}(1-e^{-M\tilde{\mu}_\beta})\right]\right| \\ &\approx \frac{2}{M+1}\left|\frac{e^{-\tilde{\mu}_\alpha}}{1-e^{-\tilde{\mu}_\beta}}\right| \\ &= \frac{2}{Lp+1}\left|\frac{e^{-\tilde{\mu}_\alpha}}{1-e^{-\tilde{\mu}_\beta}}\right| = \frac{2}{Lp+1}\gamma \qquad (97) \end{aligned}$$

where,

$$\gamma = \left|\frac{e^{-\tilde{\mu}_\alpha}}{1-e^{-\tilde{\mu}_\beta}}\right| \tag{98}$$

Length-insensitive SGT

Next, as given in Eq. 74b for the length insensitive sequence problem, Λ_{uv} is normalized by the sequence length. Thus, the edge weight, $E[\psi_{uv}]$ will be,

$$\begin{aligned} E_p[\psi_{uv}] &= \frac{\sum_{m=1}^{M}(M-(m-1))e^{-\tilde{\mu}_\alpha-(m-1)\tilde{\mu}_\beta}}{\left(\frac{M(M+1)/2}{L}\right)} \\ &\approx \frac{2L}{Lp+1}\left|\frac{e^{-\tilde{\mu}_\alpha}}{1-e^{-\tilde{\mu}_\beta}}\right| = \frac{2L}{Lp+1}\gamma \end{aligned} \tag{99}$$

Variance

Length-sensitive SGT

The variance of an edge feature, ψ_{uv}, can be computed as,

$$\begin{aligned} \mathrm{var}_p(\psi_{uv}) &= \left(\frac{1}{M(M+1)/2}\right)^2 \sum_{m=1}^{M}(M-(m-1))\left[e^{-2\tilde{\mu}'_\alpha-2(m-1)\tilde{\mu}'_\beta} - e^{-2\tilde{\mu}_\alpha-2(m-1)\tilde{\mu}_\beta}\right] \\ &= \left(\frac{1}{M(M+1)/2}\right)^2 \Bigg[\underbrace{\sum_{m=1}^{M}(M-(m-1))e^{-2\tilde{\mu}'_\alpha-2(m-1)\tilde{\mu}'_\beta}}_{V_1} - \\ &\quad \underbrace{\sum_{m=1}^{M}(M-(m-1))e^{-2\tilde{\mu}_\alpha-2(m-1)\tilde{\mu}_\beta}}_{V_2}\Bigg] \end{aligned} \tag{100}$$

Again, both V_1 and V_2 forms an Arithmetico-Geometric series. Solving for them, we get,

$$\begin{aligned} V_1 &= \sum_{m=1}^{M}(M-(m-1))e^{-2\tilde{\mu}'_\alpha-2(m-1)\tilde{\mu}'_\beta} \\ &= e^{-2\tilde{\mu}'_\alpha}\sum_{m=1}^{M}(M-(m-1))e^{-2(m-1)\tilde{\mu}'_\beta} \\ &= \frac{e^{-2\tilde{\mu}'_\alpha}}{1-e^{-2\tilde{\mu}'_\beta}}\left[M - e^{-2\tilde{\mu}'_\beta}\left(\frac{1-e^{-2(M-1)\tilde{\mu}'_\beta}}{1-e^{-2\tilde{\mu}'_\beta}}\right) - (M-(M-1))e^{-2M\tilde{\mu}'_\beta}\right] \\ &= \frac{e^{-2\tilde{\mu}'_\alpha}}{1-e^{-2\tilde{\mu}'_\beta}}\left[M - e^{-2\tilde{\mu}'_\beta}\left(\frac{1-e^{-2M\tilde{\mu}'_\beta}}{1-e^{-2\tilde{\mu}'_\beta}}\right)\right] \end{aligned} \tag{101}$$

Similarly,

$$V_2 = \sum_{m=1}^{M}(M-(m-1))e^{-2\tilde{\mu}_\alpha-2(m-1)\tilde{\mu}_\beta}$$
$$= \frac{e^{-2\tilde{\mu}_\alpha}}{1-e^{-2\tilde{\mu}_\beta}}\left[M - e^{-2\tilde{\mu}_\beta}\left(\frac{1-e^{-2M\tilde{\mu}_\beta}}{1-e^{-2\tilde{\mu}_\beta}}\right)\right] \quad (102)$$

Therefore, plugging in Eq. 101-102 and Eq. 75 into Eq. 100, we get,

$$\text{var}_p(\psi_{uv}) = \left(\frac{1}{Lp(Lp+1)/2}\right)^2\left[\left\{\frac{e^{-2\tilde{\mu}'_\alpha}}{1-e^{-2\tilde{\mu}'_\beta}}\left(Lp - e^{-2\tilde{\mu}'_\beta}\left(\frac{1-e^{-2Lp\tilde{\mu}'_\beta}}{1-e^{-2\tilde{\mu}'_\beta}}\right)\right)\right\} - \left\{\frac{e^{-2\tilde{\mu}_\alpha}}{1-e^{-2\tilde{\mu}_\beta}}\left(Lp - e^{-2\tilde{\mu}_\beta}\left(\frac{1-e^{-2Lp\tilde{\mu}_\beta}}{1-e^{-2\tilde{\mu}_\beta}}\right)\right)\right\}\right]$$

It is easy to show that,

$$\lim_{L\to\infty}\text{var}_p(\psi_{uv}) = 0$$

Length-insensitive SGT

Besides, for length-insensitive case, the variance will be,

$$\text{var}_p(\psi_{uv}) = \left(\frac{1}{p(Lp+1)}\right)^2\left[\left\{\frac{e^{-2\tilde{\mu}'_\alpha}}{1-e^{-2\tilde{\mu}'_\beta}}\left(Lp - e^{-2\tilde{\mu}'_\beta}\left(\frac{1-e^{-2Lp\tilde{\mu}'_\beta}}{1-e^{-2\tilde{\mu}'_\beta}}\right)\right)\right\} - \left\{\frac{e^{-2\tilde{\mu}_\alpha}}{1-e^{-2\tilde{\mu}_\beta}}\left(Lp - e^{-2\tilde{\mu}_\beta}\left(\frac{1-e^{-2Lp\tilde{\mu}_\beta}}{1-e^{-2\tilde{\mu}_\beta}}\right)\right)\right\}\right]$$

which also has limiting value of 0 as the sequence length increases.

4.7.3 Appendix C: Proof for SGT expression for undirected sequences

Proof for Eq. 85:

By definition,

$$\Lambda_{uv}(s) = \{(l,m) : x_l = u, x_m = v, l < m, (l,m) \in 1,\ldots,L^{(s)}\}$$

Similarly, Λ_{uv} can be expressed as,

$$\begin{aligned}\Lambda_{uv}(s) &= \{(l,m):\, x_l = u, x_m = v, l > m, (l,m) \in 1,\ldots,L^{(s)}\} \\ &= \Lambda_{vu}^T(s)\end{aligned}$$

Next, Eq. 83 can be expanded as,

$$\begin{aligned}\tilde{\Lambda}_{uv}(s) &= \{(l,m):\, x_l = u, x_m = v, (l,m) \in 1,\ldots,L^{(s)}\} \\ &= \{(l,m):\, x_l = u, x_m = v, l < m, (l,m) \in 1,\ldots,L^{(s)}\} \\ &\quad +\{(l,m):\, x_l = u, x_m = v, l > m, (l,m) \in 1,\ldots,L^{(s)}\} \\ &= \Lambda_{uv}(s) + \Lambda_{uv}^T(s)\end{aligned}$$

Thus, proving Eq. 84. Next, the SGT for undirected sequence in Eq. 85, can be expressed as,

$$\begin{aligned}\tilde{\Psi}_{uv}(s) &= \frac{\sum_{\forall(l,m)\in\tilde{\Lambda}_{uv}(s)} \phi_\kappa(d(l,m))}{|\tilde{\Lambda}_{uv}(s)|} \\ &= \frac{\sum_{\forall(l,m)\in\Lambda_{uv}(s)} \phi_\kappa(d(l,m)) + \sum_{\forall(l,m)\in\Lambda_{uv}^T(s)} \phi_\kappa(d(l,m))}{|\Lambda_{uv}(s)| + |\Lambda_{uv}^T(s)|} \\ &= \frac{|\Lambda_{uv}(s)|\Psi_{uv}(s) + |\Lambda_{uv}^T(s)|\Psi_{uv}^T(s)}{|\Lambda_{uv}(s)| + |\Lambda_{uv}^T(s)|}\end{aligned}$$

Thus, proving Eq. 85.

4.7.4 Appendix D: Proof for Alphabet Clustering

We have,

$$\begin{aligned}\frac{\partial\Delta}{\partial\kappa} &= \frac{\partial}{\partial\kappa}E[\phi_\kappa(X) - \phi_\kappa(Y)] \\ &= E[\frac{\partial}{\partial\kappa}\phi_\kappa(X) - \frac{\partial}{\partial\kappa}\phi_\kappa(Y)] \qquad (103)\end{aligned}$$

For $E[X] < E[Y]$, we want, $\frac{\partial\Delta}{\partial\kappa} > 0$, in turn, $\frac{\partial}{\partial\kappa}\phi_\kappa(X) > \frac{\partial}{\partial\kappa}\phi_\kappa(Y)$ (from Eq. 103). This will hold, if

$$\frac{\partial^2}{\partial d \partial \kappa}\phi_\kappa(d) \quad > \quad 0 \tag{104}$$

that is, slope, $\frac{\partial}{\partial \kappa}\phi_\kappa(d)$ is increasing with d. For an exponential expression for ϕ (Eq. 72, the condition in Eq. 104 holds true if $\kappa d > 1$. Hence, under these conditions, the *separation* increases as we increase the tuning parameter, κ.

4.8 References

Altschul, S. F., Madden, T. L., SchÃ€ffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research, 25(17), 3389-3402.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2003). Model-based clustering and visualization of navigation patterns on a web site.Â Data Mining and Knowledge Discovery,Â 7(4), 399-424

Comin, M., & Verzotto, D. (2012). Alignment-free phylogeny of whole genomes using underlying subwords. Algorithms for Molecular Biology, 7(1), 1.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (2), 224-227.

Didier, G., Corel, E., Laprevotte, I., Grossmann, A., & LandÃšs-Devauchelle, C. (2012). Variable length local decoding and alignment-free sequence comparison. Theoretical Computer Science, 462, 1-11.

Ding, C. H., & Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics, 17(4), 349-358.

Dong, G., & Pei, J. (2007). *Sequence data mining* (Vol. 33). Springer Science & Business Media.

Edgar, R. C. (2004). MUSCLE: a multiple sequence alignment method with reduced time and space complexity.Â BMC bioinformatics,Â 5(1), 113.

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST.Â Bioinformatics,Â 26(19), 2460-2461.

Fu, L., Niu, B., Zhu, Z., Wu, S., & Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data.Â Bioinformatics,Â 28(23), 3150-3152.

Gaber, M. M. (2009). *Scientific data mining and knowledge discovery.* Springer. p207-247.

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Hauser, M., Mayer, C. E., & SÃ¶ding, J. (2013). kClust: fast and sensitive clustering of large protein sequence databases. BMC bioinformatics, 14(1), 1.

Helske, S., & Helske, J. (2016). Mixture Hidden Markov Models for Sequence Data: the seqHMM Package in R.

Kumar, P., Krishna, P. R., & Raju, S. B. (2012). *Pattern discovery using sequence data mining: applications and studies.* Information Science Reference.

Linial, M., Linial, N., Tishby, N., & Yona, G. (1997). Global self-organization of all known protein sequences reveals inherent biological signatures. Journal of molecular biology, 268(2), 539-556.

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins.Â Journal of molecular biology,Â 48(3), 443-453.

Pearson, W. R. (1990). [5] Rapid and sensitive sequence comparison with FASTP and FASTA. Methods in enzymology, 183, 63-98.

Ranjan, C., Paynabar, K., & Helm, J. E. (2015). Heterogeneous Elective Inpatient Flow Modeling and Scheduling.Â arXiv preprint arXiv:1505.07752.

Remmert, M., Biegert, A., Hauser, A., & SÃ¶ding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nature methods, 9(2), 173-175.

Siyari, P., Dilkina, B., & Dovrolis, C. (2016). Lexis: An Optimization Framework for Discovering the Hierarchical Structure of Sequential Data. arXiv preprint arXiv:1602.05561.

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. Journal of molecular biology, 147(1), 195-197.

Tomović, A., Janičić, P., & KeÅ¡elj, V. (2006). n-Gram-based classification and unsupervised hierarchical clustering of genome sequences. Computer methods and programs in biomedicine, 81(2), 137-153.

CHAPTER V

CONCLUSION AND FUTURE DIRECTIONS

This dissertation focused on developing new statistical learning and data mining methods for service systems improvement. From a large pool of research problems in this area, three important and challenging problems were studied and novel methodologies were developed. Each of these methodologies is shown to have a wide variety of applications like healthcare resource optimization, medical decision making, web data analysis, bioinformatics, business, and more.

Novel research contributions were made in each methodology. They were experimentally validated, and real world applications were demonstrated through case studies. In the following, the developed research methodologies and their contributions are summarized. Also, future research directions are suggested.

- **The Impact of Estimation: A New Method for Clustering and Trajectory Estimation in Patient Flow Modeling**

The *Hospital Admission Scheduling and Control* problem is comprised of two main components: *census modeling* and *resource scheduling*. Previous work on this long-standing problem has considered one or the other, but not both. In this research, we develop a new method based on *semi-Markov model* (SMM) clustering for identifying patient type clusters and estimating cluster trajectory distributions that integrates seamlessly with existing scheduling optimization approaches. This integration is proven to be extremely important, as optimal solutions using our SMM approach dramatically outperform optimal solutions using the traditional empirical estimation techniques.

As a theoretical contribution, our novel approach is able to model an entire hospital of any size as a coordinated system with complex, general network of wards and patient transitions between them. Further, the model has been shown to be *scalable*, accounts for *ward interactions*, and for patient *heterogeneity*, which has not been previously achieved by

other methods in the literature. Further, our SMM-clustering is a general purpose algorithm applicable to any movement or sequence data having spatial and temporal dimension, for example, *clickstream* data of users on a website or movement of cell-phone users among a network of towers.

Our SMM approach was designed to integrate with resource scheduling approaches that provide an optimal controllable schedule by patient type for each day of week. Thus, this approach can be adopted by any specialty or multi-specialty hospital for streamlining their procedures, stabilizing the operating environment for their personnel, efficient utilization of hospital resources, and cost savings for both patients and hospitals. The automated, algorithmic approach for clustering and trajectory estimation is also appealing compared to ad-hoc, manual, and heuristic approaches currently employed in practice (which can take months to implement and are difficult to validate statistically).

The SMM-clustering method was validated by simulating data from *known* generating mixture distributions. The SMM estimated clusters and their distributions were found to be statistically the same as the generating mixture distributions at a 95% confidence level. We validated the efficacy of the developed Clustering and Scheduling Integrated (CSI) method under simulated conditions based on a small hospital scenario, where we know the true number of patient types, how they are clustered, and their trajectory distributions. Optimizing the elective schedule based on inputs from our SMM method achieved outcomes that were very close to the the "true optimum" (i.e. given perfect knowledge of patient flow dynamics), while the existing traditional method gave significantly worse results.

A case study using real hospital data showed that the number of elective admissions could be increased by 97% (with the same level of access) compared to only a 30% increase using traditional empirical methods (which are comparable to previous optimization improvements reported in the literature). Moreover, the average ward utilization could be improved by 22% using our approach compared with only an 8% improvement using the traditional approach.

In conclusion, our approach develops a novel method for spatio-temporal clustering and estimation that has a profound impact on an important patient flow problem with the potential to improve revenues and/or cost, quality, and access to care.

For a future research, other potential ways to model patient flow when the underlying Markovian distribution may not hold should be explored. Besides, incorporating several other real world factors, like patient movement blocking (patient diverted to another ward due unavailability), effects from resource shortages, and dynamic changes due to any change in hospital wards layout can be considered.

- **Longitudinal MRI Data Analysis in presence of Measurement Error but absence of Replicates**

Existing methods rely on a complete sanctity of data for analyzing them and drawing inferences. This can be problematic if the available data has measurement errors, especially in medical decision making. This research focused on analysis of longitudinal data in a challenging scenario, when there are measurement errors but replicates are not available. Our major contribution was development of a new EM-Variogram technique for estimating an extended linear mixed effect model with a parametric covariance structure. The parametric covariance expression decouples the measurement error variance from the overall variance. The developed estimation technique, thus, isolates the measurement error, and hence, provides a more accurate effect estimation and statistical inferences.

The performance of the model was experimentally validated via simulations for various scenarios. We find that the methodology is effective and accurate in modeling, and is robust to missing values, commonly found in a longitudinal data. The methodology was also applied to a longitudinal MRI dataset for evaluation of hippocampal volume in (potential) Alzheimer disease patients with mild cognitive impairment (MCI). The measurement error variance was accurately decoupled from the noise variance. After isolating the measurement error, we were able to obtain more precise (narrower) confidence interval for the effect estimates, leading to more powerful statistical tests. Moreover, it can also detect more subtle effects with less data and, thus, can a) do an early detection of a patient's condition allowing necessary treatment assessment and diagnosis at early stages of the disease, and b) reduce costs of longitudinal studies.

Besides, the numerical experiments show that the proposed method may have relatively

poor performance in random effect variance estimations if the random noise in the model is relatively high. However, this situation typically indicates that the selected model is much different from the true underlying model, and thus, can be resolved with a better model selection. In practical applications, one can find a proper choice for the underlying model by trying a wide range of model and comparing them using a model selection criterion, like, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), etc. Moreover, an effective model selection is also necessary to keep the modeling error minimal, an underlying assumption of the developed approach.

In conclusion, the experimental and the case study results indicate that the proposed methodology can effectively address the analysis problem for longitudinal data where replicated measurement are very costly or difficult to take, for example drug efficacy tests, destructive tests, etc. Besides, the proposed method is flexible and can be used in cases where other covariates (e.g., sex, age, etc.) besides time are included in the model.

In this research, we make assumptions on the homogeneity of measurement device conditions over time, which makes the reproducibility analysis irrelevant. However, the extension of the proposed methodology for a simultaneous gauge repeatability and reproducibility (GRR) analysis would be a topic of interest for future research. Another potential direction for future research, is to develop methods for GRR study of multivariate longitudinal data streams.

- **Sequence Graph Transform (SGT): A Feature Extraction Function for Sequence Data Mining**

Sequence data mining is an important but challenging area due to absence of a method to effectively quantify a sequence's patterns characteristics in a finite-dimensional euclidean space. We made a significant contribution in this area by developing a new function, Sequence Graph Transform (SGT), for this purpose. The proposed SGT has two main variants for *length-sensitive* and *length-insensitive* sequence problems.

SGT's ability to capture the overall pattern — short- and long0 range structures — of a sequence into a fixed finite-dimensional space makes it stand out. To justify the claim,

this pattern capturing ability was theoretically proved. Importantly, SGT is capable of extracting the short- and long- term patterns with a small computational complexity.

Besides, this pattern capturing ability was further discussed and contrasted with a first-order Markovian model. It showed that a Markov model cannot always differentiate between two sequences from transition probabilities even when their overall patterns are different. Also, they are not robust to presence of noises in a sequence. On the other hand, SGT is shown to be efficient in both cases.

SGT's performance was validated via clustering of synthesized sequence datasets under various scenarios. A clustering operation was used as it requires an accurate comparison of sequences. In all the scenarios, SGT is shown to markedly outperform other sequence clustering methods, viz. n-gram, (semi, hidden)-Markov models, CD-HIT, MUSCLE and UCLUST. Besides, SGT's runtime is also shown to be smaller.

SGT's application in four main sequence mining areas: clustering, classification, visualization and search operations is shown, using four real datasets. Mainstream data mining techniques could be applied on the sequence datasets via SGT. A k-means clustering is performed on the SGT's of sequences from msnbc.com user navigation sequence dataset. A graph-visualization is used to interpret the k-means clustering results to understand the user behaviors. Besides, Support Vector Machines (SVM) are fitted for classification models on SGT's of a protein and network intrusion sequence datasets. Lastly, a search operation is demonstrated using principal component analysis (PCA) on SGT's of a different protein sequence dataset.

SGT is, thus, shown to bridge the gap between sequence mining and mainstream data mining methods by mapping sequences in a euclidean space. Some extensions of SGT for "undirected" sequence and alphabet clustering, are also provided, and tested. Moreover, it has been shown to have a universal applicability to any sequence problem with various applications. Importantly, due to its low computational complexity and ease of parallelization it can be scaled to any big sequence data problem.

For future research, we should attempt to use SGT to develop new methods for diverse sequence problems in speech recognition, text analysis, bioinformatics etc., or use it as an

embedding layer in deep learning architectures.