Machine Learning in Healthcare: An Investigation into Model Stability

by

Shivapratap Gopakumar

M.Tech

Submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Deakin University

February 2017

DEAKIN UNIVERSITY
CANDIDATE DECLARATION

I certify the following about the thesis entitled (10 word maximum)

Machine Learning in Healthcare: An Investigation into Model Stability

submitted for the degree of **Doctor of Philosophy**

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

d. That any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

e. All research integrity requirements have been complied with.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: **Shivapratap Gopakumar**

Signed: Signature Redacted by Library

Date: **08/02/2017**

DEAKIN UNIVERSITY
ACCESS TO THESIS - A

I am the author of the thesis entitled **Machine learning in Healthcare: An Investigation into Model Stability,** submitted for the degree of **Doctor of Philosophy**

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: SHIVAPRATAP GOPAKUMAR

Signed: Signature Redacted by Library

Date: 06/07/2017

Contents

List of Figures

List of Tables

Abstract

STABILITY is fundamental to prognosis. Besides good performance, a prognostic model needs to be interpretable and stable to warrant clinical adoption. This translates to a small group of succinct predictors that are consistent in the face of data re-sampling. Hence strong feature selection is key when deriving clinical models.

It has been found that when data is high dimensional and correlated, automated feature selection causes instability in clinical prediction models. But these aspects are intrinsic to modern healthcare data. A typical patient database will contain details on demographics, history of hospital visits, diagnosis, procedures, physiological measurements, bio-markers and interventions that are recorded over time. Further, in such high-dimensional data, medical conditions often co-occur, especially in aged cohorts. Comorbidities or diseases that co-exist with the primary disease in a patient, cause multiple diagnoses that are strongly correlated to each other. Applying traditional methods for sparse feature selection results in instability in feature subsets and feature weights.

In this thesis, we address the open problem of stable feature selection in clinical settings, to ensure the stability of predictors in a linear prognostic model derived from patient data in electronic medical records (EMRs). We begin by demonstrating the problem of instability in clinical prediction for a patient flow problem. To date, there has been limited work in predicting ward-level discharges. Our case study for model instability investigates forecasting total next-day discharges from an open ward. We build seven prediction models from administrative data stored in hospital records. On patient data of four years, we find the performances of predictive models to be comparable. Yet, the model estimations and predictors exhibit instability under data resampling. Including clinical information could enhance predictive performance, but also aggravate instabil-

ity. We conclude our case study by proposing a stabilization framework for linear models using lasso regularization.

In our first stabilization scheme, we propose a knowledge-based approach, by exploiting inherent temporal and semantic relationships in medical data. To reduce variance in the selected features that are predictive of prognosis, we introduce Laplacian based regularization into a regression model. The Laplacian is derived on a feature graph that captures (i) temporal relations in diagnosis, prognosis and intervening events, and (ii) hierarchical structures of disease family through semantics in the diagnosis codes. Using a large cohort of patients with myocardial infarction, we demonstrate better stability through feature graph stabilization.

For our second stabilization scheme, we extend our feature graph regularization to discover underlying statistical relations in training data. We examine the effect of different feature graphs constructed from common statistical similarity measures. An aggregate graph that combines the semantic and statistical relations is also derived. All experiments are performed on a Cox time-to-events model derived from two real-world datasets. We demonstrate that the feature graph regularization built from Jaccard scores and aggregate scores improved stability without hurting predictive performance (measured as AUC). The Jaccard graph regularization proved to be the best for stabilizing parameter weights, whereas aggregate Jaccard scores and semantic EMR graph was superior in stabilizing feature subsets. Transferring Jaccard scores from a related cohort also improved stability when compared with lasso and elastic net.

Our third and final stabilization scheme exploit higher order correlations in training data. Using a linear model as basis for prediction, we achieve feature stability by regularizing latent correlation in features. Latent higher order correlation among features is modelled using an autoencoder network. Stability is enhanced by combining our previous feature graphs and augmenting external unlabelled data during autoencoder training. Our methods demonstrate significant improvement in feature stability and model estimation stability when compared to baselines.

Acknowledgements

SHOULDERS of many giants have been graciously offered in building this thesis. I am deeply indebted to the following people and I dedicate my thesis to them.

To my principal supervisor, Dr Truyen Tran, for his constant support, expert guidance and encouragement throughout my research. If not for your patience and wisdom, my thesis would have been a frustrating and overwhelming pursuit.

To my co-supervisor, Prof. Dinh Phung, for taking time out of his busy schedule to provide support, feedback and constructive criticisms during the course of my research.

To my co-supervisor, Prof. Svetha Venkatesh, for being the Harvey Specter to my Mike Ross. You taught me to get it together and win big, instead of losing small. I deeply cherish all our sessions, and consider myself very fortunate to be your student.

To all postdocs, and my mates in the lab, Adham, Pratibha, Viet, Tu, Cheng, Iman for your support. The discussions we had greatly helped to shape my perspective.

To my parents and my wife, without whose support and love this would not have been possible.

And finally, to my spiritual guru Satguru Sri Mata Amritanandamayi Devi, who has guided me every step of my life.

कामादि सर्प व्रज गारुडाभ्यां विवेक वैराग्य निधि प्रदाभ्याम्
बोध प्रदाभ्यां द्रुत मोक्षदाभ्यां नमो नमः श्री गुरुपादुकाभ्याम्

Relevant Publications

Part of this thesis has been published as manuscripts as detailed below:

Chapter 3:

- **S. Gopakumar**, T. Tran, W. Luo, D. Phung, and S. Venkatesh. "Forecasting Daily Patient Outflow From a Ward Having No Real-Time Clinical Data". In: *JMIR Med Inform 4.3* (2016), e25 , 2016, DOI: 10.2196/medinform.5650
- **S.Gopakumar**, T.Tran, W.Luo, D.Phung, and S.Venkatesh."Forecasting patient outflow from wards having no real-time clinical data". In: *Proceedings of IEEE International Conference on Healthcare Informatics*. Chicago, USA, pp.177–183, 2016.

Chapter 4:

- **S. Gopakumar**, T. Tran, T. D. Nguyen, D. Phung, and S. Venkatesh. "Stabilizing High-Dimensional Prediction Models Using Feature Graphs". In: *IEEE Journal of Biomedical and Health Informatics*, 19.3, pp. 1044– 1052, 2015, DOI: 10.1109/JBHI.2014.2353031

Chapter 5:

- **S. Gopakumar**, T. Nguyen, T. Tran, D. Phung, and S. Venkatesh. "Stabilizing Sparse Cox Model Using Statistic and Semantic Structures in Electronic Medical Records". In: *Advances in Knowledge Discovery and Data Mining*. Vol. 9078.

Springer International Publishing, 2015, pp. 331–343. DOI: 10.1007/978-3-319-18032-8_26. ***(Awarded best student paper runner up)***

- **S. Gopakumar**, T. Tran, D. Phung, and S. Venkatesh. "Stabilizing Sparse Cox Model using Clinical Structures in Electronic Medical Records", In: *Proceedings of the Second International Workshop on Pattern Recognition for Healthcare Analytics*, Sweden, 2014.

Chapter 6:

- **S.Gopakumar**, T.Tran, D.Phung, and S.Venkatesh."Stabilizing Linear Prediction Models using Autoencoder". In: *Proceedings of the 12th International Conference on Advanced Data Mining and Applications (ADMA)*. pp. 651-663, 2016

Abbreviations

Abbreviation	Description
AE	Autoencoder
AG	Augmented data
ARIMA	Autoregressive Integrated Moving Average
ARMAX	Autoregressive moving average with exogenous variables
AUC	Area Under the Receiver Operating Characteristic Curve
CI	Consistency Index
DB	Diabetes
ED	Emergency Department
EMR	Electronic Medical Records
HF	Heart failure
ICD-10	International Classification of Diseases (version 10)
kNN	k-nearest neighbour
Lasso	Least Absolute Shrinkage and Selection Operator
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MFE	Mean Forecast Error
PCA	Principal Components Analysis
RBF	Radial Basis Function
RF	Random Forests
RMSE	Root Mean Square Error
sMAPE	symmetric Mean Absolute Percentage Error
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
SVR	Support Vector Regression
TL	Transfer Learning

"Non-reproducible single occurrences are of no significance to science."

The Logic of Scientific Discovery, Karl Popper

Chapter 1

Introduction

SCIENCE advances through corroboration. Repeatability and reproducibility form cornerstones of scientific method. Karl Popper, one of the greatest science philosophers of the 20^{th} century, in his seminal work "The Logic of Scientific Discovery" (Popper, 1959), writes:

> "We do not take even our own observations quite seriously, or accept them as scientific observations, until we have repeated and tested them. Only by such repetitions can we convince ourselves that we are not dealing with a mere isolated *coincidence*, but with events which, on account of their regularity and reproducibility, are in principle inter-subjectively testable". (p23)

More than 50 years later, a survey by Nature (Baker, 2016) asked 1,576 scientists in fields ranging from physics to biomedicine: "How much published work in your field is reproducible?" resulting in some shocking statistics (see Figure 1.1). Around 70% of researchers failed to reproduce another experiment, while close to 50% failed to reproduce their own experiment. More than half of the surveyed scientists agreed to a significant crisis of reproducibility. This is especially a cause for concern in fields like medicine, where pharmaceutical and biotechnology industries rely on scientific results for new therapeutics and biomarkers. Repeatable results in research is imperative in this era of evidence based medicine.

Steyerberg (2009) defines evidence based medicine as "the conscientious, explicit and

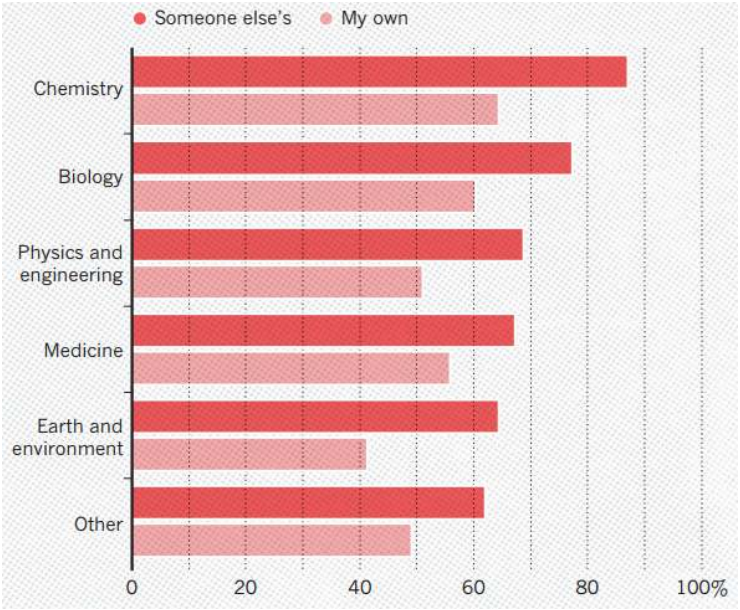


Figure 1.1: Results of "Have you failed to reproduce an experiment?" surveyed from 1,576 scientists. Figure adapted from (Baker, 2016)

judicious use of current best evidence in making decisions about the care of individual patients." In this regard, clinical prediction models play a vital role in providing statistical evidence that helps determine whether a certain outcome is present in an individual (diagnosis) or will occur (prognosis) . The conservative estimate of medical risk derived by such models can be used to identify high risk patients and can be translated into treatment decisions by the clinicians (Moons et al., 2009; Harrell, 2015). Recent advances in machine learning have resulted in increasing popularity of clinical prediction models for statistical analysis of high dimensional patient data from hospital databases (Obermeyer and Emanuel, 2016; Thottakkara et al., 2016; Kourou et al., 2015; Steyerberg, 2009). This shifts the burden of reproducibility to statisticians and data scientists responsible for formulating the analysis. Yu (2013) articulated this concern by characterizing reproducibility as statistical stability claiming: "*At a minimum, reproducibility manifests itself in the stability of statistical results relative to reasonable perturbations to data and to the method or model used.*" Commonly, stability relates to robust performance against reasonable perturbations in data, achieved through diverse methods such as jackknife, bootstrap or cross-validation. The stability of selected features is often overlooked in prediction models – particularly if consistent performance alone is the goal.

But feature stability matters. Even when the prognosis performance is robust. When

building models from high dimensional data, feature selection algorithms choose a small subset of features that maximizes model performance. These features, predictive of the prognosis, are important because they could be hypothesis generating thus meriting further investigation (Saeys et al., 2008). In clinical situations, explaining the prognosis is as important as the prognosis itself. Consequently, consistent predictors in spite of data resampling, are critical for clinical adoption. Since clinicians rely on the set of predictors chosen by the model to understand the prognosis, uncertainty in predictors and estimates also need to be quantified for clinical adoption of the model. Feature stability is crucial not only in clinical prognosis – as example, stable biomarkers aid model reproducibility in bioinformatics (Awada et al., 2012; Khoshgoftaar et al., 2013).

Unfortunately, the nature of clinical data introduces several challenges. Clinical prediction models are built on data largely derived from medical records. The rising popularity of Electronic Medical Records (EMRs) is good news for data mining researchers, as these databases are a potential goldmine of medical knowledge. However, deriving a prediction model from EMR is a challenging task, largely due to the nature of the data. EMR data can be characterized as temporal, high dimensional and highly correlated (He et al., 2013; Jensen et al., 2012). It contains thousands of diagnoses, procedures and medications, many of which may be absent for some patients. Some features may have different values over time (e.g. blood pressure, sodium level). As EMRs are collected mainly for administrative and billing purposes, the recording of medical events and measurements are episodic and irregular. Getting labelled quality data is difficult - generated samples are characterized by small size and high dimensionality - causing models to overfit. Multiple recording schemes and possibility of duplicate clinical entries result in noisy data with high correlation and redundancy. As example, Figure 1.2 illustrates the interactions between diagnosis codes in a cohort diagnosed with heart failure. There is significant correlation among various medical conditions (as represented by the edge thickness in the graph).

Automatic feature selection from such data has been known to cause instability in linear (Austin and Tu, 2004) and survival models (Lin and Lv, 2013). But these models are most preferred among clinicians due to their ease of formulation and interpretation. Hence there is an urgent need to look beyond traditional methods for feature selection. In this thesis, we propose alternative regularization schemes to simultaneously prevent overfitting and guarantee stable models.

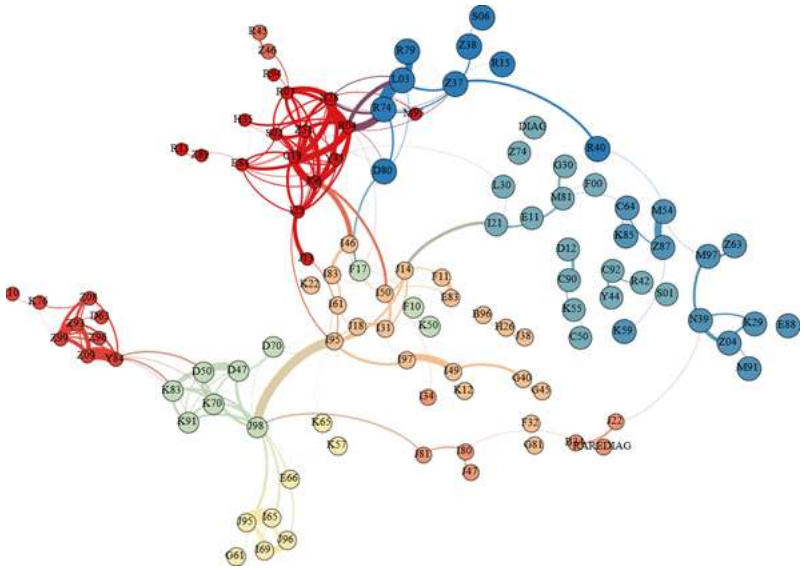


Figure 1.2: A part of comorbidity cluster: co-occurring diseases in a heart failure cohort of over 1000 patients. Nodes represent disease codes – size proportional to prevalence. Edges represent correlation strength.

1.1 Aim and Scope

Our central research question in this thesis can be stated as: "*Can we ensure the stability of predictors in a linear model for prognosis using EMR data?*" To measure the performance of this stability, we adopt variance in selected model parameters across data resampling. Stabilizing sparse clinical prediction involves two steps: (1) Identifying the inherent structure and correlations in data, and (2) employing the discovered data relationships to guide model selection (characterized by predictors and weights). We focus on lasso based methods that favour simultaneous regularization and model selection. We investigate three broad stabilization schemes:

- *Stabilization using data correlations and structures from prior medical knowledge.* The patient medical records in EMR database employ different coding systems to document patient condition and disease progression. These codes for diagnosis, procedures and medications are structured and possibly related in time. We utilize such pre-defined relationships to regularize sparse feature selection.
- *Stabilization using statistical relationships in medical data.* We look beyond predefined relationships to investigate statistical correlations in medical data. We

employ popular data driven methods for statistical correlation to stabilize high-dimensional learning. Open questions include: Does combining statistical relationships with prior knowledge lead to better stability? Can we transfer such knowledge among cohorts for generalization ?

- *Stabilization using higher-order correlations in medical data.* Our final aim is to include all orders of data correlation in guiding feature selection. To this end, we factorize the learning parameters of the model and capture higher order data correlations using a classical autoencoder. We address the following open questions: Does incorporating higher-order correlations improve sparsity as well as stability? Can we use principles of self-taught learning (Raina et al., 2007) to improve and generalize high-dimensional clinical prediction?

To achieve these aims, we propose approaches grounded in machine learning theory and healthcare analytics.

1.2 Significance and Contributions

Our novelty is to identify the importance of stable feature selection in a clinical setting and to propose solutions based on additional regularization of a lasso model by exploiting feature relationships discovered using knowledge driven and data driven methods. Specifically, embedding these relations reduces the fragmentation of selection in the lasso model, delivering our goal of feature stability. The significance of our contribution is to reset the thinking of prognosis from "*model performance only*" to "*model performance and feature stable models*"—without these two components, many of our advanced models will be rendered futile in a clinical setting. The main contributions of this thesis are as follows.

- Our methods were derived and validated on *real-world patient data* from Barwon Health, a regional hospital in Victoria, Australia. Hence our proposed models demonstrate potential to be included in clinical pathway.
- Our proposed approaches illustrate a nexus between modern healthcare and machine learning techniques. Specifically, it systematically examines the applicability of recent advances in machine learning (such as structured regularization,

autoencoders) to recent advances in healthcare (such as secondary use of patient records).

- Our methods propose a structural representation of medical knowledge using feature graphs, where nodes represents EMR features and edges represent feature relationships. We look at knowledge driven and data driven relationships. Initially we use the hierarchical nature of diagnosis and procedure codes along with the temporal nature of recording in building feature graphs. Next, we model feature relationships using common statistical measures such as RBF similarity, Euclidean, Cosine, and Jaccard similarities derived directly from the given patient data. Finally, we construct an aggregated feature graph by combining statistical and semantic relationships.

- We propose pairwise and groupwise regularizers for stabilizing lasso-based models using our constructed feature graphs. Using our statistical and semantic graphs, we formulate (i) Lagrangian regularizer that focuses on pairwise similarity, and (ii) random walk regularizer that encourages groupwise similarity, in stabilizing high dimensional clinical prediction. On two of the most popular clinical models: logistic regression and cox regression, our methods demonstrate superior performance in improving feature subset stability and model estimation stability using measures of Consistency index and signal-to-noise ratio (SNR), when compared to the standard baselines. These results were verified for $1,784$ index admissions in heart failure patients and $2,370$ index admissions in diabetic patients.

- We propose a stabilization scheme by detecting higher order feature correlations. Using a linear model as basis for prediction, we achieve feature stability by regularizing latent correlation in features. We factorize the model parameter into two: (i) a lower dimensional vector, stable and easy to learn, and (ii) high dimensional matrix, that captures all order of correlations in data. This high dimensional component of the linear model is then jointly modelled as encoding weights in an autoencoder network and is used to regularize the prediction model. This approach can be combined with graph based regularization and demonstrates superior stability properties while encouraging model sparsity.

- Finally, we demonstrate the efficacy of our proposed methods for transfer learning and self-taught learning. Collecting data is expensive. Since related cohorts

may share many common predictors and comorbidities, transferring feature graphs among such cohorts can improve stability. Also, a robust estimation of higher order correlation can be performed by augmenting external training data during autoencoder learning. We demonstrate through our experiments that when getting high quality training data becomes difficult, transferring feature graphs or augmenting autoencoder learning with external unlabelled data ensures stable models without loss in prediction performance.

1.3 Outline of Thesis

This thesis contains 7 chapters with supplementary sections in the Appendix. The rest of the thesis is organized as follows:

Chapter 2: presents literature and background relevant to the thesis. This chapter consists of two major sections. The first section reviews popular approaches in healthcare analytics including electronic medical records (EMRs) as data source, popular prediction models, and evaluation measures. The second section focuses on the different aspects of stability, with emphasis on feature selection stability, popular techniques for stabilization and common metrics for evaluating stability.

Chapter 3: opens the Pandora's box of sparsity and stability for a relatively simple but important forecasting problem in hospitals. Specifically, we tackle the problem of predicting next day discharges from a ward using administrative data. To this purpose, we derive 7 popular regression models. While the performance is comparable, we face instability in model parameters. We conclude the chapter by introducing three strategies for model stabilization.

Chapter 4: details the first strategy for stabilization using a knowledge driven approach. For prognosis, we use a logistic regression model for 6 months readmission after heart failure - a deadly and costly disease with a majority of patients returning within a year after discharge. Automatic feature selection was achieved by the sparsity-promoting shrinkage method of lasso. To stabilize this model, we hypothesize exploiting the inherent structures of EMR data to enforce statistical sharing. We consider temporal and hierarchical structures. Since features are accumulated over multiple time granularities

(1 month, 3 months, etc.), features that lie in consecutive time periods are considered to be related. The hierarchies are exploited through the semantics in the ICD-10 tree and the procedure cube (ACHI) - codes that share similar prefix are considered to be related. We embed these relations in a feature graph and add the feature graph regularization term into the lasso model to stabilize heart failure readmission in 6 months.

Chapter 5: extends our work in the previous chapter by using data driven methods to construct feature graphs. These feature graphs are characterized using nodes as EMR features and edges as relationship between features. To model feature relationships, we use popular measures as RBF, Euclidean, Cosine and Jaccard similarity. A random walk regularization of the proposed graphs is used to stabilize a sparse Cox model that predicts time to readmission. Our experiments are conducted on two real world hospital datasets: a heart failure cohort and a diabetes cohort. We measure feature stability using the Consistency index and model estimation stability using signal-to-noise ratio (SNR). Using the best performing Jaccard graph as basis we propose two more graphs: (i) aggregate of Jaccard score and the semantic EMR link used in previous chapter (ii) Jaccard scores between features transferred from a related cohort. Our experiments demonstrate superior performance during graph aggregation and transfer learning.

Chapter 6: proposes a novel methodology to stabilize a sparse high dimensional linear model using recent advances in deep learning and self-taught learning. We propose that the linear model parameter $\boldsymbol{w}$ is a combination of a lower dimensional vector u, and a high dimensional matrix W, where W encapsulates the feature correlations. By modelling W as the encoding weights of an autoencoder network, we capture higher order feature correlations in data. We introduce three regularizers for our sparse linear model: 1) autoencoder derived from training cohort, 2) combination of autoencoder and feature graph derived from training cohort, 3) combination of feature graph derived from training cohort and autoencoder derived from augmenting an external cohort to training data. This process of augmenting external data to autoencoder training results in more robust estimation of higher order correlation matrix W. We demonstrate the efficacy of our proposed stabilization schemes on heart failure cohort from a regional Australian hospital.

Chapter 7: presents the conclusion and future work for this thesis.

"If I have seen further, it is by standing on the shoulders of giants."

Sir Issac Newton

Chapter 2

Background

QUESTIONS addressed in this thesis are at the intersection of healthcare and machine learning. Our work uses machine learning techniques to stabilize high-dimensional linear clinical prediction. The goal of this chapter is to introduce different concepts used throughout our thesis, along with a review of current work and background in healthcare analytics and stability.

We have divided this chapter into two main sections. The first section on healthcare analytics reviews the type of clinical data used for our work. We then present popular linear prediction models in medicine. The second section gives an overview of different types of stability in prediction models. We focus on stability of selected feature subsets and feature weights and review popular techniques and measures.

2.1 Healthcare Analytics

Healthcare analytics is a broad term used to describe the analysis of healthcare data using machine learning techniques. Recently, electronic medical records (EMRs) have become a popular data source for this process. We begin this section by providing an overview of EMR structure to store patient data. We then list various secondary uses of EMRs. We highlight two popular applications derived from EMR data: (i) Patient flow analysis, and (ii) Risk prediction using clinical prediction models.

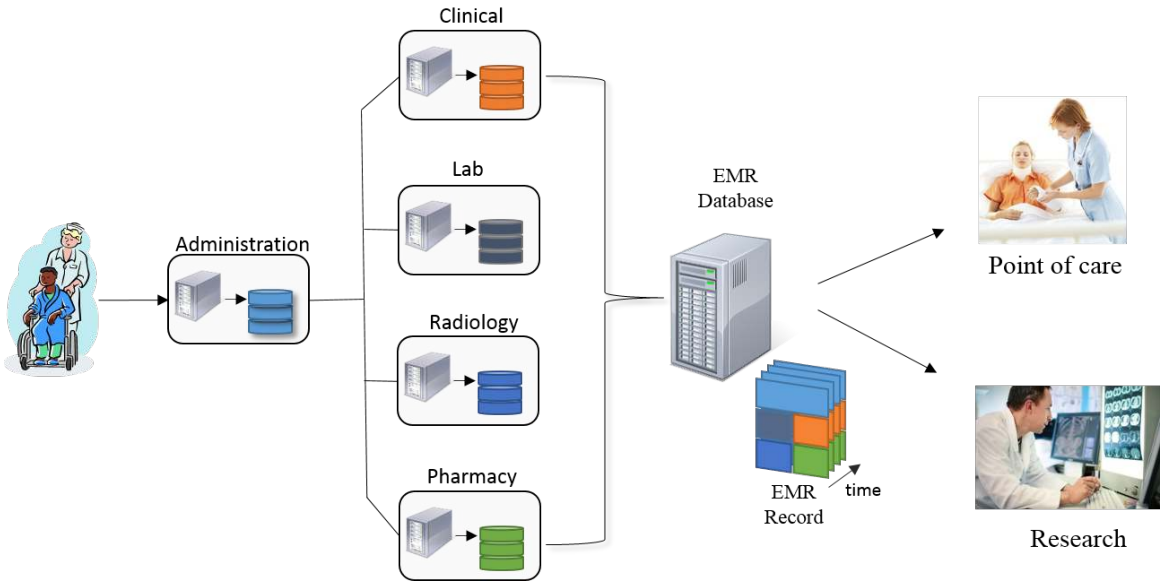


Figure 2.1: Basic components of an EMR system

2.1.1 Electronic Medical Records

An Electronic Medical Record (EMR) is a digital version of a patient's medical history. The primary purposes of EMR are setting objectives, planning patient care, documenting the delivery of care and assessing the outcome of care (Häyrinen et al., 2008). Ideally, EMR handles data over all aspects of care over time (Jensen et al., 2012), and the data is recorded using controlled vocabularies (Section 2.1.1.1). Besides automating data management, EMR systems also help to streamline the workflow in a medical setting. A typical EMR contains unstructured narrative text, structured coded data, and time stamped events. The basic components in an EMR system are shown in Figure 2.1. The patient EMR is thus an aggregation of data generated by each component and can be used for (1) administrative purposes, e.g: billing, reimbursement (2) diagnosis and prognosis by physicians (3) data mining and knowledge discovery by researchers.

Administrative Data Much of the data in the EMR serves administrative purposes. Administrative data is usually made up of socio-demographic information about the patient, medical reports and summaries. Every patient is registered with a unique identifier that links all patient data generated from different EMR components.

Clinical Data Clinical data could contain narrative text which includes treatment plans and patient summaries, results of clinical tests, medical images from radiology, medication and dosage instructions from pharmacy. The recording, storing and transmitting of data from each EMR component (as in Figure 2.1) is governed by defined standards. Some of the popular coding systems used to handle medical data are described below.

2.1.1.1 Coding systems

One of the primary reasons in adopting EMR is to facilitate and promote exchange of information among different healthcare settings. Seamless exchange of information requires coding standards. An interoperable EMR requires standards in four major areas (Reddy and Rahman, 2014): (1) Interaction with users (2) System communication (3) Information processing and management (4) Consumer device integration. Some popular coding standards implemented in most EMR are given below.

International Classification of Diseases The International Classification of Diseases[1] (often referred as ICD) is the official coding standard introduced by WHO (World Health Organization) to standardize disease and health related information exchange. It is a system of codes that covers diseases and related problems, social circumstances and external causes of injury or disease. The ICD system has gone through various revisions since its introduction. ICD-9 (ninth revision) is the most popular version released in 1978. The current revision is ICD-10, which was released by WHO in 1994. ICD-10 covers more diseases and diagnosis codes when compared to its predecessors and the coding scheme is more efficient (Reddy and Rahman, 2014). Australia has its own version of ICD-10 by adding country specific codes. The eleventh revision – ICD-11, is planned for 2018 (WHO).

CPT (Current Procedural Terminology) is a similar coding system maintained by American Medical Association. When compared to ICD, CPT describes treatment, whereas ICD is used to code diseases and symptoms.

[1] http://www.who.int/classifications/icd/en/

SNOMED-CT Systematized Nomenclature of Medicine – Clinical terms (SNOMED-CT[2]) was created by the College of American Pathologists (CAP) in 1965 (Cornet and de Keizer, 2008). It is a terminology system used to encode medical concepts using inbuilt definitions and formal logic and represent data for clinical purposes (Kostick, 2012).

LOINC Logical Observation Identifiers Names and Codes (LOINC) provides terminology to identify clinical results, such as laboratory tests, clinical observations, outcomes management and research. Each record represents a single test result and consists of the following fields: (1) Component measured (2) Component characteristics (3) Time of measurement (4) Specimen of the component (5) Measurement scale (6) Measurement Method (Bui and Taira, 2009).

RxNorm RxNorm[3] is a terminology standard developed by the United States National Library of Medicine (NLM) for representing medications. It includes medication name, dosage, route of administration, ingredients, pharmacy prescriptions (Nelson et al., 2011). Typical use of RxNorm include (1) using the standard nomenclature to capture/record drug information from EMRs (2) facilitating data exchange among providers (3) facilitating medication related CDSS (Nelson et al., 2011; Bennett, 2012).

Diagnosis Related Group Diagnosis Related Groups (DRG) are used to classify patients into predefined groups based on treatment data, and relate them to the costs incurred by the hospital (Averill et al., 1998). The groups were developed as a part of hospital reimbursement system, where a binding price could be attached to each group. It is used to define the reimbursement amount to the hospital from medical insurance systems like Medicare.

DICOM Digital Imaging and Communications in Medicine (DICOM) is a standard for handling and transmitting medical images. It defines the file format and network communication protocol for biomedical images (Bidgood et al., 1997).

[2]http://www.ihtsdo.org/snomed-ct/snomed-ct0/
[3]http://www.nlm.nih.gov/research/umls/rxnorm/

2.1.2 EMR data for Medical Informatics

As seen from the previous section, EMRs provide a high definition view of patient-provider interactions. Though the primary objective of EMRs are to record patient data, the granular detail of such recording can be leveraged for many secondary uses such as reducing healthcare costs and generating clinical insights. As stated in their whitepaper by American Medical Informatics Association (AMIA), such secondary use of healthcare data can enhance individual's health care experiences, expand knowledge about diseases and treatments, strengthen understanding of health care system's effectiveness and efficiency, support public health and security goals, and aid businesses in meeting customer's needs (Safran et al., 2007). To this end, EMR data has been successfully used to generate insights and predicting events in administrative, clinical and industrial applications. We detail some of the most popular application areas below:

Understanding Diseases: EMR data can be used to investigate prevalence or incidence of a disease. For example, Jensen et al. (2014) used EMR data to investigate disease progression pattern in Denmark. Patient medical trajectories, such as functional impairments in terminal patients, can also be modelled using EMR data (Teno et al., 2001; Murtagh et al., 2008). Such data can also be used in comorbidity analysis, which is the process of understanding the relationship between frequently co-occurring diseases. Researchers have used comorbidity analysis to study patients with personality disorders (Roque et al., 2011), autism spectrum disorders (Doshi-Velez et al., 2014), hypertension (Shin et al., 2010), and rare diseases (Holmes et al., 2011; Cao et al., 2005).

Cohort Identification: This involves identifying patient groups that satisfy a given criteria. Identifying specific cohorts is an important process in clinical research studies and various biomedical applications. The diagnosis codes and narrative text in EMR database have been used to develop automated models to identify patients with cancer (Xu et al., 2011; Whyte et al., 2015), rheumatoid arthritis (Liao et al., 2010), critical care (Halpern et al., 2014) and asthma (Meystre et al., 2009).

Biomarker Discovery: A biomarker is a measurable indicator of presence or severity of a given disease state. The presence or value of biomarker can be observed to

indicate disease or health state. For example, body measurements as weight, body mass index (BMI), and waist-to-hip ratio are used to identify obesity or metabolic disorders. Similarly, abnormal haemoglobin A1C is a biomarker for type-2 diabetes, whereas hyperlipidemia is a biomarker for cardio-vascular diseases. Since these information is recorded in patient records, statistical techniques can be used to identity the few important indicators among thousands of EMR variables. For example, rule mining techniques have been used in identifying biomarkers for diabetes (Schrom et al., 2013; Simon et al., 2013). A recent study employed Bayesian Non-parametric factor analysis identity biomarkers for autism spectrum disorder (Vellanki et al., 2014).

Predicting Future Complications: A challenging application of EMR data is predicting short term and long term complications in patients: for example onset of a related disease, re-hospitalization or exacerbation of a condition. EMR databases contain large patient cohorts over longer observation periods, making it possible for clinical prediction systems to study and model patient complications over time. Generalized linear models and techniques such as Cox regression are popular choices for such applications. Recently, Yadav et al. (2015a) used Cox proportional hazards model to estimate the risk of complications arising due to type-2 diabetes such as Peripheral Vascular Disease (PVD), Cerebral Vascular Disease (CVD), Ischemic Heart Disease (IHD) and Congestive Heart Failure (CHF). Another study used temporal feature extraction from patient records and ordinal classifiers to classify mental health patients with high risk of suicide (Tran et al., 2013), and demonstrated superior performance over traditional clinician based assessments. The availability of EMR data has made it possible to predict future complications of many medical events such as predicting cancer (Algar et al., 2003; Zhao and Weng, 2011; Gupta et al., 2014; Klastersky et al., 2006), patient readmissions (Amarasingham et al., 2010; Krumholz et al., 1997; Kansagara et al., 2011; Gopakumar et al., 2015b), surgery complications (Propst et al., 2000; SooHoo et al., 2006; Ozkalkanli et al., 2009), to name a few.

Quantifying effect of Interventions: Interventions are undertaken to treat or cure a medical condition. The most common example of medical intervention is prescription of medications. The ability to quantify the effect of an intervention builds the platform for sophisticated and personalised treatment strategies. The longitudinal nature of EMR data provides an excellent opportunity to investigate the effect of interventions

in patients. As example, the effect of medications such as the statin inhibitor drug has been studied for mitigating diabetes (Schrom et al., 2013) and incidence of ischemic heart disease (IHD) events and stroke (Law et al., 2003). Life style modifications such as smoking cessations and low-calorie diet can also become interventions. Prochaska et al. (2008) studied 8000 participants in four multi-behavioral interventions: smoking, alcohol abuse, physical inactivity, and poor diet and analysed their effect on human health using five methods.

Detecting Adverse Events: Adverse events could arise due to drug reactions, incorrect practices or use of outdated guidelines. Yadav et al. (2015b) categorize such research into pharmacovigilance and patient monitoring. Pharmacovigilance is related to monitoring adverse effects of medications and drugs. A combination of data from patient records and supervised learning techniques have been used to detect adverse drug reactions and allergies (Vilar et al., 2012; Harpaz et al., 2013; Epstein et al., 2013).

Patient monitoring using diverse information helps clinicians understand the causal factors for adverse events. Such monitoring uses a variety of real-time data stored into the EMR such as biomarkers, physiological variables and behavioural data. For example, Rose et al. (2005) developed a dynamic Bayesian network to monitor the dry weight of patients suffering from renal failure and treated by hemodialysis. Bayesian networks were also used to identify fluctuating levels of serum glucose in critically ill patients (Nachimuthu et al., 2010).

Predicting Risk: In clinical setting, risk can be defined as adverse outcome of a diagnostic/therapeutic procedure or worsening of the patient's current medical state. Risk assessment in patients helps to classify patients into care groups and design treatment plans (Ng et al., 2014; Tran et al., 2015b). Besides modelling disease exacerbation, risk assessment can also be used to identify the underlying contributing factors. Study of these risk factors is important to clinicians, as such factors are subjected to further analysis to understand prognosis (Gopakumar et al., 2015b).

Machine learning models as logistic regression, Poisson regression and survival modelling as Cox proportional hazards regression are quite popular for predicting patient risk such as disease exacerbation, mortality and re-hospitalization.

In this thesis, we focus on modelling patient readmission. One of the early works of using medical records to predict re-hospitalization due to heart failure was done by Chin and Goldman (Chin et al., 1997). A total of 257 patient medical records from a hospital in Boston, Massachusetts was used to develop 11 point scoring system using Cox proportional hazards regression modelling, from 25 candidate variables to derive a risk score for death or all cause readmission to any hospital within 60 days. Though the model identified several independent risk factors, the work did not report any AUC. A more extensive study was conducted by Philbin and DiSalvo (Philbin and DiSalvo, 1999) using administrative data from Statewide Planning and Research Cooperative System (SPARCS) consisting of 42,731 patients from 236 hospitals. Multivariate logistic regression analysis was performed to derive a 15-point scoring system from 60 candidate variables to predict HF specific re-hospitalization within one year. The model reported an AUC of 0.60. A similar work was performed by Krumholz et al. (2000) that used 2,176 patient medical records from 18 hospitals to derive a 32-variable model to predict HF specific re-hospitalization within 6 months. From the given data, the model identified four independent predictors for re-hospitalization, but did not report any AUC. Felker et al. (2004) examined 41 candidate variables from 949 patient medical records in 78 hospitals to come up with a statistical model predicting death or all-cause readmission within 60 days. The AUC was reported as 0.69. Yamokoski et al. (2007) came up with a model to predict all-cause readmission to any hospital within 6 months and compared the performance with clinical judgements from nurses and physicians. The model was developed using 18 candidate variables from 373 patient records in 26 hospitals and reported an AUC of 0.60. Recently, Amarasingham et al. (2010) combined non clinical data along with clinical data from 1,372 patient records to predict death or all-cause readmission to any hospital within 30 days.

Detailed comparison studies among the existing models for predicting HF specific re-hospitalization confirm that there were no consistent predictors (Ross et al., 2008; Betihavas et al., 2012). The predictors that were common to more than one model were: history of diabetes mellitus, elevated blood urea and nitrogen, history of prior admission to hospital within one year, single marital status and race.

2.1.3 EMR Modelling

The first step in building learning models from EMR data is to derive a good feature set that reflects the temporal nature of patient data. Patient records in EMR database has a longitudinal structure which contains various information such as patient demographics, administrative details, clinical observations and results that are recorded over time (Wang et al., 2012a). Figure 2.2 gives an example for one such record for a diabetic patient. Wang et al. (2012a) and Tran et al. (2013) identified several challenges when modelling the temporal data such as:

1. **Shift-invariance:** The absolute time-points in the records become irrelevant since all patients are not temporally aligned. Also due to comorbidities, different patients follow different trajectories over longer time periods.
2. **Heterogeneity**: The data recorded consists of mixed types. For example, blood pressure is continuous whereas age is discrete. Events like birth are recorded once, but some events as heartbeats are recorded continuously. Also different events progress at different rates.
3. **Sparsity and Irregularity:** Not all events need to be recorded for all patients. Newer and healthier patients will contain lesser data than others. Medical events occur sporadically and the data recording reflects this irregularity.
4. **Quantitative Nature:** Data extraction process must be able to identify the importance of order of occurrence of clinical events and identify measures such as event duration and interval between events.
5. **Scalability:** Data extraction process must be able to handle large cohort of patients with potentially long records (for e.g: 5 years, 10 years) with different types of features.
6. **Distribution drifts:** The introduction of new procedures, policies and treatment guidelines will introduce drifts in event distribution. The data modelling algorithms should be able to accommodate these changes.

Existing research tackles feature extraction from EMR in different ways. A popular method is temporal abstraction of clinical data, where a set of time-stamped parameters (measurements, events, goals) are converted into higher level abstractions relevant

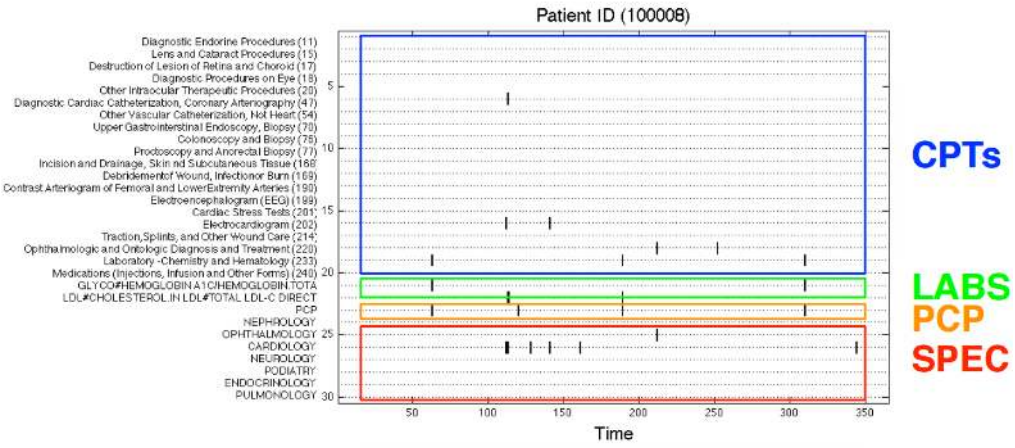


Figure 2.2: An example of time-indexed EMR record of a diabetic patient for one year adapted from (Wang et al., 2012a). The X-axis represents day 1 to day 365. The Y-axis represents clinical events categorized into 4 groups: procedures (CPTs), lab results (LABs), visits to primary care physician (PCP) and visits to specialists (SPEC). Dots in the graph represent presence of the corresponding event.

for clinical decision making (Shahar and Musen, 1996). An early work built a system named RESUME that used ontologies from medical domain along with temporal events from patient database to create abstractions based on inference and interpolation (Shahar and Musen, 1992, 1996). Sacchi et al. (2007) proposed a knowledge driven approach using variations of temporal association rule and an Apriori-like technique that extracts frequent occurrences of precedence between episodes. The approach was testing on two different biomedical datasets. Batal et al. (2013) proposed a Minimal Predictive Temporal Patterns framework to extract predictive features from patients with blood disorder. Schmidt and Gierl (2005) added case based reasoning to temporal abstraction for prognosis of kidney function and spread of influenza.

Variations of deep learning techniques have been proposed for modelling irregularity in event times and interventions. Lasko et al. (2013) pioneered a deep learning approach for clinical phenotype discovery from irregular and sparse EMR data. This data driven method resulted in discovering phenotypes that were accurate as those engineered by a domain expert. Pham et al. (2016) demonstrated superior performance of a Long short-term memory (LSTM) network using time parametrizations over standard classifiers that use non-temporal features.

Following the success of data representation in text and multimedia learning (Bengio

et al., 2013), deep learning approaches have also been applied for feature extraction and representation of patient data from EMRs. A prominent work introduced "Deep Patient", a model to capture patterns in patient data using stacked autoencoders (Miotto et al., 2016). The study utilized 700,000 patients and demonstrated superior prediction performance on 78 diseases including cancer, diabetes and schizophrenia. Choi et al. (2016) introduced Med2Vec based on the popular Skipgram model (Mikolov et al., 2013) to learn distributed representations from 3 million patient visits to provide clinically meaningful interpretations. Another recent work used restricted Boltzmann machine (RBM) to embed patient features from EMR into a vector space and demonstrated superior performance when compared to clinicians for predicting suicide risk (Tran et al., 2015a). Variations of RBM have also been used for patient profiling with applications in studying disease correlation and risk prediction (Nguyen et al., 2013). Along the similar lines, a deep learning architecture for clinical prediction - Deepr, uses convolutional neural nets (CNN) to detect motifs from words and phrases in medical records and is used to predict future risk in patients (Nguyen et al., 2016).

In our work we use feature engineering from EMR data along lines of recent work in Tran et al. (2014). Here, patient records are considered as a sparse temporal image and are subjected to a one-sided filter bank resulting in aggregations over multiple time periods and granularities. Apart from the applications listed in Section 2.1.2, one of the most popular uses of EMR data is in modelling patient flow.

2.1.4 Patient Flow Analysis

Patient length of stay directly contributes to hospital costs and resource allocation. Long-term forecasting in health care aims to model bed and staffing needs over a period of months to years. Such forecasts are typically made with the help of administrative and clinical data in EMRs. Cote and Tucker categorized the common methods in health care demand forecasting as percent adjustment, 12-month moving average, trend line, and seasonalized forecast (Cote and Tucker, 2001). Although each of these methods is built from historical demand, seasonalized forecasting provides more realistic results as it takes into account the seasonal variations and trends in the data. Mackay and Lee (2005) advise modelling the patient flow in health care institutions for tactical and strategic forecasting. To this end, compartmental modelling (McClean and Millard,

1995, 1998), queuing models (El-Darzi et al., 1998; Mills, 2004) and simulation models (Mills, 2004; Costa et al., 2003; El-Darzi et al., 1998; Hoot et al., 2008) have been applied to analyse patient flow. To understand long-term patient flow, studies analyse metrics such as bed occupancy (Mackay and Lee, 2005; Harper and Shahani, 2002; McClean and Millard, 1995; El-Darzi et al., 1998; Mackay, 2001; Gorunescu et al., 2002), patient arrivals (Peck et al., 2012), and individual patient length of stay (El-Darzi et al., 1998; Barnes et al., 2016; Levin et al., 2012; Clark and Ryan, 2002; Marshall et al., 2005). The most popular unit of interest is the emergency or acute care department because this is often a key performance indicator metric in assessing quality of care (Kulinskaya et al., 2005; Lindsay et al., 2002). The available techniques for patient flow forecasting can be broadly categorized into time series and smoothing methods, simulation methods and regression methods.

2.1.4.1 Time Series and Smoothing Methods

When looking at discharges as time series, autoregressive moving average models are the most popular (Jones et al., 2002; Littig and Isken, 2007; Lin et al., 2011). Exponential smoothing techniques have also been used to forecast monthly (Lin, 1989) and daily patient flow (Jones et al., 2008). Jones et al. (2002) used the classical ARIMA to forecast daily bed occupancy in emergency department of a European hospital . The model which included seasonality terms demonstrated reasonable performance to predict bed occupancy. The authors speculated whether non-linear forecasting techniques could improve over ARIMA. A recent study confirmed the effectiveness of this forecasting technique in a US hospital setting (Schweigler et al., 2009). ARIMA models were also successfully used to forecast the number of occupied beds during a severe acute respiratory syndrome (SARS) outbreak in a Singapore hospital (Earnest et al., 2005). A recent study used patient attendances in a paediatric emergency department to model daily demand using ARIMA (Kadri et al., 2014). Jones et al. (2008) compared the ARIMA mode with exponential smoothing and artificial neural networks to forecast daily patient volumes in emergency department. The study revealed no single model to be superior and concluded that seasonal patterns play a major role in daily demand.

2.1.4.2 Simulation Methods

Modelling using simulation is typically used to study the behaviour of complex systems. An early work investigated the effects of emergency admissions on daily bed requirements in acute care, using discrete-event stochastic simulation modelling (Bagust et al., 1999). Sinreich and Marmor (2005) proposed a guide for building a simulation tool based on data from emergency departments of 5 Israeli hospitals. Their method analysed the flow of patients clustered into 8 types along with time elements. The simulation demonstrated that patient processes are better characterized by type of the patients, rather than specific hospitals visited. Yeh and Lin (2007) used a simulation model to characterize patient flow through a hospital emergency department and reduced waiting times using a genetic algorithm . A similar experiment was carried out in a geriatric department using a combination of discrete event simulation and queuing model to analyse bed occupancy (El-Darzi et al., 1998).

2.1.4.3 Regression for Forecasting

Regression models analyse the relationship between the forecasted variable and features in the data. Linear regression that encoded monthly variations was used to forecast patient admissions over a 6-month horizon and outperformed quadratic and autoregressive models (Boyle et al., 2008). Another study used clustering and Principle Component Analysis (PCA) to find significant predictors from patient data to model emergency length of stay using linear regression (Combes et al., 2014). A non-linear approach using regression trees was proposed in forecasting patient admissions which demonstrated superior performance over a neural net framework (Garcia and Chan, 2012). Barnes et al. (2016) used 10 predictors to model real-time inpatient length of stay in a 36-bed unit using a random forest (RF) model. Non-linear regression is better suited to model the changing dynamics of patient flow. In the area of pattern recognition, k-nearest neighbours (kNN) (Cover and Hart, 1967) are the most effective method that exploits repeated patterns. The non-parametric regression using kNN has been successfully applied in many forecasting applications, for example forecasting time series in financial data (Arroyo and Maté, 2009), short-term traffic forecasting (Davis and Nihan, 1991; Zhang et al., 2013) and electricity load forecasting (Al-Qahtani and Crone, 2013; Tsakoumis et al., 2002). However, kNN regression has not been studied

for patient flow. Another powerful and popular regression technique, support vector regression (SVR), uses kernel functions to map features into a higher dimensional space to perform linear regression. Though this technique has not seen much application in medical forecasting, support vector machines have been successful in financial market prediction, electricity forecasting, business forecasting, and reliability forecasting (Sapankevych and Sankar, 2009). RF and SVR regression are powerful modelling techniques requiring minimum tuning to effectively handle non-linearity in the hospital processes. Recently, RF forecasting was used to predict total patient discharges from a 36 bed unit in an urban hospital (Barnes et al., 2016). Apart from 4 demographic and 2 timing predictors, this study used 3 clinical predictors for patients: (1) reason for visit: identified by a physician and recorded using International Classification of Diseases: version 9 (ICD-9) diagnosis codes[4], (2) observation status: assigned to patients for monitoring purpose, and (3) pending discharge location. Total number of discharges was estimated from aggregate of individual patient length of stay. The absence of real-time clinical information in our data makes calculating patient length of stay impossible.

2.1.5 Clinical Prediction Models

With the emergence and wide adoption of Electronic Medical Records, it has become possible to bridge the inferential gap between cohort characteristics and individual patient detail. The EMRs have become a potential gold mine for data mining researchers. Clinical prediction models provide evidence based input that facilitate shared decision making involving both doctors and patients (Steyerberg, 2009). Clinical prediction models can be used to identify strong predictors from large and noisy databanks, for example: investigating the effect of C-reactive protein on acute coronary syndrome (Van de Werf et al., 2008). They can also be used to provide absolute risk estimates for individual patients (Harrell et al., 1996; Altman et al., 2009). The model gives an estimate of risk as a function of different variables which may involve patient characteristics, disease characteristics and treatment characteristics. Here, risk refers to unwanted outcome (mortality, readmission, exacerbation). Table 2.1 lists the different categories of outcome encountered in clinical prediction.

[4]http://www.cdc.gov/nchs/icd/icd9cm.htm

Outcome	Prediction Model	Examples
Continuous	Linear Regression, Generalized Additive Models	medical costs prediction
Binary	Logistic Regression, Binary classification trees, Bayesian models	disease diagnosis, mortality prediction, medical image segmentation
Categorical	Ensemble approaches: Multiclass prediction, Multinomial logistic regression, Maximum Entropy	Classification of cancer, tumour
Ordinal	Ordinal Regression	Grading severity of illness
Time-to-event	Survival analysis, Cox Model	Time to death or re-hospitalization

Table 2.1: Outcomes in clinical prediction

Statistical models for prediction can be broadly categorized into regression models, classification models and neural networks (Hastie et al., 2001b). We briefly describe the popular models in clinical prediction below. But first we begin by introducing the notations for prediction models used in this thesis.

Notations Unless otherwise specified, throughout this thesis, we will use M to represent the number of data points (instances or observations) and N to denote the number of features in the data. Vectors are denoted in bold lower case and the i^{th} component of vector $\boldsymbol{x}$ is denoted as: x_i. The parameter of the learning model is denoted as $\boldsymbol{w}$. We use $X \in \mathbb{R}^{M \times N}$ to denote the input data containing M examples with N features. The target vector of labels is denoted as $\boldsymbol{y} \in \mathbb{R}^M$.

2.1.5.1 Linear Regression

Regression analysis involves predicting the value of a continuous target variable for a given value of input. Given M training examples as $X = x_1, x_{2,} \cdots, x_M$, a simple linear regression models the target vector $\boldsymbol{y}$ as:

$$\begin{aligned} \boldsymbol{y}(X, \boldsymbol{w}) &= w_o + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M \\ &= \sum_{i=0}^{M} w_i x_i \end{aligned} \tag{2.1}$$

where the coefficient of w_0 (intercept or bias term) becomes $x_o = 1$. At its heart, a two dimensional regression problem becomes the task of fitting a single line through a scatter plot of target variables. The best line is characterized by the optimum value of $\boldsymbol{w}$ that reduces the error in prediction. The most popular method to estimate coefficient $\boldsymbol{w}$ is using residual sum of squares (RSS) estimate as:

$$\text{RSS}(\boldsymbol{w}) = \sum_{i=1}^{M} (y_i - w_i x_i)^2 \tag{2.2}$$

An intuitive representation of (2.2) is illustrated in Figure 2.3. Since RSS($\boldsymbol{w}$) is a quadratic function, the minimum exists, though not necessarily unique. In general, when we have M data points with N features, we can rewrite (2.2) in matrix notation as:

$$\text{RSS}(\boldsymbol{w}) = (y - X\boldsymbol{w})^T (y - X\boldsymbol{w})$$

Assuming X is a full rank matrix, the unique solution becomes:

$$\boldsymbol{w} = (X^T X)^{-1} X^T \boldsymbol{y}$$

We can extend the class of models in (2.1) by modelling the target vector $\boldsymbol{y}$ as a linear combination of non-linear functions $\phi(x_i)$, $i \in (1, M)$. By using such functions as polynomial, sigmoid and $tanh$ functions, we can capture non-linearity in data (Bishop, 2006).

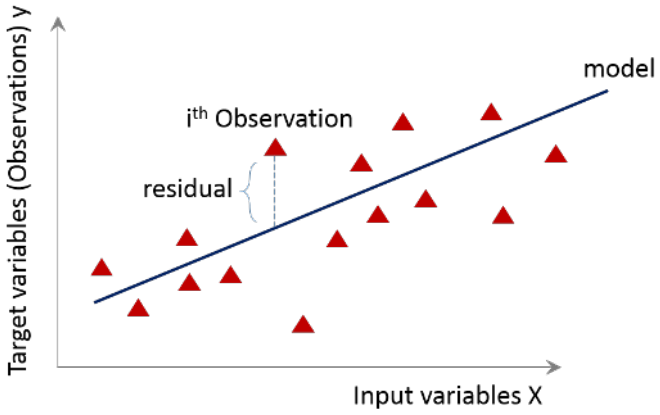


Figure 2.3: Geometric interpretation of least squares regression in two dimensions.

2.1.5.2 Logistic Regression

Logistic regression is a type of regression analysis used for predicting the outcome of a categorical dependent variable based on one or more predictor values (Peng et al., 2002). When compared to linear regression, logistic regression uses our input variables $X = x_1, x_2, \cdots, x_M$ and parameter $\boldsymbol{w}$ to predict a binary valued output $y \in \{0, 1\}$. Hence we require the hypothesis function for logistic regression, $h_{\boldsymbol{w}}(X)$, to be between 0 and 1. Formulating $h_{\boldsymbol{w}}(X)$ as a sigmoid function ensures that $0 \leq h_{\boldsymbol{w}}(X) \leq 1$. A sigmoid function has the form as shown in Figure 2.4. The sigmoid function is also known as the *logistic* or *logit* function and has the form: $g(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{e^z+1}$. The function asymptotes at 1 when z approaches infinity and asymptotes at 0 when z approaches negative infinity. If we take $z = \boldsymbol{w}^T X$, we have:

$$g(z) = g(\boldsymbol{w}^T X) = \frac{1}{1 + e^{-\boldsymbol{w}^T X}} \quad (2.3)$$

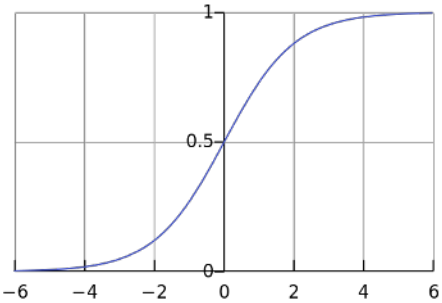


Figure 2.4: Sigmoid function. The X-axis represents z and Y-axis corresponds to $g(z)$

Hence our hypothesis function becomes:

$$h_{\boldsymbol{w}}(X) = \frac{1}{1 + e^{-\boldsymbol{w}^T X}}$$

For $\boldsymbol{y} \in \{0, 1\}$ and a given X, let us assume the following

$$P(y = 1|\, X;\, \boldsymbol{w}) = h_{\boldsymbol{w}}(X)$$
$$P(y = 0|\, X;\, \boldsymbol{w}) = 1 - h_{\boldsymbol{w}}(X)$$

The conditional probability $P(\boldsymbol{y}|X;\, \boldsymbol{w})$ is thus a Bernoulli variable, which can be written as:

$$P(\boldsymbol{y}|X;\, \boldsymbol{w}) = (h_{\boldsymbol{w}}(X))^{y}\,(1 - h_{\boldsymbol{w}}(X))^{1-y}$$

Now if we consider that M training examples were generated independently, the likelihood of the parameters become:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}) &= \prod_{i=1}^{M} p(y^{(i)}|\, x^{(i)},\, \boldsymbol{w}) \\ &= \prod_{i=1}^{M} \left(\frac{1}{1 + e^{-\boldsymbol{w}^T X}}\right)^{y^{(i)}} \left(1 - \frac{1}{1 + e^{-\boldsymbol{w}^T X}}\right)^{1-y^{(i)}} \end{aligned} \tag{2.4}$$

Here, (2.4) represents a likelihood. The cost function or the risk function that we need to optimize is obtained by taking the negative logarithm of this likelihood. The cost function for logistic regression: $J(\boldsymbol{w})$ becomes:

$$\begin{aligned} J(\boldsymbol{w}) &= -\log \prod_{i=1}^{M} \left(\left(\frac{1}{1 + e^{-\boldsymbol{w}^T X}}\right)^{y^{(i)}} \left(1 - \frac{1}{1 + e^{-\boldsymbol{w}^T X}}\right)^{1-y^{(i)}} \right) \\ &= \sum_{i=1}^{M} - y^{(i)} \log h_{\boldsymbol{w}}(x^{(i)}) - (1 - y^{(i)}) \log\left(1 - h_{\boldsymbol{w}}(x^{(i)})\right) \end{aligned} \tag{2.5}$$

Further, (2.5) represents $J(\boldsymbol{w})$ in terms of entropy between y_i and $h_{\boldsymbol{w}}(x_i)$, and hence is also called cross-entropy. Model learning intuitively implies to minimize the cost function and minimize the cross entropy. Parameter estimation is detailed in Appendix section A.1.

2.1.5.3 Survival Analysis and Cox Regression

Survival analysis is a collection of statistical procedures dealing with analysis of time until one or more events occur. The outcome variable of interest is *time until an event occurs* (Kleinbaum and Klein, 2006) and is often referred to as failure time, survival time or event time. Common examples are: time until tumour recurrence, time until

next heart attack after some treatment intervention, time until chronic obstructive pulmonary disease (COPD) exacerbation.

Modelling time to event data for a cohort introduces the following challenges. First, event time will be different for different patients. At the end of study, chances are that the event might not have occurred for some patients. Hence time interval is not normally distributed. Second, some patients may drop out of the study due to various reasons. Such patients are marked as censored observations. The patients in whom the event has not yet occurred by the end of the study are also treated as censored. Finally, the explanatory variables in some cases can also be time-varying. Hence, conventional statistical methods like linear or logistic regression cannot be applied to such data.

The following assumptions are made in all survival studies. Patients are recruited over a period and followed up to a fixed date. Survival prospects stay the same throughout the study. Censored patients have the same prognosis as the others. Finally, the probability of an individual patient to be censored is unrelated to the probability of suffering the endpoint event.

Survival Time: Survival time data measures time to a certain event (death, response, relapse, development of a disease). Survival time is a random variable and its distribution is characterised by a survivorship function (Survival function), a probability density function and a hazard function.

Survival Function If T denotes the survival time, the survival function S(t) can be defined as: S(t) = P(an individual survives longer than t). Mathematically, we can express this as: $S(t) = P(T > t)$. In other words: S(t) = 1 - P(an individual fails before t). Hence $S(t) = 1 - F(t)$, where $F(t)$ is the cumulative distribution function. A typical survival function looks like in Figure 2.5. Survival function will have the following properties:

1. It is non increasing
2. At $t = 0$, $S(t) = 1$. Probability of surviving past time 0 is 1
3. At $t = \infty$, $S(t) = S(\infty) = 0$. Probability of surviving past infinite time is 0.

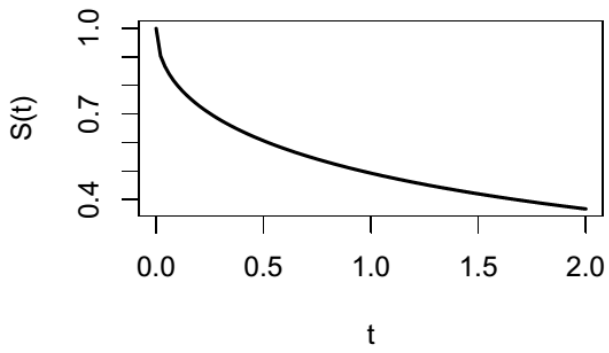


Figure 2.5: Survival curve: graph of $S(t)$ with t

If there are no censored observations, the estimate of survivor function $\widehat{S}(t)$, is calculated as:

$$\widehat{S}(t) = \frac{\text{no of patients surviving longer than } t}{\text{total no of patients}}$$

This estimate does not hold true when censored observations are present and we resort to non-parametric methods.

Probability Density Function (Density function) The probability density function of survival time T can be expressed as:

$$f(t) = \lim_{\triangle t \to 0} \frac{P\left(\text{an induvidual dying in the interval } (t + \triangle t)\right)}{\triangle t}$$

If there are no censored observations, the estimate of the density function $\widehat{f}(t)$, is calculated as:

$$\widehat{f}(t) = \frac{(\text{no of patients dying in interval begining at } t)}{(\text{total no of patients}) \times (\text{interval width})}$$

This estimate does not hold true when censored observations are present.

Hazard Function In simple words, the hazard function $h(t)$ gives the probability of succumbing to the event at a particular instant $(t + \triangle t)$, given that you have survived up to the instant t. Mathematically, we can express this as:

$$h(t) = \lim_{\triangle t \to 0} \frac{P\left(\text{an induvidual fails in the interval } (t + \triangle t) \,|\, \text{induvidual survived to } t\right)}{\triangle t}$$

When there are no censored observations, hazard function estimate can be calculated as:

$$\begin{aligned}\widehat{h}(t) &= \frac{\text{no of patients dying in interval begining at } t}{(\text{no of patients surviving at } t) \times (\text{interval width})} \\ &= \frac{\text{no of patients dying in interval begining at } t}{\frac{(\text{no of patients surviving at t}) - (\text{no of deaths at t})}{2} \times (\text{interval width})}\end{aligned}$$

Cox Regression Survival models can be viewed as consisting of two parts: an underlying hazard function $h_0(t)$ and the effect of predictor variables (covariates). The underlying hazard function $h_0(t)$ models the risk with time at baseline levels of the predictor variables. When the exact form of the underlying survival function is unknown (as in many real-life scenarios), we resort to non-parametric methods such as the Cox proportional hazards model (Lee and Wang, 2013).

Cox proportional hazards model has the property that hazard ratio of any two individuals is a constant. In other words, the ratio of hazard functions for two individuals $\frac{h(t|x_1)}{h(t|x_2)}$ does not vary with time t. If the effect of covariates is characterised by the function $g(x)$, we can write the hazard ratio as:

$$\frac{h(t|x_1)}{h(t|x_2)} = \frac{h_0(t) \times g(x_1)}{h_0(t) \times g(x_2)} = \frac{g(x_1)}{g(x_2)} \tag{2.6}$$

which is a constant, independent of time.

We now deviate from our standard notations to express the Cox model mathematically. The Cox proportional hazards model can be written as:

$$h(t|x) = h_0(t|x) \exp(\beta^T x) \tag{2.7}$$

where x represents a $p \times 1$ vector of covariates (predictors) such as treatment indicators and prognostic factors, while β represents a $p \times 1$ vector of regression coefficients. There is no intercept β_0 for the model in (2.7). In the absence of predictor variables, we have:

$$h(t|x = 0) = h_0(t)$$

Here $h_0(t)$ is called the baseline hazard function. It can be interpreted as the hazard

function for the population in the absence of predictor variables ($x = 0$). This baseline hazard function can take any shape as a function of time. The only requirement is $h_0(t) > 0$. Hence, $h_0(t)$ is non-parametric while $\exp(\beta^T x)$ is parametric, making the cox model, characterised by (2.7), a semi-parametric model.

Interpretation of a proportional hazards model We use a simple example to illustrate the principles of Cox regression. Consider we are modelling the diagnosis of lung cancer in a cohort of smokers and non-smokers. To isolate the effect of smoking on lung cancer, we model the hazard function – probability that a patient is diagnosed with lung cancer at time t, denoted as $h(t)$, using a single predictor x_{smoke}, where $x_{\text{smoke}=1}$ indicates patient is a smoker. Using Cox regression with parameter β, the hazard function for lung cancer in an individual with a single predictor x_{smoke} can be written as:

$$h(t|x_{\text{smoke}=1}) = h_0(t|x) \exp(\beta^T x_{\text{smoke}=1})$$

Here, the baseline or underlying hazard function $h_0(t|x)$ corresponds to probability of having lung cancer when all explanatory variables are zero. Hence, $h_0(t|x) = h(t|x_{\text{smoke}=0})$. The hazard ratio of smoker to non smoker ($x_{\text{smoke}=1}$ and $x_{\text{smoke}=0}$) becomes:

$$\frac{h(t|x_{\text{smoke}=1})}{h(t|x_{\text{smoke}=0})} = \exp(\beta)$$

To understand the effect of smoking on lung cancer, we just need to estimate β. If $\beta = 0$, smoking has no effect on lung cancer. If $\beta < 0$, then smoking reduces the hazard of lung cancer. If $\beta > 0$, smoking increases the risk of lung cancer.

We can convert the Cox proportional hazard model into a regression model by rewriting (2.7) as:

$$\frac{h(t|x)}{h_0(t|x)} = \exp(\beta^T x)$$

Taking natural logarithm on both sides:

$$\ln[\frac{h(t|x)}{h_0(t|x)}] = \beta^T x \tag{2.8}$$

Hence in Cox regression, the linear combination of predictors represent the hazard ratio. The parameter estimation of Cox model is detailed in Appendix section A.2.

Relationship of Survival Function and Hazard Function Given the probability density function $f(t)$, with cumulative distribution function $F(t)$, the survival function becomes:

$$\begin{aligned} S(t) &= 1 - F(t) \\ \implies F(t) &= 1 - S(t) \end{aligned}$$

The hazard function can be written as:

$$h(t) = \frac{f(t)}{1 - F(t)}$$

Hence, the relation between probability density function $f(t)$, hazard function $h(t)$ and survival function $S(t)$ can be written as:

$$h(t) = \frac{f(t)}{S(t)} \tag{2.9}$$

Since the probability density function is the derivative of the cumulative distribution function, we have:

$$\begin{aligned} f(t) &= \frac{d}{dt}F(t) \\ &= \frac{d}{dt}[1 - S(t)] \\ &= -S'(t) \end{aligned} \tag{2.10}$$

Substituting (2.10) in (2.9), we have:

$$\begin{aligned} h(t) &= \frac{-S'(t)}{S(t)} \\ &= -\frac{d}{dt}\log S(t) \end{aligned}$$

We can rewrite this as:

$$\begin{aligned}
\log S(t) &= -\int_0^t h(x)d(x) \\
S(t) &= \exp\left[-\int_0^t h(x)d(x)\right] \\
&= \exp\left[-\int_0^t h_0(x)\exp(\beta^T x)\, d(x)\right] \\
&= \exp\left[-\int_0^t h_0(x)d(x)\right]^{\exp(\beta^T x)} \\
&= S_0(t)^{\exp(\beta^T x)}
\end{aligned}$$

Hence, we can express the survival function $S(t)$ in terms of a baseline survival function $S_0(t)$ as:

$$S(t) = S_o(t)^{\exp(\beta^T x)} \tag{2.11}$$

2.1.5.4 Generalized Linear Models

In linear regression, we modelled the relationship between input variables X and observed variables $\boldsymbol{y}$ using a linear combination $\boldsymbol{y} = \boldsymbol{w}^T X$. For logistic regression, the linear combination of input variables becomes the log of odds: $\boldsymbol{w}^T X = \log\left[\frac{P(y=1|\,x)}{1-P(y=1|\,x)}\right]$. In the case of Cox regression, linear combination on explanatory variables models the hazard ratio. Such relationships can be expressed using a generalized linear model formulation.

A generalized linear model, as the name suggests is a generalization of ordinary linear regression modelling to include different relationships between observed and explanatory variables under certain conditions. Ordinary linear regression, in two dimensions, is a line (as in Figure 2.3). We focus on three characteristics: (1) the distribution of observed variable (2) the function of explanatory variable, and (3) the connection between explanatory variable and distribution of observed variable. For ordinary linear regression, observed variable follows a normal distribution: $\boldsymbol{y} \sim N(\mu, \sigma^2)$. Explanatory variables are modelled using the function: $\boldsymbol{w}^T X$. The linear model in this case repre-

sents mean of observed variable: $\mu = E(\boldsymbol{w}) = \boldsymbol{w}^T X$. The generalized linear model makes it possible to model many relationships by allowing more flexibility in these three characteristics.

Generalized linear models can have any member of the exponential family as the observed variable: $\boldsymbol{y}$. The function of explanatory variable is linear in parameters, and can have more than one explanatory variable, making it possible to have functions such as: $\boldsymbol{w}^T X + \boldsymbol{v}^T Z$ and $\boldsymbol{w}^T X + \boldsymbol{v}^T X^2 + \boldsymbol{u}^T Z$. Finally, a link function connects the mean $\mu = E(\boldsymbol{y})$ to the function of explanatory variable. In case of ordinary linear regression, the link function η is the identity. So we have $\eta = \mu = \boldsymbol{w}^T X$. We can now have a variety of link functions depending on the type of observed variable $\boldsymbol{y}$. For example, if we are modelling the count of patients discharged from a ward, μ can only take positive values. The observed variables $\boldsymbol{y}$ follows a Poisson distribution and we use the link function: $\eta = \log\mu$. Table 2.2 lists some common distributions and their link functions. In machine learning literature, the inverse of link function is referred to as the activation function (Bishop, 2006). Using the activation function $f(.)$, a generalized linear model becomes:

$$\boldsymbol{y} = f\left(\boldsymbol{w}^T X\right)$$

Here, the decision surface is linear, in spite of non-linearity in activation function. Parameter estimation can be done using maximum likelihood estimation (Bishop, 2006).

2.1.5.5 Evaluating Prediction Models

There are many methods to evaluate a prediction model. The most common approach will be to quantify the distance between predictions and actual outcome. Popular measures are explained variation (R^2 statistics) and the Brier score.

Brier Score Brier score, invented by Glenn W. Brier, gives an estimate of predictive performance of the model (Brier, 1950). More formally, Brier score measures the mean squared difference between true outcomes and predicted outcomes. For a sample of N observations where y_i, $\hat{y}_i$ are the i^{th}true outcome and predicted outcome, we calculate

Distribution	support	Clinical example	Link function	Mean function
Normal	real: $(\text{-}\infty, +\infty)$	measurement of blood pressure, weight, height	$\eta = \mu$	$\mu = \boldsymbol{w}^T X$
Exponential	real: $(0, +\infty)$	sample size estimation, survival analysis	$\eta = \mu^{-1}$	$\mu = \left(\boldsymbol{w}^T X\right)^{-1}$
Poisson	integer: $(0, \infty)$	population modelling, patient flow	$\eta = \log\mu$	$\mu = \exp\left(\boldsymbol{w}^T X\right)$
Bernoulli	integer: $(0, \infty)$	predicting readmission, mortality	$\eta = \log(\frac{\mu}{1-\mu})$	$\mu = \frac{1}{1+exp(-\boldsymbol{w}^T X)}$
Binomial	integer: $(0, \infty)$	predicting number of occurrences of a medical event		

Table 2.2: Common distributions in clinical applications along with link functions.

Brier score as:

$$\text{Brier score} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{2.12}$$

A lower Brier score implies better predictive performance.

Coefficient of Determination (R^2) In regression studies, R^2 is a measure of how well the model fits the data (Steel and James, 1960). If y_i, $\hat{y}_i$ are the i^{th} true outcome and predicted outcome for a sample of N observations, we calculate R^2 as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{2.13}$$

The numerator in (2.13) is the residual sum of squares which represents unexplained variation in the model. The denominator represents the total variability in the true outcome. Hence the ratio can be thought of as normalized total variability that is unexplained by the model. Subtracting this from one gives total explained variation in the model. A high value of R^2 translates to smaller unexplained variations and hence better model.

In case of logistic regression models, an equivalent of R^2 is pseudo-R^2. The pseudo-R^2 has a similar scale (from 0 to 1) with higher values indicating a better model, but

they cannot be interpreted the same way. Since a logistic regression model is derived from maximum likelihood estimates, most pesudo-R^2 measures compare a null model to the full model. A null model, $\mathcal{L}(\mathrm{M}_{null})$ is a model with no predictors and only the intercept, where as the full model, $\mathcal{L}(\mathrm{M}_{full})$, is a model with the full set of predictors. A common measure of pseudo-R^2 is McFadden's R^2.

McFadden's R^2 McFadden's R^2 compares the null and full models as:

$$\text{McFadden's } R^2 = 1 - \frac{\ln \mathcal{L}(\mathrm{M}_{full})}{\ln \mathcal{L}(\mathrm{M}_{null})} \tag{2.14}$$

The ratio of the likelihood is a measure of fit of the full model over the null (intercept only) model. Small ratios of log likelihood indicate a better fit.

Calibration and Hosmer-Lemeshow Goodness-of-fit Calibration is another method to examine whether the model assumptions conflict with data. For a well calibrated model, the proportion of predicted outcomes should be similar to the proportion of actual outcomes in the data (Steyerberg et al., 2010). A popular test for measuring this goodness-of-fit is the Hosmer-Lemeshow test.

Hosmer-Lemeshow test (Hosmer Jr et al., 2013) assesses the degree of fit by matching observed probabilities with estimated probabilities. The validation set is divided into G ordered groups based on estimated probability of outcome events. The Chi-squared test statistic is calculated by comparing the expected and observed number of outcome events in each group as:

$$\chi_{\mathrm{HL}} = \sum_{g=1}^{G} \frac{(O_g - E_g)^2}{E_g(1 - E_g/n_g)} \tag{2.15}$$

where $O_g =$ number of observed events in group g, $E_g =$ number of expected events in group g, and $n_g =$ number of observations in group g. For an ideal test, we have $G > 5$, $E_g > 5$ and $n_g = n_{g'}$, $(g, g') \in G$. When the significance of χ_{HL} is less than .05, we reject the null hypothesis which states there is no difference between estimated values and observed values. A large value for the test statistic with small significance (p-value < 0.05) indicates poor model fit while a small test statistic with large significance (p-value closer to 1) indicates a better fit (Pampel, 2000).

Classification Table A classification table can be used to measure the validity of predicted probabilities. For a 2-class problem, the classification table (also called confusion matrix) is tabulated as shown in Table. 2.3. Sensitivity measures the proportion of correctly classified events, whereas specificity measures the number of correctly classified non-events. Classification accuracy is proportion of correctly classified results (events and non-events) in the sample.

	Predicted Outcomes	
True Outcomes	Positive	Negative
Positive	TP	FN
Negative	FP	TN
Sensitivity = TP/(TP+FN)		
Specificity = TN/(TN+FP)		
Accuracy = TP + TN / (TP + TN + FP + FN)		

Table 2.3: Classification table for a 2-class problem

Discrimination and ROC Curve Besides being well calibrated, we would like our model to have high discriminative ability. The discrimination of a model represents the ability to distinguish between patients at high risk from those at relatively lower risk for a given event. A Receiver Operating Characteristic (ROC) curve measures discrimination by plotting the true positive rate (proportion of hits) against the false positive rate (proportion of false alarms, also expressed as 1-Specificity) (Bewick et al., 2004). For a perfect model, both sensitivity and specificity would be 1. Hence the ROC curve would start at the origin (0,0), climb through the Y-axis to (0,1) and remain constant to (1,1). A random guessing model is equally likely to produce a true positive or a false positive. The equality in true positive rate and false positive rate for a totally random model is represented as the 45° line from (0,0) to (1,1) in Figure 2.6. Area under the ROC Curve (AUC) is a single scalar value that quantifies classification performance. AUC values range from 0.5 to 1.0 where 0.5 represents a random model and 1.0 represents an ideal model. Thus, different models fitted to the same data can be compared based on their AUC. Better models have higher AUC.

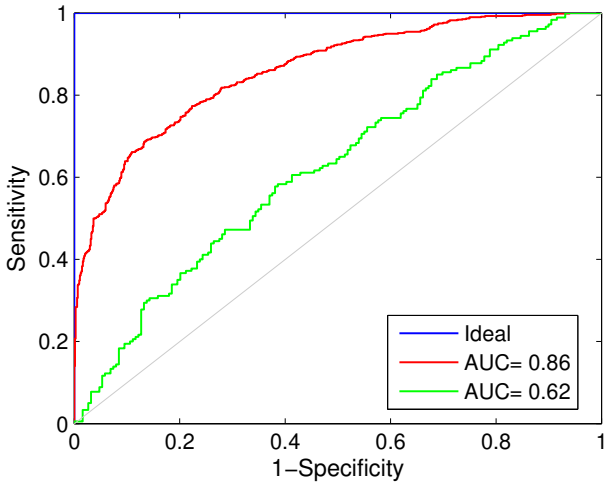


Figure 2.6: Example of ROC curve

2.1.5.6 Validating Prediction Models

Once a predictive model is built, it is important to determine how the model will generalize for external independent data. Model validation is an estimation of how well our derived model performs in practice. Validation techniques provide the means for unbiased evaluation of a predictive model. Depending on the nature of the data, these techniques are generally categorized as internal validation and external validation.

Internal Validation Here, the sample data for building the model is separated into training data and testing data. Two popular techniques used are cross-validation and bootstrap validation. In cross-validation, the sample data is randomly partitioned into complementary subsets which are used for analysis and validation (Kohavi et al., 1995). A popular approach is k-fold cross validation. Here, the data is partitioned into k equal sized subsets. Keeping one subset for testing and validation, the model is trained using the remaining $k-1$ subsets. This process is repeated k times (also called k-folds), with a different testing set during each iteration (fold). The k results from each fold is either averaged or combined for final evaluation.

During bootstrap validation, the training and testing data are created by repeatedly sampling (with replacement) the original data (Efron and Tibshirani, 1994). Hence B iterations of the bootstrap procedure will end up in B data samples that can be used in

learning and evaluation of the model. A comprehensive review of internal validation techniques applied to a logistic regression model predicting 30-day mortality after an acute myocardial infarction is presented in Steyerberg et al. (2001).

External Validation External validation techniques inspect the generalizability of the model when applied to other cohorts and settings (Reddy and Aggarwal, 2015; König et al., 2007). Two commonly used techniques are temporal validation and geographical validation. *Temporal validation* is when the model is derived from patients treated in the past and validated using more recently treated patients. *Geographical validation* of the model is performed using testing data from another geographic location (for example, patient data from another hospital).

2.2 Model Stability

Reproducibility, replication, assessments of variability are of crucial importance in any statistical research (Kass et al., 2016). Clinical prediction models derived from a small cohort need to generalize well on unseen patient groups. Stability of the learning process is crucial to obtaining conditions for generalization (Poggio et al., 2004; Bousquet and Elisseeff, 2002; Mukherjee et al., 2006). According to Turney (1995): "*The stability of a classification algorithm is the degree to which it generates repeatable results, given different batches of data from the same process*" . In the following sections, we elaborate on the notion of stability with emphasis to model stability and feature stability.

2.2.1 Stability

The most common notion of stability in computational learning theory is that of *algorithmic stability*, and was introduced by Devroye and Wagner (1979). This concept is also called perturbation analysis in statistics (Bonnans and Shapiro, 2013), while some machine learning literature attributes it to sensitivity analysis (Bousquet and Elisseeff, 2002). Essentially, algorithmic stability examines how perturbations in input affect prediction performance, and is closely associated with bounds for generalization error.

Generalization error is a measure of performance of the learning algorithm on unseen data. According to statistical learning theory, the learning process involves modelling the target function $f(.)$ using given input X and observations $\boldsymbol{y}$ to estimate a function $\hat{f}\ (.)$ that minimizes the empirical error (training error). For a given loss function $V(\hat{f}\ (X), \boldsymbol{y})$, the expected error of $\hat{f}\ (.)$ can be written as:

$$I[\hat{f}\ ] = \int_{X \times \boldsymbol{y}} V(\hat{f}\ (X), \boldsymbol{y}) \rho(X, \boldsymbol{y})\, dX d\boldsymbol{y} \tag{2.16}$$

where $\rho(X, \boldsymbol{y})$ is the joint distribution for X and $\boldsymbol{y}$. Ideally, we would want to choose the particular $\hat{f}\ (.)$ which minimizes $I[\hat{f}\ ]$, but $\rho(X, \boldsymbol{y})$ is unknown. However, we can calculate the empirical error from the given training data set: $S = (X, \boldsymbol{y})$. If the dataset has M instances, the empirical error becomes:

$$I_S[\hat{f}\ ] = \frac{1}{M} \sum_{i=1}^{M} V(\hat{f}\ (x_i), y_i) \tag{2.17}$$

Generalization error becomes the difference between expected error and empirical error. Mathematically, we can state this as: $G = I[\hat{f}\ ] - I_S[\hat{f}\ ]$. The learning function $\hat{f}$ generalizes well if $\lim_{n \to \infty} I[\hat{f}\ ] - I_S[\hat{f}\ ] = 0$. For any given domain, it is impossible to calculate $\rho(X, \boldsymbol{y})$, hence generalization error becomes impossible to compute. Instead, learning theory proposes to seek bounds for generalization error as:

$$P_G = P(I[\hat{f}\ ] - I_S[\hat{f}\ ] \leq \epsilon) \geq 1 - \delta$$

where ϵ is called the learning rate. The goal now becomes to characterize the probability $1 - \delta$ that the generalization error is less than an error bound ϵ.

Studies have shown that stable algorithms are able to generalize well. For example, Bousquet and Elisseeff (2002) introduced uniform stability: for any training set S, changing any example in S to any other possible example affects at most a small change in $\hat{f}$. They show that for uniform stability, mean generalization error becomes zero and proceed to demonstrate that regularization is uniformly stable. A common criticism of this work declares uniform stability to be too restrictive for general use (Poggio et al., 2004; Bousquet and Elisseeff, 2002; Mukherjee et al., 2006). Mukherjee et al. (2006) proposed that symmetric algorithms with bounded loss, leave-one-out cross-validation stability and expected leave-one-out cross-validation stability are generaliz-

able.

In our work, we focus on model stability – a generalized form of feature stability, which is related to algorithmic stability. The importance of model stability has been recognized since the last decade, when Famili and Turney (1991) used decision tree induction to generate rules to analyse why plans fail in an industrial planning system. Different batches of data from the same process resulted in vastly different decision trees, hindering interpretability and repeatability. In a follow up to this study, Turney (1995) proposed to measure stability of a learning algorithm based on agreement between learned concepts. The study defines concepts as explicit (for example decision tree, set of rules, feature weights), or implicit (set of stored instances), that are learnt from data during the training process. When subjected to variations in training data, the variations in learned concepts are measured using the semantic measure of *agreement*.

In this thesis, we investigate stability of linear prediction models by focussing on stability in ***feature selection*** and ***feature weights***. In the following sections we look at feature stability and stabilization methods.

2.2.2 Feature Selection Stability

The process of feature selection identifies a reduced subset of important features and removes the redundant ones from a given dataset. All further analysis is carried out using this identified subset. In such scenario, the stability of selected features is of much importance, since further analysis and model building critically depends on this set. Feature stability can be defined as the degree of agreement between feature subsets chosen by a given method to random perturbations of input data (Kuncheva, 2007; Loscalzo et al., 2009). We now describe the process of feature selection and the causes of instability.

2.2.2.1 Process of feature selection

Feature selection techniques serve as the workhorse for high-dimensional applications like bioinformatics and clinical prediction. These techniques remove unwanted, redundant and duplicate information from the dataset. This results in several advantages. A

reduced feature/predictor set results in simpler learning models. The learning performance may also improve (Guyon and Elisseeff, 2006). A reduced feature set also results in easier interpretation, visualization and reduced storage costs.

The process of feature selection can be supervised or unsupervised. Broadly, supervised feature selection techniques can be classified as filter based, wrapper based and embedded techniques.

Filter based techniques The filter methods are so named because they do feature selection (filtering) as a preprocessing step, before training the model. The selection process is independent of the learning model, and depend on the general characteristics of data and associated class label (Sánchez-Maroño et al., 2007). Classical methods consider each feature independently or with regards to the class label, and assigns a score for selection. For example, the popular fisher score and generalized fisher score evaluates features using fisher criterion when selecting optimum subsets (Gu et al., 2012).

Since the selection process is autonomous when compared to learning, these techniques can be faster and more generalizable. On the contrary, this may result in feature subsets that do not maximize the model performance. Wrapper based techniques resolve this dilemma, but at an expense.

Wrapper based techniques Unlike filter based methods, wrapper models choose the best feature subset using feedback from the learning model (Kohavi and John, 1997). Here, wrapper models formulate feature selection as a search problem. They construct and evaluate different combinations of feature sets. Each feature set is evaluated using a predictive model associated with the learning problem. The best feature set is chosen to be one that maximises model accuracy. Knowledge of the predictive model is not required and it can essentially act as a black box (Guyon and Elisseeff, 2003).

Feature subset creation can be methodical (best-fit search), stochastic (random hill-climbing algorithm) or heuristic (forward selection, backward elimination). An example of heuristic mode is the popular Recursive Feature Elimination Support Vector Machine (RFE-SVM) (Guyon et al., 2002).

When compared to filter methods, wrappers guarantee better features, but the process is computationally expensive. When large number of features are involved, filter methods are more efficient.

Embedded feature selection Embedded feature selection techniques, as the name suggests, are embedded in the learning algorithm. When the algorithm learns from a given dataset, it performs feature selection as a part of the learning process. The learning process is thus made of two competing objectives. (1) maximising the goodness-of-fit (model learning) (2) minimizing the number of model parameters (feature selection). Most popular embedded techniques are regularization methods such as lasso (Tibshirani, 1996), ridge regression (Ng, 2004) and elastic net (Zou and Hastie, 2005). In this thesis, we use lasso regularization for embedded feature selection

Sparse feature selection with Lasso Least absolute shrinkage and selection operator, or lasso, is a popular statistical method that simultaneously performs variable selection and regularization. Though initially introduced for least squares model, lasso has been successfully applied to generalized linear models, generalized estimating equations, proportional hazards models and M-estimators. For data containing many covariates, it becomes necessary to select a subset of strong features while minimizing prediction error. Lasso is able to achieve these goals by introducing the following constraint to learning model: the sum of absolute value of model parameters should be less than a predefined value, say t . Ensuring a sufficient value for t (often discovered during cross-validation) forces the coefficients of least predictive covariates to be zero, thereby choosing a simpler model. For a model with loss function $\mathcal{L}(\boldsymbol{w}|\mathcal{D})$, lasso regularization can be expressed as:

$$\begin{aligned} &\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}|\mathcal{D}) \\ &\quad \text{subject to} |\boldsymbol{w}| \leq t \end{aligned} \tag{2.18}$$

We can rewrite the general form in (2.18) as the Lagrangian form as:

$$\mathcal{L}_{\text{lasso}} = \min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}|\mathcal{D}) + \alpha ||\boldsymbol{w}||_1 \tag{2.19}$$

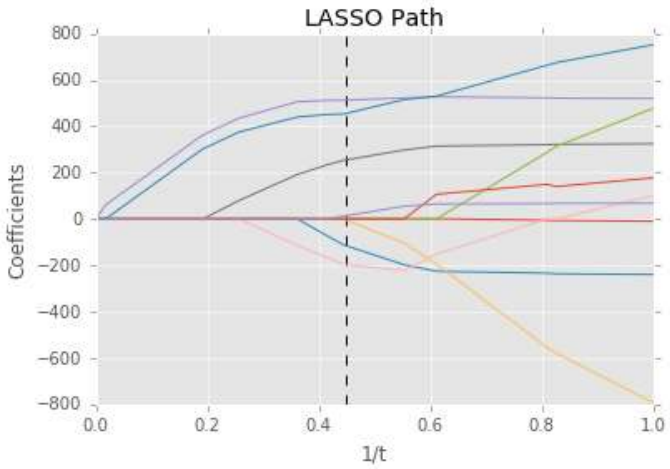


Figure 2.7: Effect of lasso regularization on a linear model derived from diabetics dataset used in Efron et al. (2004). The plotted lines trace values of each model coefficient for corresponding value of t.

Increasing the value of t (or α) results in shrinking the model parameters and inducing sparsity, thereby increasing bias and reducing variance. As example, Figure 2.7 illustrates the lasso regularization path for 10 coefficients of a linear model derived from a diabetic cohort used in Efron et al. (2004). For the optimum value of t (found using 10-fold cross-validation and illustrated by a vertical dashed line), the final model was described by 6 non-zero features. This property of lasso can be further understood by looking at the geometric and Bayesian interpretations.

Geometric interpretation of lasso The lasso constraint boundary due to the $\ell_1$norm is in general a cross-polytope. For ordinary least squares regression in two dimensions, lasso constraint region becomes a square with the corners meeting at X-axis and Y-axis, and the objective function level sets become elliptical centred at the OLS estimates as shown in Figure 2.8. The solution will be at the intersection of the contours of objective function and lasso. In most cases, this will be at the corners of the square (as shown in Figure 2.8) ensuring dimensionality reduction.

Bayesian interpretation of lasso When linear regression coefficients are assigned normal prior distributions, it becomes ridge regression. However, when they are assigned Laplace prior distributions, it becomes lasso regression. Laplace distributions are characterized by sharp peak at zero: as a result of two exponential distributions spliced

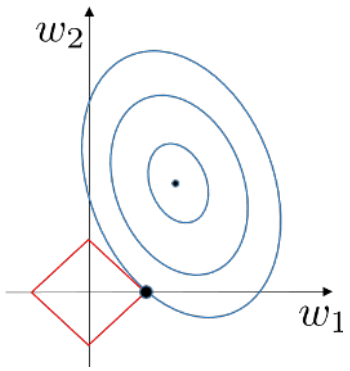


Figure 2.8: Lasso: Automatic shrinkage and variable selection in a 2D scenario. The blue contours represent the likelihood function, the red contours represent the $\ell_1$norm.

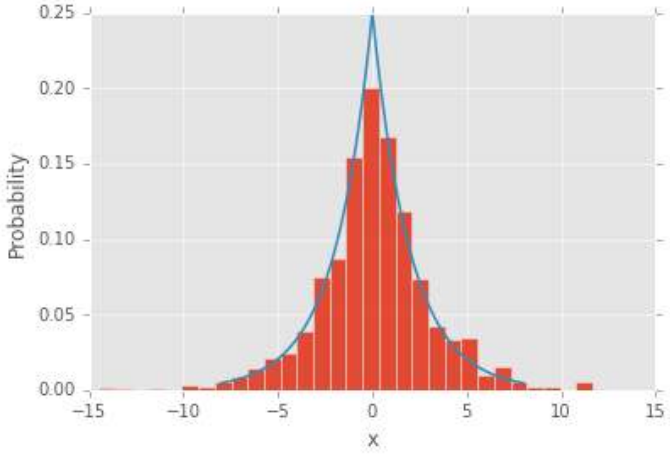


Figure 2.9: Histogram and probability density function of Laplace distribution with locality $\mu = 0$ and scale $b = 2$

together (illustrated in Figure 2.9). Hence, the gradient becomes discontinuous. The probability mass of Laplace distributions are closer to zero tending to suppress some lasso coefficients.

2.2.2.2 Causes of instability in feature selection

A feature selection method could result in a different subset of features during each training run, causing instability in selected features. Feature instability is a growing concern, particularly in high-dimensional datasets. A feature selection method could be unstable due to the following reasons:

***Design of the learning model*:** The "minimalist" design principles of the feature selection algorithm aims to find the minimum feature subset that maximizes performance (for example, accuracy or area under the ROC curve) (Yu et al., 2008; Awada et al., 2012). In this process, they often ignore stability of selected features. For example, embedded methods like lasso (Tibshirani, 1996) selects the strongest features ignoring redundant subsets that may contain important information.

***Presence of multiple feature subsets that result in similar model performance*:** As example, some bioinformatics datasets may contain multiple sets of true markers. Different runs of model training may select a different set of markers. While this could be primarily due to high correlation between the biomarkers (Yu et al., 2008), in some cases there may also be multiple non-correlated true markers (Zhang et al., 2008).

***Insufficient training data*:** High dimensional datasets with low sample sizes cause feature instability (Awada et al., 2012; Kim, 2009; Loscalzo et al., 2009). The typical domains which encounter this problem are clinical prediction (Austin and Tu, 2004; Gopakumar et al., 2015b; Tran et al., 2015b; Zhou et al., 2013), bioinformatics (Ein-Dor et al., 2006; Awada et al., 2012; He and Yu, 2010; Sun et al., 2014; Kim, 2009) and ecology (Dormann et al., 2013)

***Variance in the data*:** When there is variation between samples of training data, feature selection process may result in multiple outcomes during each training run (Alelyani et al., 2011; Han and Yu, 2012).

2.2.2.3 Stabilization Strategies

The previous section illustrated the causes for instability. We now look at the popular approaches for robust feature selection. Feature stability (also called selection stability) can be formally defined as: "*the sensitivity of a feature selection algorithm to perturbation in the training data*"(Kalousis et al., 2007; Křížek et al., 2007; Yu et al., 2008; He and Yu, 2010; Gulgezen et al., 2009). The existing methods for ensuring stable feature selection can be broadly classified into:

Group feature selection These techniques exploit a key observation: high dimensional data may contain groups of correlated features that are unaffected by data variation (Loscalzo et al., 2009; Yu et al., 2008). It is hypothesised that such correlations or feature relationships have some relevance to the associated class labels, and hence can be treated as a single group during feature ranking (Yu et al., 2008). Feature stability is ensured either by selecting one feature per identified group, or treating a whole correlated group as a single feature. Group based methods work in a two stage process: feature grouping and feature selection from the identified groups.

During feature grouping, we identify the intrinsic feature relationships using either a knowledge-driven approach or data driven approach. Knowledge driven approaches resort to existing domain knowledge to find feature correlations. For example, the bioinformatics domain utilize prior biological knowledge and pathway information to enhance the stability of biomarkers. These information, compiled from many years of research, is made available through online databases like KEGG, HPRD, Pathway Commons, Reactome, BioCarta and BioCyc (Li and Li, 2008; Cun and Fröhlich, 2013). Context specific data extracted from such databases can be used to create a graph network with nodes as genes or gene products and edges as interactions or relationships (Li and Li, 2008). Such networks can be used to stabilize learning models by either a filter based approach or using embedded feature selection techniques (Cun and Fröhlich, 2013). This approach has been used in clinical prediction to create feature correlations based on the hierarchical nature of diagnosis code (refer to EMR ICD-10 section) to stabilize feature stable models (Kamkar et al., 2015; Tran et al., 2015b; Gopakumar et al., 2015b).

Groups of identified features can also be converted into a single "*super feature*", using summary statistics (for example: mean, principal component analysis). These *super features* can then be used in place of individual features for feature selection (as in identifying biomarkers) or improving model performance (Chen et al., 2006; Chuang et al., 2007; Lee et al., 2008; Rapaport et al., 2007; Tai and Pan, 2007) .

Data driven methods learn the feature groupings/relationships directly from the given data, either using cluster analysis (Au et al., 2005; Hastie et al., 2001a; Ma et al., 2007; Park et al., 2007), density estimation (Loscalzo et al., 2009; Yu et al., 2008) or statistical analysis (Gopakumar et al., 2015a; Vinzamuri and Reddy, 2013). Cluster analysis methods employ clustering algorithms as K-means (Ma et al., 2007), attribute clustering

(Au et al., 2005) and hierarchical clustering (Park et al., 2007; Hastie et al., 2001a) to group similar features, whereas density estimation methods group features into clusters of similar densities using principles of kernel density estimation (Wand and Jones, 1994). Feature correlations were also discovered using RBF kernels (Vinzamuri and Reddy, 2013) and Jaccard similarity graphs (Gopakumar et al., 2015a).

Since lasso regularization ignores feature relationships, variations to lasso have been proposed that takes into account the many feature relationships present in data. Clinical data and biomedical data such as microarrays and genes often exhibit spatial, temporal or hierarchical relationships using trees and graphs (Yuan et al., 2009; Jenatton et al., 2011; Sun et al., 2014; Tran et al., 2014).

Group Lasso: was proposed by Yuan and Lin (2006) as a modification for lasso, when features exhibit natural groupings as in multifactor ANOVA problems, or gene cluster data (Ma et al., 2007) or analysis of PET images (Huang et al., 2009). When features are divided into k disjoint groups $\{G_1, G_2, \cdots, G_k\}$, group lasso modifies the ℓ_1- norm of lasso to $\ell_{q,1}$-norm penalty as:

$$\Omega_{\text{gLasso}}(\boldsymbol{w}) = \sum_{i=1}^{k} \lambda_i ||w_{G_i}||_q$$

where $||w_{G_i}||_q$ with $q > 1$ becomes the ℓ_q-norm of parameter w in group G_i, and λ_i is the weight for corresponding group G_i. Unlike lasso regression in (2.19), the group lasso formulation uses group information during feature selection. However $\Omega_{\text{gLasso}}(x)$ is unable to perform feature selection within each group. This can be made possible by extending group lasso to sparse group lasso (sgLasso) as

$$\Omega_{\text{sgLasso}}(\boldsymbol{w}) = \alpha||\boldsymbol{w}||_1 + (1-\alpha)\sum_{i=1}^{k} \lambda_i ||w_{G_i}||_q \tag{2.20}$$

where $\alpha \in [0, 1]$ controls the relative contribution of lasso term (sparsity at feature level) and group lasso term (sparsity at group level).

Tree Lasso: is an extension of group lasso, when feature groupings closely resemble a tree structure. In this case, we consider features at each node in the tree (Zhao et al.,

2009; Kim and Xing, 2010). The modified group lasso penalty can be written as:

$$\Omega_{\text{treeLasso}}(\boldsymbol{w}) = \sum_{i,j} \lambda_j^i \, ||w_{G_j^i}||_q$$

where G_j^i denotes group j containing subtree at depth i of the tree. In such tree structure, if a node is not selected, its children nodes will also be discarded.

Fused Lasso: was introduced by Tibshirani et al. (2005), and is yet another extension of lasso that takes into account the predefined structures in data. For example, when specific features are known to be adjacent (as in genomic data), the change in corresponding model parameters should be smooth. This smoothness structure is enforced by fused lasso as:

$$\Omega_{\text{fused}}(\boldsymbol{w}) = \alpha||\boldsymbol{w}||_1 + (1-\alpha)\sum_{i=1}^{N-1} |w_i - w_{i+1}|$$

As in sparse group lasso in (2.20), $\alpha \in [0, 1]$ controls the relative contribution of pure lasso component and smoothing component.

Ensemble feature selection Ensemble learning technique uses a voting mechanism to combine the outcome of several learners (Dietterich, 2000a; Bühlmann, 2012). Such methods are very popular since they often outperform a single model (Bühlmann, 2012). When applied to feature selection, ensemble techniques either use multiple feature rankers, or run feature selection multiple times to combine the results into a single feature list. During this process, strong features which appear frequently or consistently ranked higher are preferred over others, resulting in a more stable set.

Broadly, there can be three types of ensembles for feature selection: data ensemble, functional ensemble and hybrid ensemble. In data ensemble methods, a single feature selection algorithm is applied to multiple sub-samples (or bootstraps) of the training data. The final stability measure then becomes the average over pairwise comparisons over different samples (Saeys et al., 2008; Tran et al., 2015b; Kamkar et al., 2015; Gopakumar et al., 2015a). Functional ensemble techniques use multiple feature selection methods on the same training data. Finally, hybrid ensemble is a combination of data and functional variations. Here, different selection techniques are repeatedly applied to variations in training data.

There is no consensus on which of these methods perform the best (Awada et al., 2012). Kalousis et al. (2007) compared the stability of five popular feature selection algorithms on 11 datasets taken from three different application domains. Feature stability was investigated based on weight-scores, rank, and selected feature subsets. No algorithm was found to be superior and it was concluded that feature stability depends significantly on the dataset used.

Saeys et al. (2008) compared the stability of four methods: two filter based methods; Symmetrical Uncertainty (Press et al., 1996), RELIEF algorithm (Kononenko, 1994), and two embedded techniques: random forests (Breiman, 2001), linear support vector machines (Vapnik, 2013). A data ensemble of 40 bags of data was created using bootstrap. The feature rankings were aggregated using weighted voting.

Variance reduction method As mentioned in Section. 2.2.2.2, variation in the data could cause instability. Yu et al. (2008) proposed a two stage process for feature selection using variance reduction. In the first stage, each instance vector (x) is projected from its original space to a margin vector feature space calculated using its neighbouring instance vectors. Representation of this instance vector in this margin vector feature space (say x') reduces the effect of noise or outliers in the training data, thereby reducing data variance. In the second stage, each instance x is weighted using its average distance from all instances in the margin vector feature space. Algorithms like RELIEF (Kononenko, 1994) and SVM-RFE that use sample weighting for feature selection can now be applied on this data. This methodology was also used to find stable gene signatures from microarray data and outperformed ensemble methods (Yu et al., 2012).

The dilemma in assessing feature stability in face of sample variance was investigated by Alelyani et al. (2011). This research concluded that similarity between training samples should be considered when assessing stability of an algorithm.

2.2.3 Evaluation of Model Stability

Two aspects are involved in evaluating stability: (i) a framework for testing stability (ii) a mathematical measure for stability (Awada et al., 2012; Khoshgoftaar et al., 2013). We shall look at each aspect in detail.

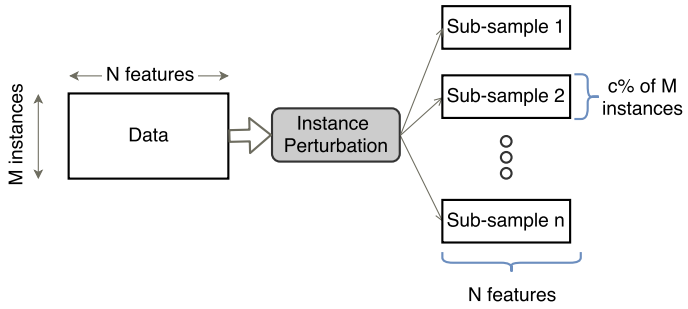


Figure 2.10: Instance perturbation for measuring stability.

2.2.3.1 Framework for testing stability

A framework to test stability usually involves some method that introduces perturbations in training data. Such perturbations usually involve randomly removing data instances, features or both. During instance perturbation of a given dataset of m instances and n features, a modified dataset is created by retaining only a fraction c of the original dataset. The remaining fraction of $(1 - c)$ instances are dropped. This process is randomly repeated to create multiple training sub-samples, as shown in Figure. 2.10.

Such techniques have been used in the works of Saeys et al. (2008); Boulesteix and Slawski (2009); Dittman et al. (2011); Wang et al. (2011) to measure the stability of ranked feature lists. Wang et al. (2011) used perturbations using four c values (95%, 90%, 80%, 66.67%) to measure the stability of 18 feature rankers on 3 software engineering datasets. When creating training sub-samples, Alelyani et al. (2011) cautioned that variance among samples could influence the assessment of stability. The dilemma in selection stability is: are the selected feature subsets different due to the instability of the feature ranker, or due to the difference in training data? Their study proposes considering the percentage of overlap between training sub-samples, demonstrating higher stability when overlap is higher. The study concludes that when data variance is not considered, current methods do not assess stability, rather rank the algorithms according to repeatability of results.

Cross-validation is also a popular method in dataset perturbation (Van Hulse et al., 2009). Cross-validation (also called rotation estimation) is generally used to prevent overfitting in machine learning. The process involves dividing the training data into k equal partitions or folds of the same size. The model is trained using the first $k-1$ folds and tested on the remaining folds. This process is repeated k times, where each fold is

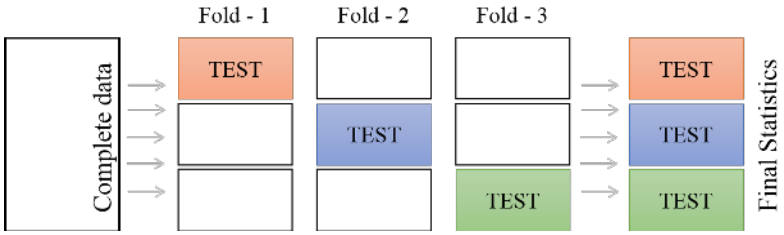


Figure 2.11: The process of cross validation with number of folds (k) as 3.

used for testing (as illustrated in Figure. 2.11). There is no overlap between the folds. A recent study suggests that stability is best evaluated on datasets with no overlap (Haury et al., 2011). A more recent study introduces a predetermined amount of overlap by proposing a fixed overlap partitioning algorithm to create two subsets of the same size (Wang et al., 2012b). The control on overlap allows to test the stability of algorithms as suggested by Alelyani et al. (2011).

Finally, bootstrapping (random sampling with replacement) is a well accepted method of sampling the training data.

2.2.3.2 Measuring Stability

Once the model is run on each of the training sub-samples, we need a similarity measure to assess the amount of agreement among model parameters during each training run. Kalousis et al. (2007) has broadly classified the existing stability measures into three categories:

1. Stability by index: these measures see if a particular feature is selected or rejected, without considering ranking or relevance weights.
2. Stability by rank: these measures take into account the rank of selected features.
3. Stability by weight: these measures look at correlation between weights of corresponding features.

Though features are selected based on the weights assigned by the learning model, different applications would be interested in specific information: some applications would

require to know if a feature is selected or not. Some applications would require a ranking of features, while some other would require the exact weights.

Stability by index The measures in this category quantify the amount of overlap between selected feature subsets. These feature subsets are not ordered by rank or weight. Let us assume that the models were trained on K sub-samples of data resulting in list of feature subsets as: $S = \{S_1, S_2, \cdots, S_K\}$. Further, let each feature subset contain top k selected features. Hence we have $|S_i| = k$. We use this notation to explain the following measures.

Hamming distance Hamming distance (Hamming, 1950) is quite popular in coding theory to quantify the similarity between equal-length strings. Given two feature subsets: S_i and S_j, the pairwise hamming distance can be written as:

$$H(S_i, S_j) = \sum_{p=1}^{n} |S_{ip} - S_{jp}|$$

where S_{ip} is the p^{th} feature in subset S_i with a total of n features. Here each feature subset S_i is a binary vector, where the components indicate the presence or absence of a feature. For K feature subsets in our data, the total Hamming distance becomes:

$$H_t = \sum_{i=1}^{|K|-1} \sum_{j=i+1}^{|K|} H(S_i, S_j)$$

The averaged normalised Hamming distance (ANHD) represents the stability across all feature pairs and is calculated by normalising H_t as:

$$\widehat{H} = \frac{2 \times H_t}{n \times |K| \times (|K| - 1)}$$

Dunne et al. (2002) used the averaged normalized Hamming distance to measure the selection stability of wrapper based models on 4 datasets. ANHD is in the range $[0, 1]$, where 0 represents maximum similarity and 1 represents maximum variance.

Jaccard index Jaccard index (Real and Vargas, 1996) measures similarity as a fraction between cardinalities of intersection and union feature subsets. Given two feature sets S_i and S_j, the pairwise Jaccard index reads:

$$J_C(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{2.21}$$

The Jaccard index evaluating all K subsets is averaged as:

$$J_S = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} J_C(S_i, S_j) \tag{2.22}$$

Jaccard index is bounded in $[0, 1]$ and increases when k increases. The Jaccard index or Jaccard similarity coefficient is a popular measure and has been used to quantify feature stability in the works of Saeys et al. (2008); Alelyani et al. (2011); Peteiro-Barral et al. (2012); Gopakumar et al. (2015b); Kamkar et al. (2015).

Tanimoto Distance is a generalization of Jaccard index, and is formulated by re-writing (2.21) as:

$$T(S_i,\ S_j) = 1 - \frac{|S_i| + |S_j| - 2\,|S_i \cap S_j|}{|S_i| + |S_j| - |S_i \cap S_j|} \tag{2.23}$$

The Tanimoto distance is a generalization of Jaccard index to multiple classes. Here, (2.23) measures the amount of overlap between two subsets of arbitrary cardinality. Kalousis et al. (2007) used this measure to evaluate the stability of six feature selection methods.

Dice's Coefficient is also related to Jaccard index. It is also known as Sørensen Dice coefficient or Sørensen index and was used to calculate feature stability from microarray data (Yu et al., 2008). This measure is a variation of (2.21) as:

$$D(S_i, S_j) = \frac{2 \times |S_i \cap S_j|}{|S_i| + |S_j|} \tag{2.24}$$

As with Jaccard and Tanimoto metrics, Dice's coefficient takes values between 0 and 1, where 0 indicates no overlap and 1 indicates complete overlap. Although Dice, Jaccard and Tanimoto indices have similar characteristics, dice similarity measure returns more meaningful results. For example, if two subsets S_i and S_j with $k = 10$ features have

5 common features ($|S_i \cap S_j| = 5$), Dice's coefficient by (2.24) becomes 0.5 which is closer to 50% overlap than the values of Tanimoto and Jaccard index which returns 0.33. An issue with all three measures is increase in score as the size of S_i increases. This could be attributed to overlapping of large subsets due to chance.

Consistency Index also called Kuncheva index was proposed by Kuncheva (Kuncheva, 2007) to correct for the overlapping due to chance. Considering a pair of subsets S_i and S_j, the pairwise Consistency index I_C is defined as:

$$I_C(S_i, S_j) = \frac{rd - k^2}{k(d-k)} \tag{2.25}$$

in which $|S_i \cap S_j| = r$ and d is the total number of features in the original data. Taking the average of all pairs, the overall Consistency index is:

$$I_S = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} I_C(S_i, S_j) \tag{2.26}$$

The Consistency index is bounded in $[-1, +1]$.

Stability by rank These metrics require the feature subsets to be ordered by ranks. Unlike stability by index measures, they cannot handle subsets with different cardinalities; they operate on the full feature set. Popular measures in this category are as follows.

Spearman's rank correlation coefficient (SRCC) Given two ranked subsets as r and r' with m observations, the pairwise SRCC becomes:

$$SRCC(r,\, r') = 1 - 6 \sum_i \frac{(r_i - r'_i)^2}{m(m^2-1)}$$

The value of SRCC will vary from -1 (inverse correlation) to 1 (perfect correlation), with 0 representing no correlation. This metric was used to verify the rank stability of six feature selection methods (Kalousis et al., 2007).

Canberra Distance (CD) measures the absolute difference between two ranked feature sets r and r' as:

$$CD(r,\, r') = \sum_{i=1}^{N} \frac{|r_i - r'_i|}{r_i + r'_i} \tag{2.27}$$

Jurman et al. (2008) used a weighted version of (2.27) to study the stability of top k ranked subsets. There is no upper bound for the formulation in (2.27); the value increases with increasing number of features.

Stability by Weight These measures look at the variation in weights assigned to features among the feature subsets. The feature subsets should be of the same size. The popular measures are detailed below.

Pearson's Correlation Coefficient (PCC) calculates the correlation between two weighted sets w and w' as:

$$PCC(w,\, w') = \frac{\sum_i (w_i - \mu_w)(w'_i - \mu_{w'})}{\sqrt{\sum_i (w_i - \mu_w)^2 \sum_i (w'_i - \mu_{w'})^2}} \tag{2.28}$$

where μ is mean. Pearson's correlation coefficient ranges from -1 (anti-correlation) to +1 (perfect correlation), with 0 representing no correlation. This measure was used in a study of stable feature selection in high dimensional space (Kalousis et al., 2007).

Signal-to-Noise Ratio (SNR) We borrow the concept from signal processing to measure the robustness of feature weights against variations in the subset. For the i^{th} feature, if the mean feature weight across K subsets is $\bar{w}_i$ with corresponding standard deviation as σ_i, then signal-to-noise ratio becomes:

$$SNR(i) = \frac{\bar{w}_i}{\sigma_i}$$

This metric has no upper bound, and increases with increasing feature weights. Recently, SNR was used to measure the stability of model parameters in clinical prediction applications (Tran et al., 2015b; Gopakumar et al., 2015a,b).

2.3 Concluding remarks

The research direction of this thesis ultimately aims to enhance the stability of any and all predictive discoveries. This is important for a number of reasons. First and foremost, models discover truth about data, and universal truth do not vary. Hence stability should be one of the most important characteristic when selecting a model. Biologically plausible models are generally stable against data variations. Further, models need to be transferable and generalizable from one cohort to another. High quality training data is hard to obtain. A similar work in stability by Zhou et al. (2013) declares: "*Because of the highly noisy nature of EHR data, clinical experts often have to be involved in the annotation process in order to obtain reliably labeled training data. As a result, in many cases only limited labeled data can be obtained.*" In such conditions, its imperative that models generalize well, and transfer between cohorts. The experiments on data need to be reproducible for clinical adoption. But the nature of our data introduces the following problem.

Clinical data used in this thesis is a classic example of large p small n paradigm, characterized by large number of features (p) with relatively smaller number of samples (n). Hence, we need a strong feature selection technique to provide insights into underlying causal relationships by focusing on smaller feature subsets, exclude noisy features for more reliable estimates and derive faster more efficient models for further analysis (Ma and Huang, 2008).

In this thesis, we focus on lasso regularization (as detailed in Section 2.2.2.1) for sparse feature selection. We choose lasso because of it is highly interpretable, and interpretability is key for clinical process. For a classification problem, lasso sets the weights of weak covariates to be exactly zero during optimization. Such features have no discriminatory power between classes. Hence lasso regularization results in simultaneous shrinkage and variable selection. The weights of non-zero covariates represent the relative importance of features that is able to discriminate between classes, making the model highly interpretable. The interpretability of lasso is also discussed in literature, most notably by James et al. (2013), which illustrates that restrictive models are in general more interpretable, and hence preferred when the goal is inference from data (Figure 2.12).

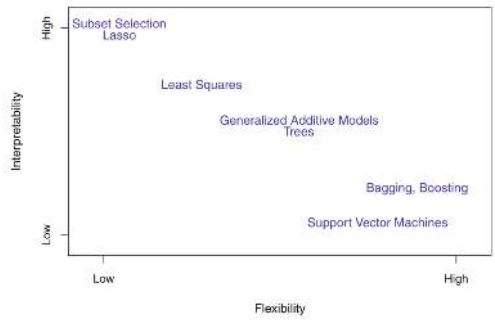


Figure 2.12: Interpretability of lasso and other traditional methods. Adapted from James et al. (2013)

In the next chapter, we explain the instability problem of lasso regularization using a case-study of modelling patient flow. The following chapters detail three novel strategies to overcome this instability. When compared to traditional lasso variants such as group lasso, tree lasso, fused lasso and elastic nets (Section 2.2.2.3), our methods address the following limitations. Elastic net regularization encourages grouping effect, but does not take the underlying structure of features into account. While tree lasso, group lasso and fused lasso require predefined feature groupings as input, our methods are data driven and automatically discover underlying feature groupings and latent structures.

We now proceed to illustrate model instability problem in clinical domain with a case study using data from a regional hospital in Victoria, Australia.

"I can't explain myself", said Alice, "because I'm not myself, you see."

Alice in Wonderland, Chapter V

Chapter 3

Model Instability: A Case Study

Model stability, in the scope of this thesis, refers to the sensitivity of learning model parameters to variations in the training data. Specifically, we look at two important model parameters: (i) predictors or features selected by the learning algorithm to represent the final model, and (ii) the corresponding feature weights. Due to the nature of medical data (as detailed in Section 2.1.3), popular learning algorithms become susceptible to model instability. In this chapter, we demonstrate model instability in a medical setting. To this purpose, we address the open and important problem of forecasting daily discharges from a ward with no real-time clinical data.We study patient outflow from an open ward in an Australian hospital, where currently bed allocation is carried out by a manager relying on past experiences and looking at demand. We build three linear and three non-linear models to predict the total number of next-day discharges. The data for all our models is extracted from the hospital database and consists solely of administrative information. The ward presented no real-time clinical data.

We begin by giving a brief background on predicting patient discharge, and then proceed to describe our data extraction process. Next, we introduce the six popular prediction algorithms and explain our experimental setting. We demonstrate that the algorithms have comparable performances. But in choosing the most interpretable model, we encounter instability in predictors and parameters. We conclude by proposing our stabilization strategies.

3.1 On ward-level forecasting

Discharge forecasting is an important tool for efficient bed management (Wong et al., 2010), which is critical for meeting rising demand in health services and reducing cost. Such demand has become unsustainable in recent decades (Kalache and Gatti, 2002; OECD, 2003). This is largely due to increase in population and life expectancy, escalating costs, increased patient expectations and workforce issues (Mackay and Lee, 2005). Efficient bed management is highly challenging given the number of inpatient beds in hospitals has come down by 2% since the last decade (OECD, 2003; Alijani et al., 2003).

Daily discharge rate is a real-time indicator of operational efficiency (Wong et al., 2010). From a ward-level perspective, a good estimate of next-day discharges will enable hospital staff to foresee potential problems such as changes in number of available beds and changes in number of required staff. Efficient forecasting reduces bed crisis and improves resource allocation. This foresight can help accelerate discharge preparation, which has huge cost on clinical staff and educating patients and family, requiring post-discharge planning (Connolly et al., 2010, 2009). However, studying patient flow from general wards offers several challenges.

Ward-level discharges incorporate far greater hospital dynamics that are often non-linear (Harper and Shahani, 2002). Accessing real-time clinical information in wards can be difficult because of administrative and procedural barriers, such data may not be available for predictive applications. Because the diagnosis coding is performed after discharge, there is little information about medical condition or variation in care quality in real time. In addition, factors other than patient condition play a role in discharge decisions (Wong et al., 2009, 2010; van Walraven and Bell, 2002).

The current practice of bed allocation in general wards of most hospitals involve a hospital staff/team, who use past information and experience, to schedule and assign beds (Daniels et al., 2005). Modern machine learning techniques can be used to aid such decisions and help understand the underlying process. As an example, Figure 3.1 illustrates a decision tree trained on past discharges and ward occupancy statistics, which models the daily discharge pattern from an open ward in a regional Australian hospital. Although the absence of patient medical information affected forecast performance, the decision rules provide important insight into the discharge process.

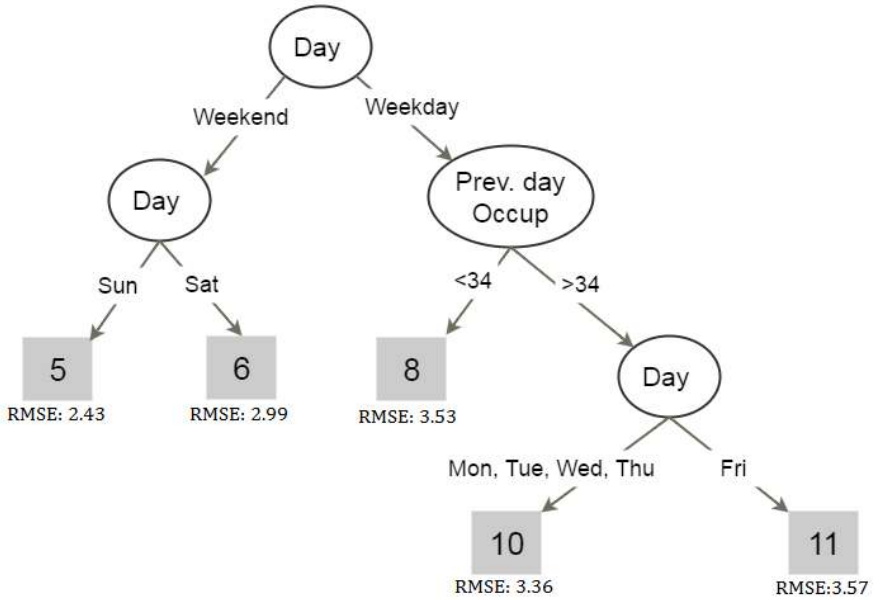


Figure 3.1: Decision tree modelling of total discharges from an open ward using day of the week and ward occupancy (Prev. day Occup) data for five years. The leafs represent total number of patient discharges.

3.2 Methods

We describe seven diverse methods that are applicable to forecasting under complex data dynamics. Of these, three methods are linear: (1) the classical autoregressive integrated moving average (ARIMA), (2) autoregressive moving average with exogenous variables (ARMAX) and (3) Sparse Linear Regression. Rest of the methods exploit non-linearity to model data. Specifically, we employ the most popular non-linear models: (1) k-nearest neighbour (kNN) regression, (2) Decision trees, (3) random forest (RF) regression, and (4) support vector regression (SVR). Autoregressive methods and linear regression model temporal linear correlation between nearby data points in the time series. Nearest patterns lift this linearity assumption and assumes that short periods form repeated patterns. Decision trees, RF and SVR models look for a non-linear functional relationship between the future outcomes and descriptors in the past. Finally, we inspect the stability (in terms of model reproducibility) for each of these models.

We formally begin our discussion by detailing the feature extraction process. This process is common to all models.

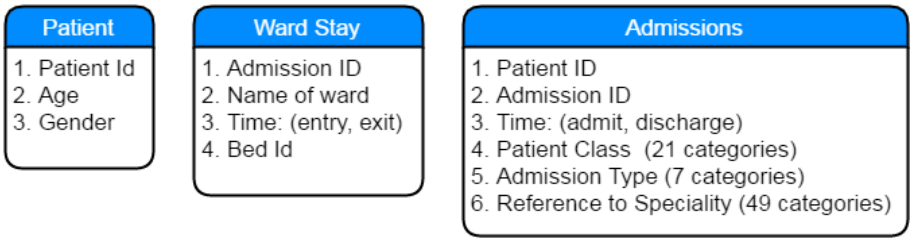


Figure 3.2: Tables in hospital database used in our data collection

Total patient visits	12,141
Unique patients	10,610
Length of stay: mean, median, IQR	4.26, 3, 5
Discharges per day: mean, median, IQR	8.7, 8, 5
Admissions per day: mean, median, IQR	8.6, 8, 5
Mean ward occupancy, IQR	30.9, 4
Gender	54.8% Female
Age: mean, median	66, 63.23

Table 3.1: Cohort details

3.2.1 Data and Feature Extraction

The data for our case study was collected from Barwon Health, a regional hospital in Australia. The total number of available beds depended on the number of staff assigned to the ward. On average, the ward had 36 staffed beds, but fluctuated between 20 and 80 beds with varying patient flow. The physicians in the ward had no teaching responsibilities.

The data for our study came from three tables in the hospital database, as shown in Figure. 3.2. Additional real-time data that described patient condition or disease progression were unavailable because diagnosis coding using medical codes is done after discharge. Patient flow was collected for a period of 4 years. Using the admission and discharge times for each patient, we calculated the daily discharges from our ward in study. A total of 12,141 patients were admitted into the ward with a median discharge of 8 patients per day from January 1, 2010, to December 31, 2014. Table. 3.1 summarizes the main characteristics of our data.

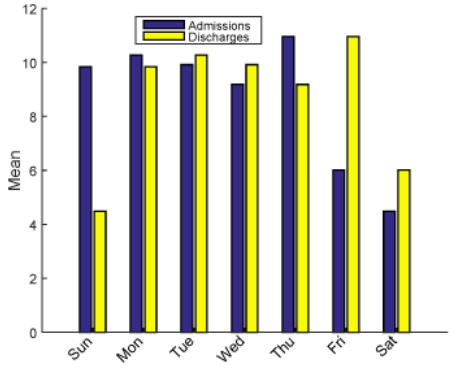


Figure 3.3: Mean admissions and discharges per day from ward.

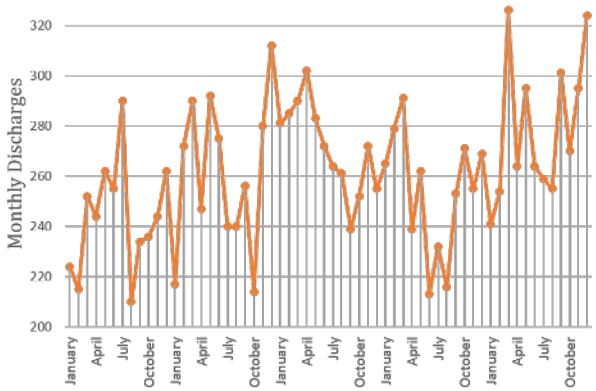


Figure 3.4: Time series of monthly discharges from ward.

A time series decomposition of our data revealed strong seasonal variations and high non-linearity in daily discharge patterns. There was a defined weekly pattern–discharge from ward peaked on Fridays and dropped significantly on weekends (see Figure 2). This seasonal nature is in tune with previous studies (Wong et al., 2009; Lin et al., 2011). Aggregating the daily discharges into a monthly time series revealed defined monthly patterns (see Figure 3.4). The data displayed no significant trend. In addition, the daily discharge pattern was found to be highly non-linear. Our forecasting methods must be able to handle such data dynamics.

As the first step in our case study, we inspect and extract features from commonly available administrative data in the hospital database. Two main groups of features were identified: (1) ward level and (2) patient level. Our feature creation process resulted in 20 ward-level and 88 patient-level predictors, as listed in Table 3. The ward-level descriptor: trend of next-day discharge was calculated by fitting a locally weighted

Ward level predictors	
Seasonality	current day-of-week, current month
Trend	calculated using locally weighted polynomial regression from past discharges on the same weekday
Admissions	Number of admissions during past 7 days
Discharges	Number of discharges during past 7 days, number of discharges in previous 14^{th} day and 21^{st} day
Occupancy	ward occupancy in previous day
Patient level predictors	
Admission type	5 categories
Patient Referral	49 categories
Patient Class	*21* categories
Age Category	8 categories
Number of wards visited	4 categories
Elapsed length of stay	Calculated daily for each patient in the ward

Table 3.2: Features constructed from ward data in hospital database. The random forest and support vector regression models used the full set of features. The ARMAX (autoregressive moving average with exogenous variables) model used seasonality and occupancy. All other models were derived from daily discharges.

polynomial regression (Cleveland et al., 1992) from past discharges. An example of this regression fitting is shown in Figure 3.5.

3.2.2 Classic Forecasting Methods

Here, we describe two most common techniques to model forecasts - ARIMA and ARMAX. These methods are linear, hence they are relatively simple and interpretable. They are also surprisingly effective and are often used as benchmarks for more complex hypothesis.

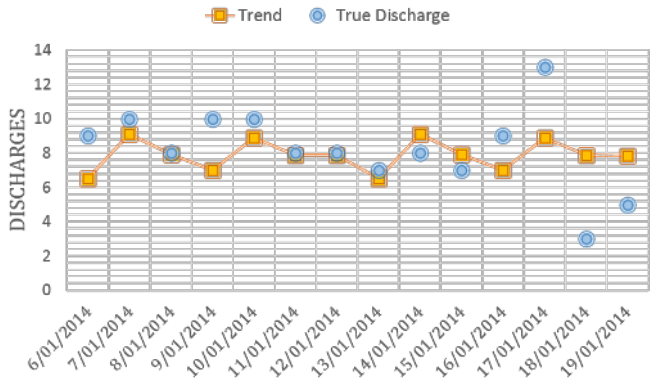


Figure 3.5: An example of the discharge trend, as derived from a locally weighted polynomial regression model.

3.2.2.1 Autoregressive Integrated Moving Average (ARIMA)

Time series is defined as a sequence of observations over time (Chatfield, 2013). Daily discharges from the ward is a discrete time series. To predict next day discharge, we capture the variation in discharge series. Traditional methods in time series analysis decompose this variation into a trend component, a seasonal component and irregular fluctuations (or noise) (Chatfield, 2013). Trend signifies the long term change in the mean level of the time series. Trend can increase or decrease in a linear or non-linear manner. Seasonal component captures the regular or semi-regular variations in data. Seasonal variations (also called seasonality) in data refers to predictable changes that repeats within a time frame (weekly, monthly or annual).

Time-series forecasting methods can analyse the pattern of past discharges and formulate a forecasting model from underlying temporal relationships (Chatfield, 2013). Such models can then be used to extrapolate the discharge time series into the future. Autoregressive Integrated Moving Average (ARIMA) models are widely used in time-series forecasting. Their popularity can be attributed to ease of model formulation and interpretability (Kane et al., 2014). ARIMA models look for linear relationships in the discharge sequence to detect local trends and seasonality. However, such relationships can change over time. ARIMA models are able to capture these changes and update themselves accordingly. This is done by combining autoregressive (AR) and moving average (MA) models. Autoregressive models formulate discharge at time $t = y_t$, as a linear combination of previous discharges. On the other hand, moving average models characterize the discharge at time t as linear combination of previous forecast errors.

For ARIMA model, the discharge time series is made stationary using differencing. Let ϕ be autoregressive parameters, θ be moving average parameters, and ϵ be the forecast errors. Such an ARIMA model can be defined as:

$$y_t = \mu + \sum_{i=1}^{p} \phi_i \, y_{t-i} + \epsilon_t - \sum_{i=1}^{q} \theta_i \, \epsilon_{t-i} \tag{3.1}$$

where μ is a constant. By varying p and q, we can generate different models to fit the data. Box Jenkins method (Box and Jenkins, 1990) provides a well-defined approach for model identification and parameter estimation. In our experiments, we choose the *auto.arima()* function from the forecast package (Hyndman and Khandakar, 2008) in R (R Core Team, 2013) to automatically select the best model.

3.2.2.2 Autoregressive Moving Average With Exogenous Variables (ARMAX)

Dynamic regression techniques allow adding additional explanatory variables, like day of the week and number of current patients in the ward, to autoregressive models. Autoregressive moving average with exogenous variables (ARMAX) modifies ARIMA model by including external variable x_t at time t, as shown in (3.2). We model x_t using features from the hospital database.

$$y_t = \mu + \sum_{i=1}^{p} \phi_i \, y_{t-i} + \epsilon_t - \sum_{i=1}^{q} \theta_i \, \epsilon_{t-i} + \beta x_t \tag{3.2}$$

3.2.3 Sparse Linear Regression

Classic forecasting relies on strong assumption of the temporal dynamics. However, the methods described till now only used a small subset of extracted features. Our feature extraction process was designed to generate data descriptors that are expected to contain all information of the history and dynamics of patient flow from available administrative data. Machine learning algorithms like regression can then learn to combine these features to predict the future. To this end, we resort to sparse linear regression. Linear regression (detailed in Section 2.1.5.1) is able to model future discharges using a linear combination of all available descriptors, while sparsity ensures interpretability and

discards irrelevant features.

We describe our sparse linear regression model as follows. Let $\mathcal{D} = \{\boldsymbol{x}_\ell, y_\ell\}_{\ell=1}^n$ be the training dataset. For each day $\ell \in [1, n]$, where n is the total number of days in data, $\boldsymbol{x}_\ell \in \mathbb{R}^p$ denotes the high-dimensional feature vector and y_ℓ is the number of patients discharged on day ℓ. Linear regression assumes $y = \beta + \boldsymbol{w}^\top \boldsymbol{x} + \epsilon$ where β is the mean output, $\boldsymbol{w} \in \mathbb{R}^p$ are sparse feature weights, and ϵ is random noise. The weights are estimated by maximizing lasso (Tibshirani, 1996):

$$\mathcal{L}_{\text{lasso}} = \frac{1}{2} \sum_{\ell=1}^{n} \left(y_\ell - \beta - \boldsymbol{w}^\top \boldsymbol{x}_\ell \right)^2 - \alpha \sum_i |w_i| \tag{3.3}$$

where $\alpha > 0$ is the penalty controlling sparseness of the feature weights. Lasso checks overfitting while simultaneously performing feature selection. Under lasso, weights of weak features are driven towards zeros, and thus the resulting model is sparse. This process has been detailed in Section 2.2.2.1.

The main advantages of lasso-based forecasting are that lasso methods tend to be more interpretable. Sparse models have a smaller feature subset. Prediction can be explained using this smaller subset, with feature weights indicating the relative contribution of each feature. This leads to a simple check-list style estimate for understanding the prediction process.

3.2.4 Machine Learning: Non-linear Methods

Linear methods may be less optimal in predictive power if the outcome is non-linear in features. For example, the ARIMA assumes that data is linearly auto-regressive. However, a close examination on the time-series suggests there are strong weekly patterns with complex dynamics. Under the lack of theoretical structure of the dynamics, we assume that although they are complex, the patterns might be repetitive over time. No further assumption is then made. To provide an estimate of upper-bound on predictive accuracy, we employ several best-known non-linear methods. These methods may be better suited to handle the underlying data dynamics. However, they lack interpretability and it is difficult to assess feature importance in some cases.

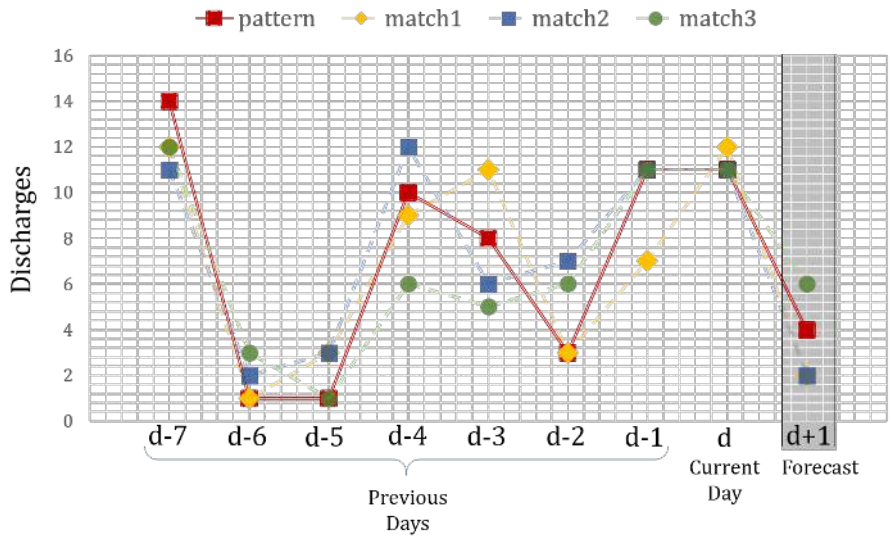


Figure 3.6: k-nearest neighbour forecasting example with k = 3 and P = 7.

3.2.4.1 k-Nearest Neighbours

k-nearest neighbours (kNN) (Cover and Hart, 1967) are effective in exploiting repeated patterns. The kNN algorithm has been successfully applied to forecast to histogram time series in financial data (Arroyo and Maté, 2009). The non-parametric regression using kNN was also used for short term traffic forecasting (Davis and Nihan, 1991). However, kNN regression has not been studied for patient flow.

The basic assumption is that similar historical patterns will result in similar outcome in the near future. The kNN algorithm takes advantage of the locality in data space. We assume the next day discharge depends on the discharges happening in previous d days. Using kNN principles, we can do a regression to forecast the next day discharge. To forecast the next day discharge: y_{t+1}, we look at the discharges over the past d days as: disch_vec $= [y_{t-d} : y_t]$. Using Euclidean distance metric, we find k closest matches to disch_vec from the training data. An estimate of next day discharge $\hat{y}_{t+1}$ is calculated as a measure of the next day discharges of the k matched patterns $(y_{\text{match}})_i$, $i \in (1 : k)$. Figure 3.6 shows an example of kNN based forecasting. Here, disch_vec in red $[y_{d-7} : y_d]$ results in 3 matches from the training data. For simplicity, we have plotted the matched patterns alongside disch_vec, although they had occurred in the past. The next-day forecast $\hat{y}_{d+1}$ becomes a measure of $(y_{\text{match}})_i$, where $(y_{\text{match}})_i$ $i \in (1 : 3)$ is the $(d+1)^{th}$ term of each of the matched patterns (Altman, 1992).

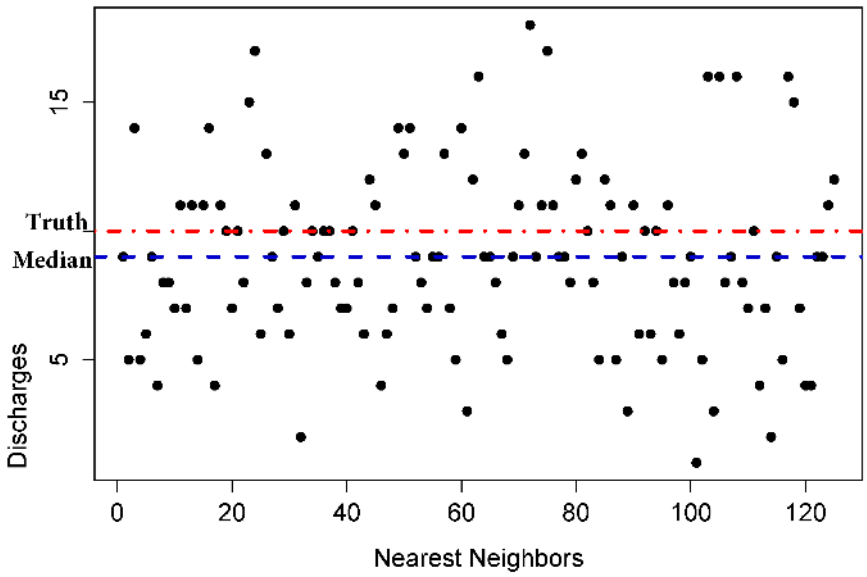


Figure 3.7: Scatterplot of next-day forecast using k-nearest neighbour for a given day. X-axis represents each matched nearest neighbour pattern. Y-axis represents the next day forecast of that matched pattern.

One popular method of calculating $\hat{y}_{t+1}$ is by minimizing the weighted quadratic loss:

$$\begin{aligned} \hat{y}_{t+1} &= \min_y \sum_{i=1}^{k} w_i \left((y_{\text{match}})_i - y\right)^2 \\ &= \sum_{i=1}^{k} w_i (y_{\text{match}})_i \end{aligned}$$

where w_i is subject to $w_i \in (0,1)$ and $\sum_{i=1}^{k} w_i = 1$. However there are two main drawbacks making it less desirable for our data. First, the quadratic loss is sensitive to outliers. Second, it is difficult to robustly estimate $\{w_i\}$. Our data contains significant noise, causing large variations in next day forecasts of the k matched patterns. The problem is illustrated in Fig. 3.7. The scatterplot of next day forecasts from the matched 125 patterns display significant variations. In such scenario, we resort to estimating $\hat{y}_{t+1}$ by minimizing the following robust loss:

$$\begin{aligned} \hat{y}_{t+1} &= \min_y \left(\sum_{i=1}^{k} |(y_{\text{match}})_i - y| \right) \\ &= \text{median } [(y_{\text{match}})_{i=1 \text{ to k}}] \end{aligned}$$

3.2.4.2 Decision tree and Random Forest

While kNN is non-parametric and the assumptions are minimum, it requires a large amount of data to search for a good local match. The assumption that the surface patterns repeat may not hold due to constant changes in healthcare dynamics. The kNN algorithm also depends greatly on the choice of similarity measure, the number of related patterns and the combination methods between related patterns. A better technique should be able to distil underlying dynamics from the surface patterns. This leads to non-parametric function approximation methods.

A popular and widely used method is the decision tree algorithm. Decision Trees mimic the human thinking process by formulating learning as a sequence of decision steps from a series of well-designed questions (Reddy and Aggarwal, 2015). The root and interior nodes corresponds to one of the input features in the data. The edges of a node correspond to possible values of the feature corresponding to that node. The leaves of the tree represent the target label (for classification tree) or value (for regression tree). There are many variations of decision trees, for example: ID3, C4.5, C5, and Classification and Regression Trees (CART) (Rokach and Maimon, 2014). In this study, we use the CART algorithm that recursively partitions the feature space based on gini index (Breiman et al., 1984). We then extend this decision tree regression using an ensemble approach as detailed below.

We assume the next-day discharge as a function of historical descriptor vector $\mathbf{x}$. We use each day in the past as a data point, where next-day discharge is the outcome y, and the short-period prior to discharge is used to derive descriptors $\mathbf{x}$. A regression tree approximates a function $f(\mathbf{x})$ by recursively partitioning the descriptor space. At each region R_p, the function is approximated as:

$$f(\mathbf{x}) = \frac{1}{|R_p|} \sum_{x_j \in R_p} y_j$$

where $|R_p|$ is the number of data point falling in region R_p. While regression trees are susceptible to overfitting, random forest is currently one of the most powerful methods to model the function $y = f(\mathbf{x})$ (Breiman, 2001; Hastie et al., 2001b). A random forest is an ensemble of regression trees. The random forest creates a diverse collection of random trees by varying the subsets of data points to train the trees and the subsets of

descriptors at each step of space partitioning. The final outcome of random forest is an average of all trees in the ensemble. Since tree growing is a highly adaptive process, it can discover any non-linear function to any degree of approximation if given enough training data. However, the flexibility makes regression tree prone to overfitting, that is, the inability to generalize to unseen data. This requires controlling the growth by setting the number of descriptors per partitioning step, and the minimum size of region R_p.

The voting leads to great benefits: reduce the variations per tree. The randomness helps combat against overfitting. There is no assumption about the distribution of data, or the form of the function $f(\mathbf{x})$. There is controllable quality of fits. This process also generates a large number of weak learners that control overfitting and produce stable predictions (Schapire, 1990).

Related to random forest is bagging (Breiman, 1996), boosting (Friedman et al., 2000) and randomization (Dietterich, 2000b). Gradient tree boosting (Friedman, 2001) is a high competitive methods with even greater control of flexibility and overfitting. But random forest has the reputation of ease of use and of great prediction quality.

3.2.4.3 Support Vector Regression

The historical descriptor vector x, used in the random forest model can also be used to build a Support Vector Regression (SVR) model (Vapnik, 2013). Given the set of data $(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)$, where each $x_i \in \mathbb{R}^m$ denotes the input descriptor for the corresponding next day forecast $y_i \in \mathbb{R}^1$, a regression function takes the form: $\hat{y}_i = f(x_i)$. Support vector regression works by (i) mapping the input space of x_i into a higher dimensional space using a non-linear mapping function: ϕ (ii) performing a linear regression in this higher dimensional space. In general, we can express the regression function as:

$$f(\boldsymbol{x}) = (\boldsymbol{w}\phi(\boldsymbol{x})) + b$$

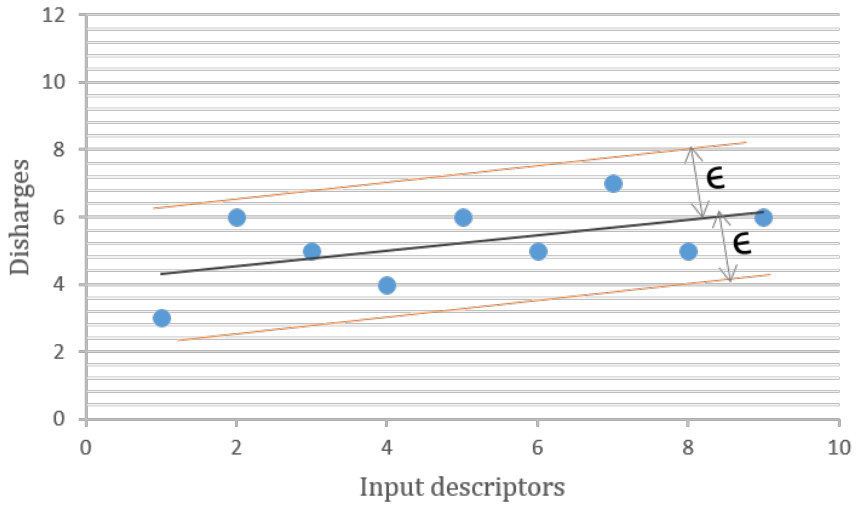


Figure 3.8: The loss function fits a tube of radius ϵ during support vector regression

where $w \in \mathbb{R}^m$is the weights and $b \in \mathbb{R}^1$ is the bias term. Vapnik (2013) proposed the ϵ-insensitive loss function for SVR, which takes the form:

$$\mathcal{L}_\epsilon\left(f(\boldsymbol{x}) - \boldsymbol{y}\right) = \begin{cases} |f(\boldsymbol{x}) - \boldsymbol{y}|, & |f(\boldsymbol{x}) - \boldsymbol{y}| \geq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

The loss function $\mathcal{L}_\epsilon$tolerates tolerates errors that are smaller than the threshold: ϵ, resulting in a "tube" around the true discharge values (see Figure 3.8)

Model parameters can be estimated by minimizing the following cost function:

$$R = C \times \frac{1}{n}\mathcal{L}_\epsilon\left(f(\boldsymbol{x}) - \boldsymbol{y}\right) + \frac{1}{2}||\boldsymbol{w}||^2$$

where C is a constant that penalizes error in training data.

In our work, we use an RBF kernel (Schölkopf et al., 2004) for mapping our input data to higher dimensional feature space. RBF kernels are a good choice for fitting our non-linear discharge pattern because of its ability to map the training data to an infinite dimensional space, and easy implementation. The solution to the dual formulation of SVR cost function is detailed in (Vapnik, 2013; Smola and Schölkopf, 2004).

3.3 Experimental Setting

We extracted all data from the database tables (as in Figure 3.2) for our ward in study. Patient flow was analysed for a period of 5 years. We formatted our data as a matrix where each row corresponds to a day and each column represents a feature (descriptor).

The current hospital strategy involves using past experience to foresee available beds. To compare the efficiency of our proposed approaches, we model the following baselines: (1) Naive forecasting using the last day of week discharge: Studies (Wong et al., 2009; Lin et al., 2011) have shown a strong weekly pattern in daily discharges, we model the next day discharge as the number of discharges for the same day during previous week; (2) naive forecasting using mean of last week discharges: to better model the variation and noise in weekly discharges, we model the next-day discharge as the mean of discharges during previous 7 days; and (3) naive forecasting using mean of last 3-week discharges: to account for the monthly and weekly variations in our data, we use mean of daily discharges over the past 3 weeks to model the next-day discharge.

3.3.1 Evaluation Protocol

Our training and testing sets are separated by time. This strategy reflects the common practice of training the model using data in the past and applying it on future data. Training data consisted of 1460 days from January 1, 2010, to December 31, 2013. Testing data consisted of 365 days in the year 2014. The characteristics of the training and validation cohort are shown in Table 3.3. Most stays were short, around 65% of patients stayed for less than 5 days.

We compare the next-day forecasts of our proposed approaches with the baseline methods on the measures of mean forecast error, mean absolute error, symmetric mean absolute percentage error and root mean square error (Shcherbakov et al., 2013; Hyndman and Koehler, 2006). If y_t is the measured discharge at time t, and f_t is the forecast discharge at time t, we can define the following errors.

	Training (2010-2013)	Testing (2014)
Total days	1460	365
Mean discharges per day	8.47	9.17
Number of admissions	9630	2511
Gender:		
Male	4329 (44.9%)	1135(45.2%)
Female	5301 (55.1%)	1376 (54.8%)
Mean age (years)	63.65	61.62
Length of Stays:		
1-4 days	6377 (66.22%)	1636 (65.15%)
5 or more days	3253 (33.78%)	875 (34.85%)

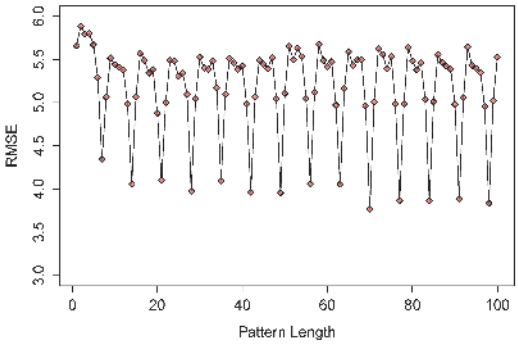
Table 3.3: Training and validation cohorts characteristics.

Mean Forecast Error (MFE): is used to gauge model bias and is calculated as – MFE $=$ mean $(y_t - f_t)$. For an ideal model, MFE = 0. If MFE $>$0 model tends to under-forecast, and if MFE $<$0 , model tends to over-forecast.

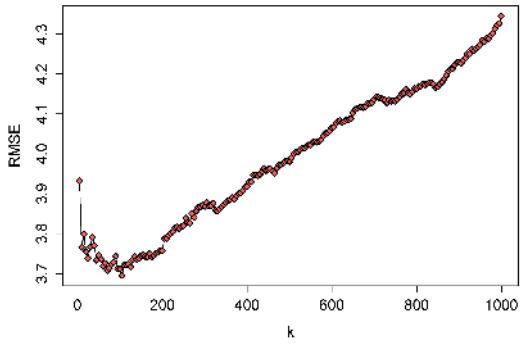
Mean Absolute Error (MAE): is calculated as the average of unsigned errors – MAE $=$ mean$|y_t - f_t|$. MAE indicates the absolute size of the errors. The use of unsigned error terms prevents negative and positive error from offsetting each other.

Root mean square error (RMSE): is a measure of the deviation of forecast errors. It is calculated as – RMSE $= \sqrt{\text{mean}\,(y_t - f_t)^2}$. Due to squaring and averaging, large errors tend to have more influence over RMSE. In contrast, individual errors are weighted equally in MAE. There has been much debate on the choice of MAE or RMSE as an indicator of model performance (Willmott and Matsuura, 2005; Chai and Draxler, 2014).

Symmetric mean absolute percentage error (sMAPE): is an alternative to mean average percentage error (MAPE). It is scale independent and hence can be used to compare forecast performance between different data series. It overcomes 2 disadvantages of mean absolute percentage error (MAPE) namely, (1) the inability to calculate error when the true discharge is zero and (2) heavier penalties for positive errors than negative errors. sMAPE is a more robust estimate of forecast error and is calculated as – sMAPE $=$ mean $(200|y_t - f_t|/(y_t + f_t))$. Also, sMAPE ranges from -200% to 200%,



(a) Forecast error (in RMSE) with changing values of pattern length



(b) Forecast error with changing values of number of nearest neighbours (k)

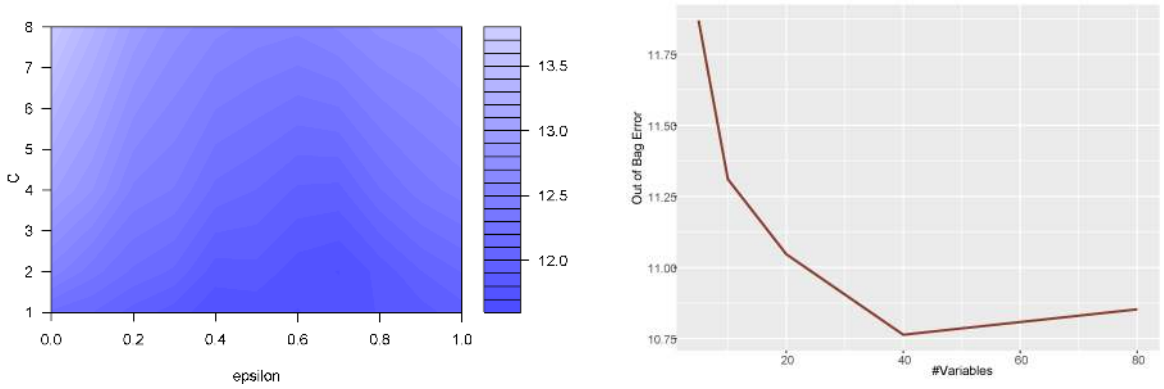
Figure 3.9: Parameter tuning in kNN forecasting.

giving it an ambiguous interpretation (Hyndman, 2006).

3.3.2 Model Implementation

The model parameters for lasso, kNN forecast, RF, and SVR models were tuned to minimize forecast errors. For kNN regression, the optimum value of pattern length: d and number of nearest neighbours: k, was obtained by analysing forecast RMSE for values $d \in (1, 100)$ (see Figure 3.9a) and $k \in (5, 1000)$ (see Figure 3.9b). Minimum RMSE of 3.77 was obtained at $d = 70$ and $k = 125$.

The SVR parameters$C$ (penalty cost) and ϵ (amount of allowed error) were determined by choosing the best value from a grid search, that minimized the model RMSE. This is illustrated in Figure 3.10a.Similarly, the optimum number of variables in building each node of the RF was chosen by examining its effect on minimizing the out-of-bag estimate (see Figure 3.10b). We compared the naive forecasting methods with our proposed approaches using MFE, MAE, RMSE, and sMAPE.



(a) SVR Performance (in RMSE) for different values of C and epsilon. Darker regions imply better performance, and smaller RMSE.

(b) Random forest performance for different number of variables selected in building nodes. Smaller values imply better performance.

Figure 3.10: Parameter tuning for (a) SVR and (b) RF models

3.4 Results

In this section, we examine the performance of each of our models in terms of forecast errors mentioned in Section 3.3.1. We then look at model reproducibility in terms of parameter stability.

3.4.1 Model Performance

The results are summarized in Table , whereas Figure 3.11 compares the distribution of actual discharges with different model forecasts.

The naive forecasts are unable to capture all variations in the data and resulted in the maximum error when compared with other models. The variations in seasonality and trend are better captured in ARIMA and ARMAX models. The time series consisting of past 3-month discharges were used to generate the next-day discharge forecast. The ARMAX model also included the day of week and ward occupancy as exogenous variables, which resulted in better forecast performance over ARIMA.

Interestingly, kNN was more successful than ARIMA and ARMAX in capturing the variations in discharge, demonstrating about 3% improvement in MAE, when compared with ARMAX. However, the kNN model tends to under forecast (MFE = 1.09),

Model	MFE	MAE	sMAPE	RMSE
Naive Forecast using discharge from:				
last weekday	0.03	3.81	45.70 %	4.95
last week (mean)	0.02	3.57	41.68 %	4.42
last 3 weeks(mean)	0.04	3.44	40.14 %	4.34
ARIMA Forecast	0.06	3.27	38.32 %	4.15
ARMAX Forecast	-0.01	2.99	34.86 %	3.84
kNN Forecast	1.09	2.88	34.92 %	3.77
Lasso	0.68	2.75	32.91 %	3.58
CART	0.60	2.77	32.96 %	3.64
Support vector regression	0.73	2.75	32.88 %	3.64
Random forest	0.44	2.70	32.15 %	3.56

Table 3.4: Forecast accuracy of different models

possibly because of resorting to median values for forecast.

In comparison, RF and SVR forecast models demonstrated better performance. This can be expected because they are derived from all the 108 features. However, RF demonstrated a relative improvement of 3.3 % in MAE over SVR model (see Table 3.4). When looking at forecast errors for each day of week, RF model confirmed better performance, as shown in Figure 3.12.

The process of SVR with RBF kernel maps all data into a higher dimensional space. Hence, the original features responsible for forecast cannot be recovered, and the model acts as a black box. Alternatively, RF algorithm returns an estimate of importance for each variable for regression. Examining the features with high importance could give us a better understanding of the discharge process.

The features in random forecast model were ranked on importance scores (see Figure 3.13). The top 10 significant features are described as follows. The day of week for the forecast proved to be the most important feature. Other features were number of patients in the ward during the day of forecast, the trend of discharges measured using locally weighted polynomial regression, number of discharges in past 14th day,

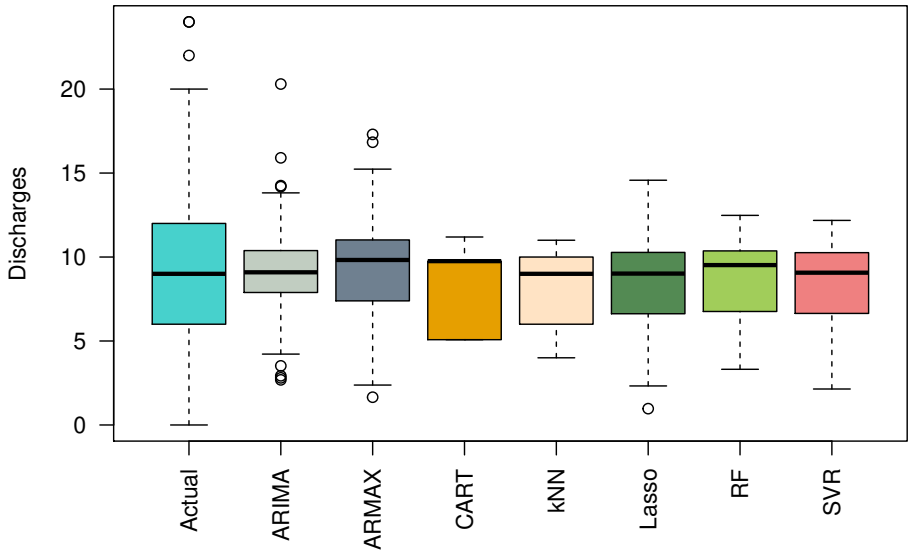


Figure 3.11: Comparison of actual and forecasted discharges from ward for each day in 2014.

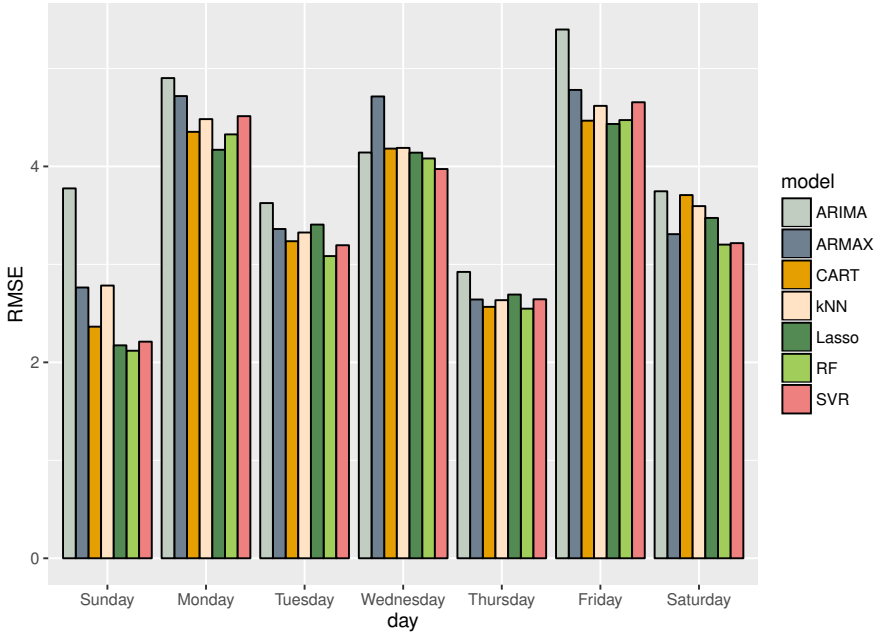


Figure 3.12: Forecast error in predicting each day of week in 2014.

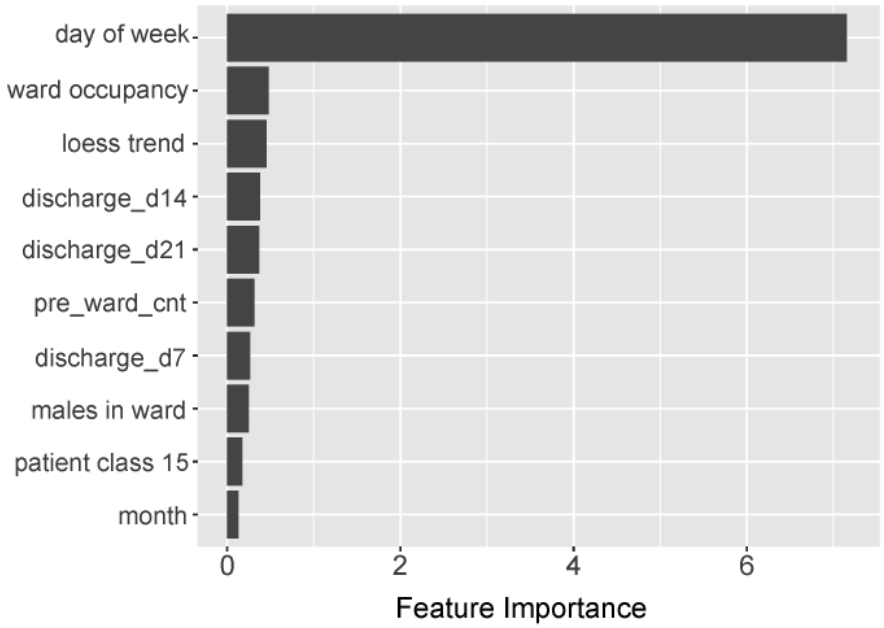


Figure 3.13: Features ranked by importance in the random forest model

number of discharges in past 21st day, number of patients who had visited only one previous ward, the number of males in the ward, number of patients labelled as: "public standard," and current month of forecast.

3.4.2 Assessing Model Stability and Reproducibility

The most interpretable models are ARIMA variations, kNN, lasso regression and decision trees. When we have a host of algorithms with comparable performances, we choose one that is most transparent . Transparency translates to interpretability and repeatability – the model parameters (or predictors; hyperparameters are not considered) should be stable. In the case of patient outflow, we see that lasso and SVR have similar performance. The RF model demonstrates a $1.7\,\%$ improvement. The SVR model using RBF kernel maps the original data into a higher dimensional space, essentially working as a blackbox, and loses all meaning of the original features. The RF model works uses bagging to aggregate the result of an ensemble of decision trees. Hence the reasoning process is inherently random, even though the final prediction is stable. Since the performance of lasso, CART, SVR and RF models are comparable, one would typically choose either lasso regression model or CART. We now investigate the stability

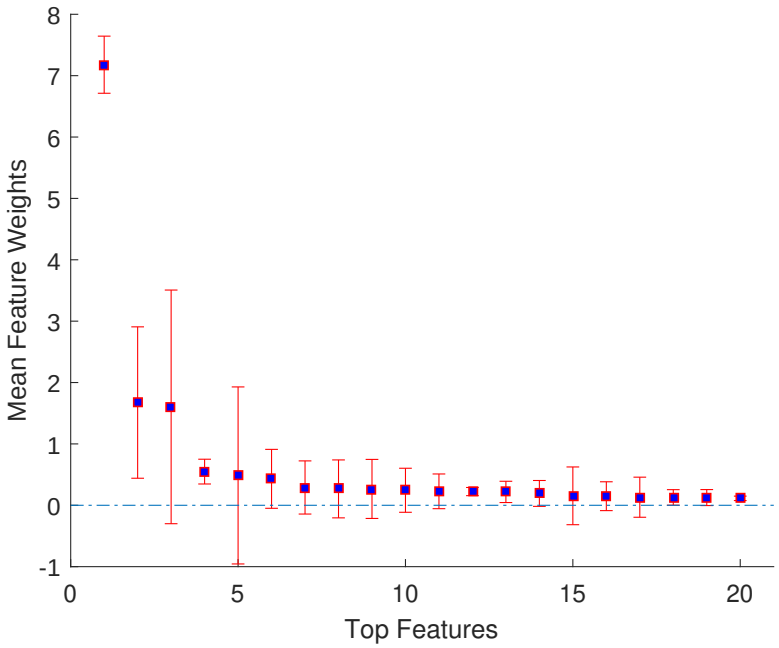


Figure 3.14: Variation in feature weights of top 20 features in lasso regression for modelling patient outflow. Figure plots mean feature weight with standard deviation for 100 bootstraps of training data.

of lasso and CART to judge whether these models are reproducible.

We first perturb the original training data using bootstrap method of sampling with replacement. The lasso model was subjected to 100 bootstraps. The top features selected by lasso exhibited significant variation in weights. This variation is illustrated in Figure. 3.14. We see that, except for the top 2 features, all other weights have a possibility to reduce to zero during a training run. Thus these features can be dropped when the model is re-trained, causing instability in feature sets. We also measured the correlation between feature ranking using Spearman's correlation. The 100 ranked features from lasso model returned a correlation score of .06, indicating that there is minimum correlation between feature ranks selected during each run of lasso.

When looking at CART, each bootstrap of training data resulted in a different decision tree. We have illustrated this in Figure. 3.15 - 2 different tree architecture resulting from two bootstraps of our reduced training set. The instability of decision tree algorithm - producing significantly different hypothesis from training sets that vary slightly - have been well studied in literature (Turney, 1995; Li and Belford, 2002; Dwyer and Holte,

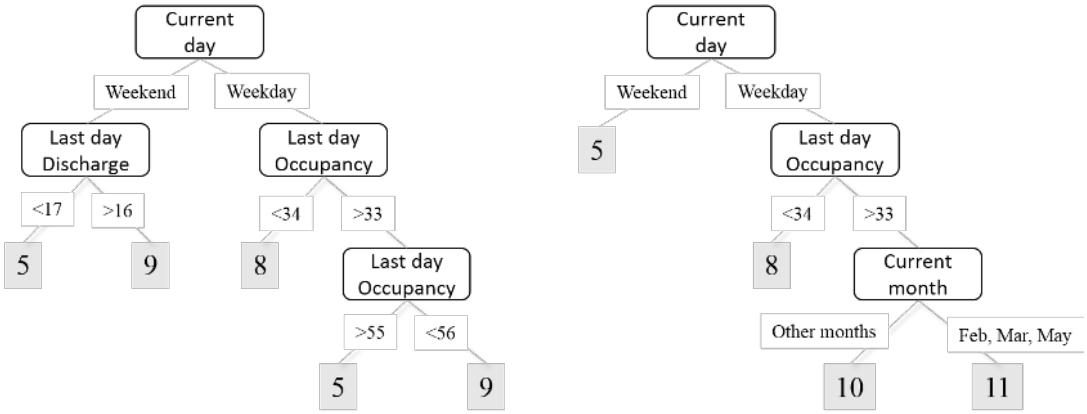


Figure 3.15: Decision trees resulting from two different bootstraps of training data with reduced set of features.

2007). Turney (1995) when studying the effects of yield from a manufacturing process using decision trees noted:"*The engineers are disturbed when different batches of data from the same process result in radically different decision trees. The engineers lose confidence in the decision trees, even when we can demonstrate that the trees have high predictive accuracy.*" Also, deep decision trees have a tendency to overfit, due to their low bias and high variance. This high variance is reduced in random forests by bagging an ensemble of decision trees. However, the bagging process introduces bias and loss of interpretability (Hastie et al., 2001b). This is because aggregating methods such as bagging are designed to stabilize predictions and ignore decision rules. In this process, it becomes difficult to interpret the exact rules for a given prediction (Li and Belford, 2002).

3.4.3 Sources of Instability

From our experiments, we see that the best performing models: Lasso, SVR and RF, are not reproducible since they are inherently unstable. We had detailed the causes of instability in Section 2.2.2.2. Applying those principles to our case study, we make the following observations.

RF and SVR (with RBF kernel) models are unstable by model design. The primary aim of these models is to reduce generalization error and improve performance. Interpretability and model reproducibility is overlooked when deriving such models. For example,

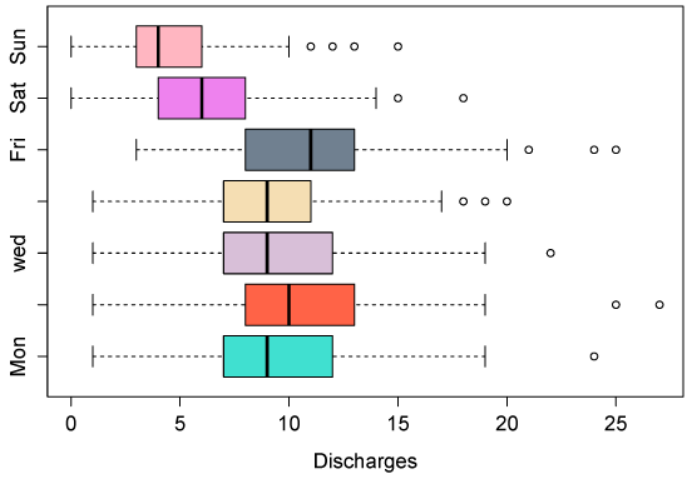


Figure 3.16: Distribution of discharges per day.

RF model is built from an ensemble of randomly sampled decision trees during bootstrap. Each node in the forest is grown using a random input or combination of inputs. This randomness minimizes the data correlation and helps increase accuracy (Breiman, 2001). Hence, different training run results in a different RF model, although accuracy remains the same. On the other hand, the SVR model uses an RBF kernel which transforms the input vectors into an infinite dimensional space. The features lose their physical meaning, and the model loses interpretability and reproducibility.

Linear models are preferred for clinical prediction due to their interpretability and reproducibility. However, the lasso model in our case study displayed significant instability due to the nature of medical data. The discharge data extracted from patient records was characterized by: (i) variation in data, and (ii) correlation in data.

Variation in training data could lead to unstable models. Patient length of stay is inherently variable, partly due to the complex non-linear structure of medical care (Harper and Shahani, 2002). In this study, we have used administrative data from a hospital database. Such data is often characterized by variations in recording, redundancy and irregularities. This could be attributed to management of hospital processes such as ward rounds, inpatient tests, and medication. The non-linear nature of these processes contributes to unpredictable length of stay even in patients with similar diagnosis. The number of discharges from a ward is strongly related to the length of stay of the current patients in the ward. Hence, the variability in ward-level discharges is compounded by the variability in individual patient length of stay. In our study, the daily discharge pat-

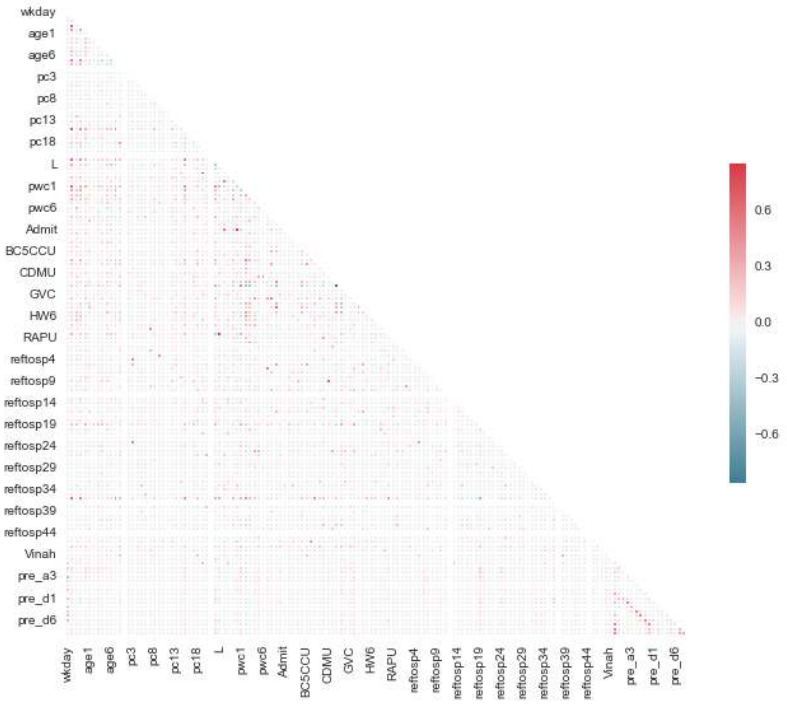


Figure 3.17: Correlations among features of patient flow data.

tern from ward shows great variation for each day of week, as illustrated in Figure. 3.16.

Further, correlations among features can also lead to model instability in sparse linear models and decision trees. The correlation plot for our data in Figure. 3.17 reveal significant feature interactions. We must consider such interactions during learning stage to stabilize our model.

3.5 Discussion

This chapter set out to explore the challenges in obtaining a prediction model from hospital data that is interpretable and reproducible. Sparsity promotes interpretability while stability ensure reproducibility. We compared three linear and four non-linear approaches to modelling next-day ward discharges. Our case study highlights the following observations. First, high performing models such as random forests and support vector regression are highly non-linear, complex and are not interpretable. The SVR kernel maps the features into a higher dimensional space during the regression process.

Hence, the physical meaning of the features is lost, making it difficult to interpret the model. The nonlinear SVR kernel computes similarities between data points. With lots of features, it is close to impossible to judge the features responsible for prediction. In fact, features are often treated equally in kernels, making it difficult to assess feature contributions.

By design, random forests are unstable in parameters and it is difficult to trace the exact reasoning behind predictions. RFs are based on hundreds of decision trees, each of which is randomly generated. Hence it is very difficult to quantify the contribution of each tree, and the variables. Although relative variable importance can be computed, we still cannot quantify the effect on output.

One might argue that model reproducibility and generalization is more important than performance (Haury et al., 2011; Johansson et al., 2011; De Bock and Van den Poel, 2012). Though model accuracy and discrimination are important, for decision support and care management, a model also needs to be stable in choosing the risk factors and weights associated with its predicted risk score. The factors selected by the model are often subjected to further analysis and study to understand the underlying causes of disease and modify patient intervention. Hence, when the selected risk factors change during each training run, they lose validity and the model cannot be clinically accepted (Saeys et al., 2008; Zhou et al., 2013).

3.6 Conclusion

The patient flow case study employed administrative data recorded over 5 years. The feature extraction process was simple since the data contained numeric variables as ward admission/discharge statistics and categorical variables such as types of wards visited. These statistics were aggregated into distributions for each day.

For the rest of the thesis, we will look at more complex scenarios where data includes both administrative and clinical information. Our focus will be to stabilize readmission models derived from electronic medical records (EMR). The features in such data vary over time requiring advanced more sophisticated feature extraction. We will be dealing with patient cohorts with almost twice the number of features than sample size. Patient

data will have missing entries, and will be of variable length. The clinical data will also exhibit high correlation among features due to related diagnosis, co-occurring diseases and related medical procedures.

A small number of succinct features offer better interpretability and stability of these features ensure reproducibility. To this end, we will focus on sparse linear models for clinical prediction. We use the two most popular models in medicine because of their ease of interpretability: logistic regression and cox regression. To ensure sparsity, we use lasso regularization. Our case study, in line with similar studies (Austin and Tu, 2004; Ng, 2004; Lin and Lv, 2013), illustrated that automatic variable selection using lasso leads to unstable models. In the following chapters, we illustrate strategies to overcome instability.

3.6.1 Stabilisation Strategies

We now present a general framework for stabilizing lasso models. We shall use this framework throughout our thesis to suggest extensions to lasso regularization to ensure model stability. We formulate the framework as follows.

Sparse generalized linear models take the form $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$ subject to $\sum_{i=1}^{N} |w_i| \leq \alpha$, where $w \in \mathbb{R}^N$ is the model parameter derived from data: $\boldsymbol{x} \in \mathbb{R}^N$. Here, α is the sparsity controlling parameter, typically enforced using lasso regularization (Tibshirani, 1996). More formally, let $\mathcal{D} = \{\boldsymbol{x}_m, y_m\}_{m=1}^{M}$ denote the training data, where $\boldsymbol{x}_m \in \mathcal{R}^N$ denotes the high dimensional feature vector of data instance m, and y_m is the outcome (for example, the occurrence of future readmission). If $\mathcal{L}(\boldsymbol{w}|\mathcal{D})$ is a linear loss, we propose a stability component $R_D(\boldsymbol{w})$ to modify lasso regularization as:

$$\mathcal{L}_{\text{loss}} = \frac{1}{M}\mathcal{L}\left(\boldsymbol{w}|\mathcal{D}\right) + \alpha\sum_{i}^{N}|w_i| + R_{\mathcal{D}}(\boldsymbol{w}) \tag{3.4}$$

where $\alpha > 0$ is the penalty controlling the sparseness of the feature weights. Under lasso, weights of weak features are driven towards zeros, and thus the resulting model is sparse. The stabilization term $R_{\mathcal{D}}(\boldsymbol{w})$ ensures statistical sharing of feature weights. In the following chapters, we explore the following ways in formulating $R_{\mathcal{D}}(\boldsymbol{w})$. Specific-

ally, we use three strategies:

- Strategy I (detailed in Chapter 4): We exploit the inherent domain semantics to strengthen data-driven findings. We use domain knowledge, specifically the temporal nature of events and the hierarchical nature coding schemes to formulate $R_{\mathcal{D}}(\boldsymbol{w})$. Here, $R_{\mathcal{D}}(\boldsymbol{w})$ is not data dependant.
- Strategy II (detailed in Chapter 5): We use data-driven techniques by deriving statistical feature relations to formulate $R_{\mathcal{D}}(\boldsymbol{w})$. We look at the most popular statistical measures for $R_{\mathcal{D}}(\boldsymbol{w})$ and compare the effects on stability.
- Strategy III (detailed in Chapter 6): Finally, we exploit higher-order regularities and use the principles of self-taught learning to derive $R_{\mathcal{D}}(\boldsymbol{w})$.

We proceed to discuss the first strategy in the following chapter.

"Dont let them change you. Or even re-arrange you."

Bob Marley, "Could you be loved"

Chapter 4

Stabilization I: Knowledge-Driven

STABILITY promotes reliability – in performance, estimation, or interpretability. Our previous chapter demonstrated instability in a simple clinical prediction model derived on few hundred features from administrative data. We concluded our case study by proposing a stability component $R_{\mathcal{D}}(\boldsymbol{w})$ (as in (3.4)) to modify the sparse feature selection using lasso. In this chapter, we present our first model stabilization strategy using domain knowledge to strengthen data-driven model discovery. Similar to the case of flow forecasting in the previous chapter, our primary data source is Electronic Medical Records (EMR). But we consider an even more challenging setting: for similar data size, we now have thousands of features (instead of hundreds). In lieu of well-designed features, we call for semi-automatic feature extraction methods.

In the following chapters, we use both administrative and clinical information of patients from hospital EMR records. EMR data is temporal, strongly correlated and high dimensional (He et al., 2013). Each of these aspects poses significant challenges to data extraction and model building. High dimensional data calls for sparsity inducing feature selection (Ye and Liu, 2012). However, automatic feature selection, particularly in clinical data, has been known to cause instability in features resulting in non-reproducible models (Austin and Tu, 2004). This problem is further aggravated by strong correlations in EMR data. Sparse models often pick the strongest features from the chosen sample-set (Zou and Hastie, 2005). Under data re-sampling, an alternate feature from the correlated pair could be selected causing significant variations to the

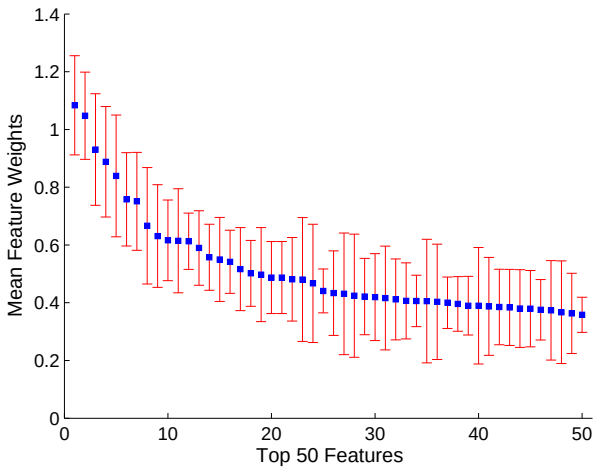


Figure 4.1: Feature instability due to data resampling. Mean weights vs standard deviation for the top 50 features selected by a lasso-regularized logistic regression model under bootstraps.

feature weights during each training run (Xu et al., 2012). This problem is illustrated in Fig. 4.1 – the mean weights of the top 50 predictors from routine EMR data for 6 months readmission due to heart failure is shown. The top predictors selected by lasso-regularized model (Tibshirani, 1996) have large variance in feature weights under bootstraps (see Fig. 4.1) - thus rendering them unusable in a clinical setting.

Here, we proceed to model $R_{\mathcal{D}}(\boldsymbol{w})$ using two key observations in medical domain. First, events that occur during consecutive time periods can be related. Second, diagnosis and procedure codes display a hierarchical nature, where codes that share the same prefix belong to the same category. For prognosis, we use logistic regression model for 6 month readmission after heart failure - a deadly and costly disease, with majority of patients returning within a year after discharge. The main contributions of this chapter are summarized as follows:

1. We use temporal relations in events and the hierarchical nature of diagnosis codes to construct a feature graph that encapsulates temporal and semantic correlations among features in patient records.

2. Using this knowledge driven feature graph, we apply a graph Laplacian regularizer to model $R_{\mathcal{D}}(\boldsymbol{w})$ in (3.4). The graph Laplacian encourages pairwise similarity among related features.

3. Our proposed methodology demonstrates improved feature selection stability as measured using Consistency index and Jaccard index, and improved model estimation stability as measured by Signal-to-Noise ratio.

4.1 EMR Data Extraction and Challenges

The first step in building a clinical prediction model is extracting features from the hospital database. For all experiments in this thesis, we collected data from Barwon Health, a regional health service provider in Victoria, Australia. The provider has been serving more than $350,000$ residents. Ethics approval was obtained from the Hospital and Research Ethics Committee at Barwon Health (number 12/83) and Deakin University. We also obtained written consent from patients in storing and using their information for research.

Patient details were stored in EMR databases. We were provided a single point of access to query patient records from the database of the hospital. For our study, we collected the retrospective data of heart failure patients via this access. The resulting cohort contains $1,405$ unique patients with $1,885$ admissions between January 2007 and December 2011. We identified patients with heart failure if they had at least one ICD-10 diagnosis code I50 at any admission. Patients of all age groups were included whilst inpatient deaths were excluded from our cohort. Among these patients, 49.3% are male and the median age is 81.5 at the time of admission. We focused our study on emergency attendances and unplanned admissions of patients. The readmission of patients was defined as an admission within the horizons of 1, 6 and 12 months after the prior discharge date.

4.1.1 Multi-granular Feature Extraction

A typical EMR consists of demographic information (e.g., age, gender and postcode) and time-stamped events (e.g., hospitalizations, ED visits, clinical tests, diagnoses, pathologies, medications and treatments). It includes International Classification of Disease

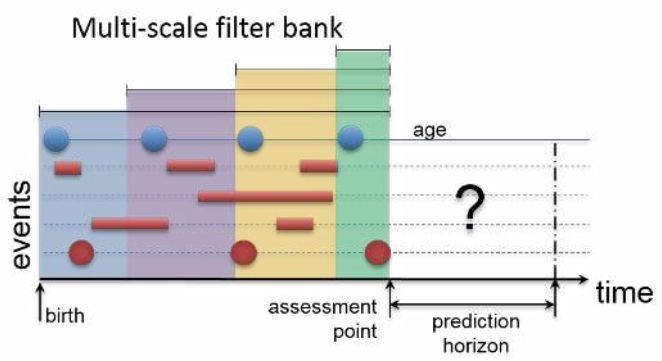
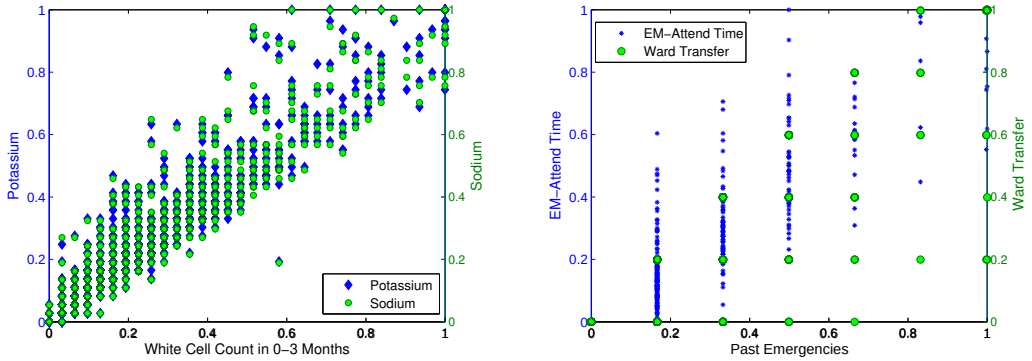


Figure 4.2: An illustration of patient clinical events (as red) over time, which is convoluted using one-sided filter bank. Adapted from (Tran et al., 2013)

10 (ICD-10) scheme[1], Australian invention coding (ACHI) scheme[2], Diagnosis-Related Group (DRG) codes, detailed procedures and discharge medications for each admission and ED visit. Our feature extraction process from EMR transforms inpatient time-stamped events into a high-dimensional feature vector at index discharge. Such events can be hospitalizations, clinical tests, diagnoses and treatments. For example, the presence of an ICD code can be considered as an event. For patient demographics, some events are time invariant (name, gender). For demography information as postcode, a change in such information is treated as an event. Patient age can be divided into categories or bands and a change is recorded (event) when the patient age moves from one category to the other. Finally, for continuous events (for example: treatment episodes), we model the event as the duration of that entire episode. A representation of such patient history is illustrated in Figure 4.2.

Extracting features from such data presents several problems. The challenges are that recorded events are sparse and irregular. As diseases progress in different paces, it is important to take multiple time scales into account. In addition, recent critical events carry more weight than mild conditions observed far back in the history. To this end, we employ the *one-sided convolutional filter bank* recently introduced in (Tran et al., 2013). The filter bank summarizes event statistics over multiple time periods and granularities: (0-3), (3-6), (6-12), (12-24), (24-48), (48-72) months.

[1]http://apps.who.int/classifications/icd10
[2]http://www.aihw.gov.au/procedures-data-cubes



(a) (b)

Figure 4.3: Example of correlations in EMR data: (a) clinical correlations (b) correlations among administrative events

4.1.2 Challenges for Model Stability

Instability in clinical prediction is largely due to the nature of data stored in hospital databases. In this section, we briefly look at the nature of data extracted from our hospital database. The primary purposes of EMR are setting objectives, planning patient care, documenting the delivery of care and assessing the outcome of care (Häyrinen et al., 2008). A typical EMR contains unstructured narrative text, structured coded data, and time stamped events. With so much diverse information, the quality of data recorded is also important (Thiru et al., 2003). The nature of data and recording process contributes to model instability due to the following reasons.

First, there is a possibility of data redundancy – the diseases, interventions, medications may be recorded in more than one way. Also, some patient records may have incomplete entries. The quality of data recording may be poor resulting in lack of precise information. Finally, the most common cause of instability is correlation among features. EMR data is characterized by high correlation among clinical and administrative events . For example, emergency admission events will be correlated with ward transfers, diagnosis of co-occurring diseases (heart failure and diabetes) will have high correlation, pathological measurements (amount of Sodium and Potassium in the body) will be related. This is illustrated in Figure. 4.3.

4.2 Feature Graph Construction

In this section, we present our first technique to stabilize lasso in the presence of high-dimensional correlated data. To ensure correlated features are selected together, we resort to regularized learning with network of features. We propose to construct this feature network using prior domain knowledge about which features are correlated and therefore should result in similar weights ($\boldsymbol{w}$ in (3.4)). To this purpose, we construct a feature graph with nodes as features and edges representing feature similarity.

Additional regularization of the sparse learning model by $R_{\mathcal{D}}(\boldsymbol{w})$ penalizes each w_i by the amount it varies from the average weight of its neighbouring feature weights. Similar to recent methods for incorporating domain knowledge for regularization, our technique can be viewed as constructing a Gaussian prior with non-diagonal covariance matrix on model parameters:$\boldsymbol{w}$ (Krupka and Tishby, 2007; Sandler et al., 2008; Li and Li, 2008). However, the covariance matrix is induced from a network.

To construct this network, we exploit two inherent feature associations in patient records: association in time, and association among diagnosis codes. We detail each of these in the following sections.

4.2.1 Temporal Structures

First, we look at temporal associations in features. Most events (clinical and administrative) in patient records are temporal. Some events such as heart attacks occur for a short amount of time, where as other events such as presence of comorbidities can be long-term. Thus our feature extraction process takes multiple time scales into account (as described in Section 4.1.1). The feature extraction process results in patient events summarized over multiple time scales. Hence when we have identical events over consecutive time periods, we consider them to be related and propose these events should have similar weights.

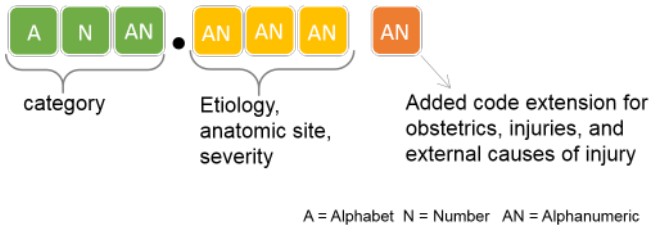


Figure 4.4: Format of ICD-10 code

4.2.2 Hierarchical Structures

The second inherent association in patient records is due to the nature of diagnosis codes. An important component in patient medical records are the diagnosis codes. These codes identify the reasons for patient encounter, such as: type of diseases, disorders, symptoms and injuries. Our data confirms with the latest coding standard of ICD-10, which supports over $16,000$ codes for describing patient condition. These codes follow a logical hierarchy for classification purposes. The general structure of ICD-10 coding scheme is as follows. Each code can consist of three to seven characters. They could be alphabets, numbers or alphanumeric. The first three characters represent the category of patient encounter, followed by a decimal point. All following characters identify the specific details of patient encounter. Figure. 4.4 illustrates this structure. The coding scheme follows a defined hierarchical structure, with additional characters used to resolve the finer specifications of patient encounter. For example, the ICD-10 codes for types of injuries to elbow and forearm is as below:

S50–S59	Injuries to the elbow and forearm
S52	Fracture of forearm
S52.5	Fracture of lower end of radius
S52.52	Torus fracture of lower end of radius
S52.521	Torus fracture of lower end of right radius
S52.521A	Torus fracture of lower end of right radius, initial encounter, closed fracture
S52.6	Fracture of lower end of both ulna and radius

Hence all diagnosis codes prefixed with $S5$ are related to elbow and forearm injuries. We can exploit this hierarchical format to construct a feature network of patient diagnosis codes.

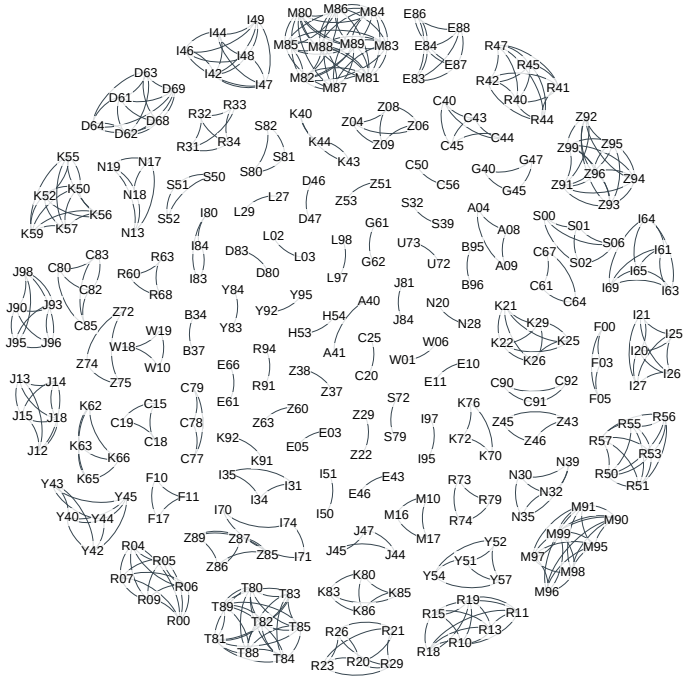


Figure 4.5: Grouping ICD-10 diagnosis codes in 1885 heart failure patients, using hierarchical coding relations. Diagnosis codes with the same 2 letter prefix are connected together.

4.2.3 Constructing $R_{\mathcal{D}}(\boldsymbol{w})$

We propose to build a feature graph by identifying connections between features that observe temporal (as in Section 4.2.1) and structural relations (as in Section 4.2.2). Two features are connected if they satisfy one of the following conditions. The first condition checks if the codes are identical and the periods are consecutive. This represents the disease progression over the time, for example, from the period of 3-6 months to the period of 0-3 months before the discharge. The second condition inspects hierarchical relationship in diagnosis codes – the periods are identical and the codes share the first two characters. This captures the diagnostic or therapeutic relations. For instance, two related features are the ICD-10 code *I25* (chronic ischaemic heart failure) and *I21* (acute myocardial infarction). The grouping of ICD-10 diagnosis codes for a heart failure cohort of 1885 patients is shown in Figure 4.5.

These groupings can be considered as prior knowledge derived from the nature of medical domain, and can be used to construct a feature graph that models $R_{\mathcal{D}}(\boldsymbol{w})$ as follows. Let $\boldsymbol{A} \in \mathbb{R}^{p \times p}$ be the incident matrix of the feature graph, i.e., $A_{ij} = 1$ if feature i and

j are related and $A_{ij} = 0$ otherwise. Sharing statistical strength between any two related features is realized by enforcing the similarity in their weights. We model the graph-regularizing term $R_{\mathcal{D}}(\boldsymbol{w})$ in (3.4) as:

$$R_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{2}\beta \sum_{ij} A_{ij} \left(w_i - w_j\right)^2 \tag{4.1}$$

where $\beta > 0$ is the correlation coefficient controlling the effect of the graph-based regularization. The graph-regularizer in (4.1) can be simplified as: $\frac{1}{2}\sum_{ij} A_{ij} \left(w_i - w_j\right)^2 =$

$$\begin{aligned} &= \sum_i \left(\sum_k A_{ik}\right) w_i^2 - \sum_i \sum_j A_{ij} w_i w_j \\ R_{\mathcal{D}}(\boldsymbol{w}) &= \boldsymbol{w}'\boldsymbol{L}\boldsymbol{w} \end{aligned} \tag{4.2}$$

where $\boldsymbol{L}$ is the Laplacian matrix of feature graph $\boldsymbol{A}$, i.e., $L_{ii} = \sum_j A_{ij}$ and $L_{ij} = -A_{ij}$ (Chung, 1997).

In (4.2), $R_{\mathcal{D}}(\boldsymbol{w})$ is a Laplacian regularizer that penalizes each edge of feature graph $\boldsymbol{A}$ equally. This formulation of $R_{\mathcal{D}}(\boldsymbol{w})$ combats the instability in several ways. First, features of the same type tend to cluster, and thus their weights are more difficult to vary as a whole. Weaker features can borrow the statistical strength from the stronger ones. Second, two strongly correlated features must either be selected or jointly suppressed by the lasso.

We proceed to apply this regularization scheme for readmission prediction of heart failure cohort using logistic regression. The following section explains our model framework.

4.3 Model Framework

We apply our proposed knowledge driven stabilization to the task of predicting heart failure readmission in 6 months as follows. Our framework consists of a training phase

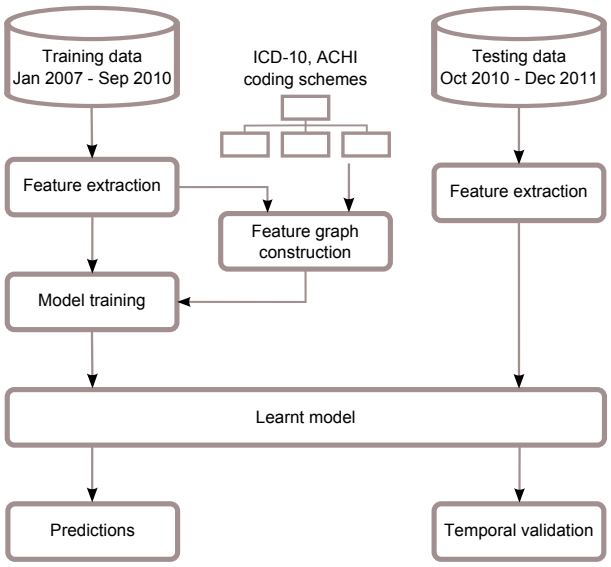


Figure 4.6: The workflow diagram of the framework for deriving graph-stabilized prediction models from Electronic Medical Records. Temporal feature relations and coding hierarchies were used to construct the feature graph (Fig. 4.8).

using data from the past and a validation phase using new admission data from the future (Fig. 4.6 for the workflow diagram). Our model development consists of three sub-phases: (i) multi-granular temporal feature extraction (as detailed in Section. 4.1.1) (ii) feature graph construction based on the temporal relations and coding hierarchies (as detailed in Section. 4.2.2), and (iii) model training with feature selection and feature graph regularization.

We use sparse logistic regression to model readmission in 6 months. We illustrate this by revisiting (3.4), as follows. Let $\mathcal{D} = \{\boldsymbol{x}_m, y_m\}_{m=1}^{M}$ be the training dataset in which $\boldsymbol{x}_m \in \mathbb{R}^N$ denotes the high-dimensional feature vector of data instance m and $y_m \in \{0, 1\}$ is the binary outcome (where 1 indicates the occurrence of future readmission). Our aim was to model the predictive distribution $P\left(y_m \mid \boldsymbol{x}_m; \boldsymbol{w}\right)$ where $\boldsymbol{w} \in \mathbb{R}^N$ are feature weights. Hence the loss function $\mathcal{L}\left(\boldsymbol{w}|\mathcal{D}\right)$ in (3.4) becomes the logistic loss function (as detailed in Section 2.1.5.2), while the stabilization scheme $R_{\mathcal{D}}(\boldsymbol{w})$ is as given in (4.2). Our final model can be written as:

$$\mathcal{L}_{\text{loss}}\left(\boldsymbol{w}|\mathcal{D}\right) = \frac{1}{M}\mathcal{L}_{\text{logit}}\left(\boldsymbol{w}|\mathcal{D}\right) + \alpha\sum_{i}^{N}|w_i| + \frac{1}{2}\beta\boldsymbol{w}'L\boldsymbol{w} \tag{4.3}$$

The objective function in (4.3) is convex (Wainwright et al., 2007; Lee et al., 2006; Boyd

and Vandenberghe, 2004). We applied the L-BFGS algorithm (Liu and Nocedal, 1989) for parameter estimation.

4.4 Data and Validation

Our model and baselines were derived on the heart failure cohort introduced in Section 4.1. All models were built from a training set and validated on a testing set. The training and testing data were separated in time. In other words, all models were externally validated in time. Patients discharged prior to 1st September 2010 were used for training, and a separate set of those discharged afterwards for testing (see Fig. 4.7). This validation strategy was chosen because it better reflects the common practice of training the model in the past and using it in the future. According to Altman et al. (2009), even though there are similarities in clinical techniques for patients in training and testing cohort, the testing data is independent of the data and process on which the model was derived.

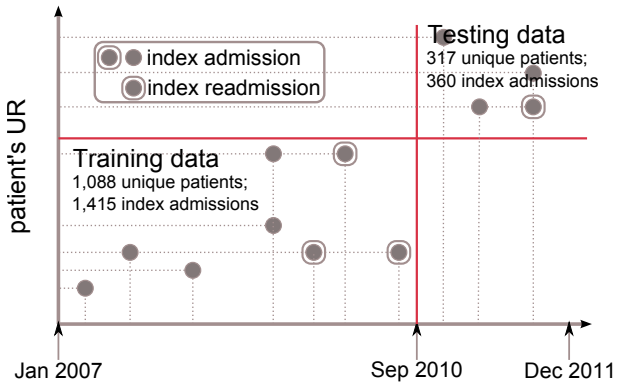


Figure 4.7: Training and test data: Time of hospitalization (x-axis) and unique patient id (y-axis), showing patient and temporal split. The temporal split of training and test data is made on 1st September 2010. The test and training set are disjoint in chosen patients.

Model performance was evaluated using measures of sensitivity (recall), specificity, precision, F-measure and AUC (area under the ROC curve) with confidence intervals based on Mann-Whitney statistic (Birnbaum et al., 1956). We used a predefined threshold to predict readmissions. The value of the threshold was chosen to maximize the F-measure computed from the training data. The details of the training and

	Derivation	Validation
Number of admissions	1415	369
Unique patients	1088	317
Gender:		
Male	541 (49.7%)	155 (48.9%)
Female	547 (50.2%)	162 (51.1%)
Mean age (years)	78.3	79.4
Length of Stays:		
1-4 days	668 (61.4%)	209 (65.9%)
5 or more days	420 (38.6%)	108 (35.1%)

Table 4.1: Training and validation cohorts characteristics.

validation cohort are shown in Table 4.1.

The stability of selected feature subsets from different regularized models were measured using Consistency index (Section 2.2.3.2) and Jaccard index (Section 2.2.3.2). Correspondingly, the stability in feature weights were measured using Signal-to-Noise ratio (Section 2.2.3.2)

4.4.1 Ranking Features by Importance

Features were ranked by their importance. For each feature, importance was calculated as the product of its weight and the standard deviation in the training data, as in Friedman and Popescu (2008). We normalized the feature importance measures in the range of [0,100].

4.5 Experiments and Results

Our proposed model was trained using the training data and validated on the validation cohort (as given in Table 4.1) for goodness-of-fit and model stability. The feature extraction process (Sec. 4.1.1) resulted in $3,338$ features. The lasso-regularized regression model (Sec. 3.3) resulted in 142 risk factors which were positively predictive of unplanned rehospitalization following heart failure discharges.

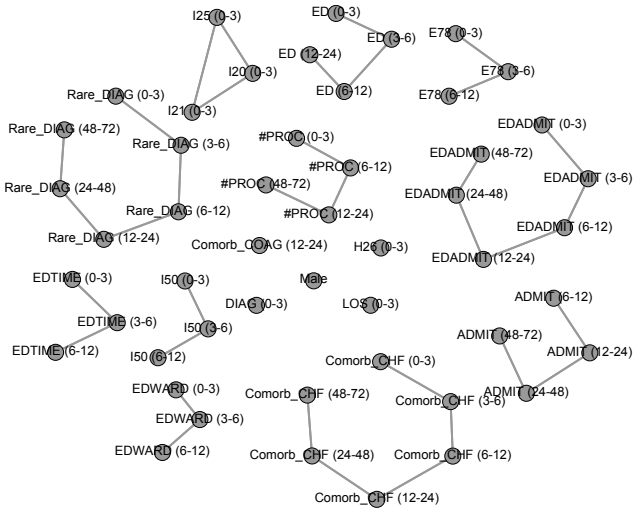
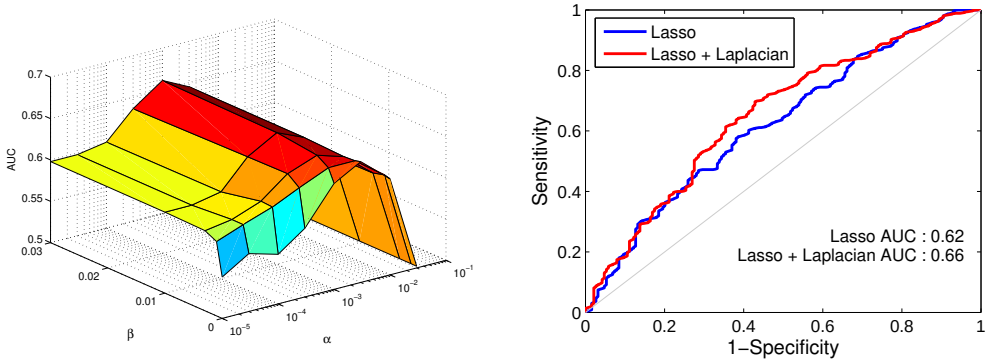


Figure 4.8: Feature sub-graph of top risk factors. Numbers in brackets are time intervals, measured by months, before the index discharges. Factors selected are: *Male*; recent length of stay (*LOS*)*;* heart failure (*I50*, *Comord_CHF*); recent ischaemic heart diseases *(*angina pectoris (*I20*), acute myocardial infarction (*I21*), chronic ischaemic heart disease (*I25*)); any time rare diagnoses (*Rare_DIAG*); time stayed in emergency department (*EDTIME*); frequencies of emergency attendance (*ED*), unplanned admissions (*EDWARD*, *EDADMIT*), admissions (*ADMIT*), diagnoses (*DIAG*) and procedures (*#PROC*); and disorders of lipoprotein metabolism (*E78*).

Graph-based regularization (Sec. 4.2) resulted in sub-graphs being selected as a whole, as shown in Fig. 4.8. The question is how does it affect model performance and feature stability against data resampling?

4.5.1 Model Performance

The model performance was measured for different values of the lasso regularization term α and the Laplacian regularization term β. Table 4.2 reports other measures (sensitivity, specificity, precision, F-measure and AUC). Overall, the discriminative measures were not sensitive of the Laplacian factor β but depended critically on the lasso factor α. Fig. 4.9a displays the AUC in finer details for α. A good discrimination was achieved at $\alpha = .001$ and $\beta = .01$, where external validation resulted in an AUC of 0.66 (95%, CIs: $[0.6, 0.71]$). For the validation cohort, the Laplacian stabilized model was able to detect more true readmissions (sensitivity = 42.22 %) than lasso regularized model (sensitivity = 38.33 %). The overall classification accuracy for Laplacian stabilized model was 59.6% as opposed to 57.9 % for lasso regularized model.



(a) Effect of graph stabilized lasso regularization on area under the ROC curve (AUC). $\beta = 0$ reduces to the baseline lasso.

(b) Comparing ROC plots.

Figure 4.9: Effect of graph stabilization on model performance.

Hyperparam.	Sens./Rec.	Spec.	Prec.	F-Meas.	AUC
$\alpha = \beta = 0$	0.49	0.59	0.54	0.51	0.54
$\alpha = .001$					
$\beta = .00$	0.41	0.79	0.62	0.51	0.62
$\beta = .01$	0.42	0.79	0.62	0.51	0.66
$\beta = .03$	0.44	0.76	0.66	0.53	0.66
$\alpha = .002$					
$\beta = 0.0$	0.49	0.73	0.66	0.55	0.65
$\beta = .01$	0.49	0.73	0.65	0.55	0.65
$\beta = .03$	0.48	0.72	0.62	0.54	0.64
$\alpha = .003$					
$\beta = 0.0$	0.46	0.76	0.64	0.54	0.62
$\beta = .01$	0.46	0.76	0.64	0.54	0.62
$\beta = .03$	0.45	0.75	0.63	0.53	0.62
$\alpha = .004$					
$\beta = 0.0$	0.44	0.77	0.66	0.53	0.63
$\beta = .01$	0.44	0.77	0.66	0.53	0.63
$\beta = .03$	0.43	0.78	0.65	0.52	0.63
$\alpha = .005$					
$\beta = 0$	0.46	0.81	0.69	0.55	0.63
$\beta = .01$	0.46	0.81	0.69	0.55	0.63
$\beta = .03$	0.45	0.82	0.69	0.55	0.63

Table 4.2: The performance of model for various settings of lasso regularization term (α) and Laplacian regularization term (β) after model averaging from 50 bootstraps.

Hosmer-Lemeshow test			
Model regularization	χ^2	**df**	**Significance**
Lasso	26.50	8	.0009
Lasso + Laplacian	7.23	8	.513
Elastic Net + Laplacian	6.25	8	.619

Table 4.3: Measuring goodness-of-fit for logistic regression (df = degree of freedom). Small χ^2 values with large significance ($p > .05$) indicate better fit.

4.5.1.1 ROC Curve Analysis

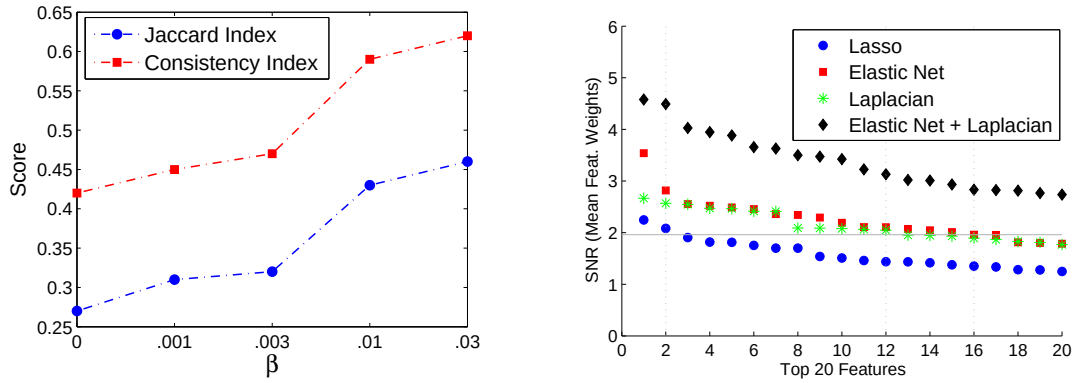
The area under the ROC curve (AUC or c-statistic) can be used to compare different models fitted to the same data. As shown in Fig. 4.9b, the application of Laplacian stabilization marginally improved the AUC over the lasso model. However a combination of elastic net and Laplacian was not able to improve the model discrimination.

4.5.1.2 Goodness-of-fit Statistics

We now compare the goodness-of-fit of models using Hosmer-Lemeshow (HL) test statistic. We divided our validation cohort into 10 groups defined by increasing order of estimated risk. Nine groups contained 37 observations, while one group contained 36. The expected frequencies in each group was more than five. Hence all conditions for reporting the HL test statistic was met (Peng et al., 2002). Both Laplacian and combination of elastic net and Laplacian regularization resulted in small values of HL test statistic with $p > .05$ suggesting that these models fit the data quite well (see Table 4.3).

4.5.2 Stability against Data Re-sampling

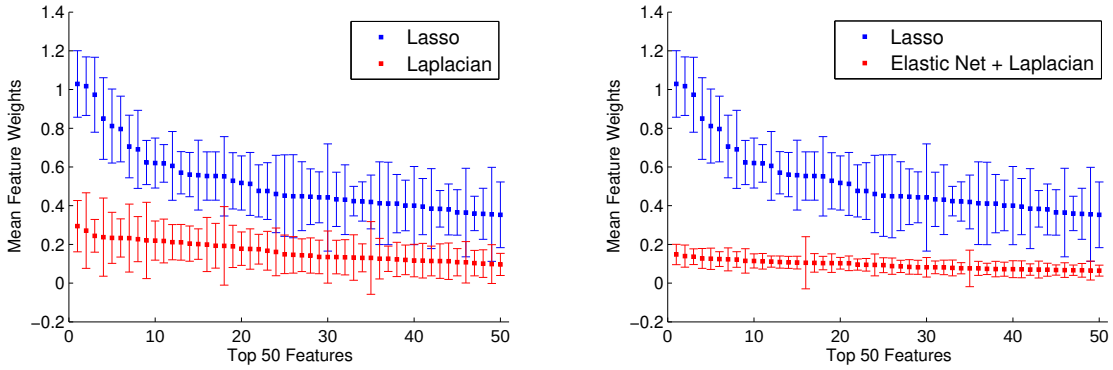
During this experiment, the lasso regularization term was fixed at $\alpha = .001$, corresponding to the value for maximum AUC of the model. Thus, feature stability through graph regularization is entirely controlled by the hyperparameter β in (4.2). The effect of β on feature stability is demonstrated in Fig. 4.10a. Both Consistency Index and Jaccard Index confirmed improvements in feature stability with increasing graph



(a) Effect of Laplacian regularization on feature stability for varying β , with $\alpha = .001$, and subset size of 100.

(b) Model estimation stability as measured by signal-to-noise ratios (SNR) of feature weights. High value of SNR indicates more stability.

Figure 4.10: Effect of EMR graph regularization



(a) Lasso versus Laplacian

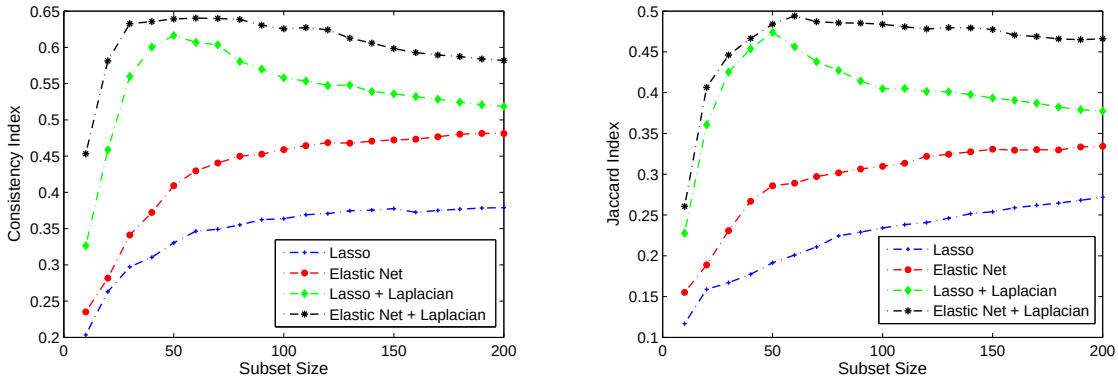
Lasso versus elastic net + Laplacian

Figure 4.11: Effect of different regularizations on mean feature weights

penalty.

Next, we compared the stabilizing effect of regularization schemes. The feature graphs were applied for lasso and elastic net, creating four alternatives – lasso (baseline, no stabilizing), elastic net, Laplacian graph, and the combined elastic net + Laplacian graph. The hyperparameters were $\alpha = .001$, $\beta = .03$, and $\lambda = .001$ for elastic net.

- For *model estimation stability*, the signal-to-noise ratios (SNR) of top individual feature weights are presented in Fig. 4.10b. Elastic net and Laplacian regularization both reduce weight variance significantly over the baseline lasso, and the Laplacian performs slightly better. With the combination of the elastic net and



(a) Consistency Index. (b) Jaccard Index.

Figure 4.12: Feature selection stability as measured by (a) Consistency Index and (b) Jaccard Index for 6-month prediction. The plot compares the similarity in feature subsets generated by models with and without different stabilization under data variations. Larger indices imply more stability.

Laplacian, the effect is greatly amplified. At 95% CIs (approximately ± 1.96 std), lasso regularization identified 2 features, Laplacian identifies 12, elastic net 16 and the combination of Laplacian+elastic net regularization identified close to 50 features. Figure. 4.11 show a finer visual representation of the effect, clearly demonstrating the reduction in weight variance using the graph regularization.

- For *feature selection stability*, Consistency Index and Jaccard Index are reported in Figure 4.12. Feature graph regularization consistently outperformed elastic net regularization for the top ranked features. Again, the combination of feature graph and elastic net resulted in the most stable set of features for all subset sizes.

4.6 Discussion

Although stability in feature selection is gaining importance (Austin and Tu, 2004; Kalousis et al., 2007; Khoshgoftaar et al., 2013), measuring the robustness of selected features in clinical prediction models has not been studied extensively. This is especially important in EMR-derived models due to its high-dimensional, dynamic and implementation-dependent nature. In practice, a stable model will allow the clinician to have more confidence on the selected features and their predictive importance.

In this chapter, we have introduced feature graphs derived from domain knowledge and Laplacian regularization to model $R_{\mathcal{D}}(\boldsymbol{w})$ for regression models to enhance stability in feature selection. Laplacian feature graphs have been used in bioinformatics (Li and Li, 2008; Cun and Fröhlich, 2013) to improve feature stability. The work (Cun and Fröhlich, 2013), for example, employs a filter-based method where the feature selection does not occur during learning of model parameters. The feature graphs were often constructed based on prior knowledge of interaction between features (e.g., genes) available from online databases. In our method, the model estimation is stabilized using a feature graph constructed from two existing feature associations in the training data. We also perform extensive numerical validation of model stability in both model estimation and feature selection.

Our experiments confirm that stability of a high dimensional linear clinical prediction model can be improved by using temporal and structural relations in EMR database. Our EMR feature graph regularisation resulted in 22% increase in feature subset stability and 40% increase in model estimation stability when compared to elastic net regularisation. The combination of Laplacian regularization with existing state-of-the-art binary elastic net resulted in most stable features without hurting the model discrimination. Thus with Laplacian regularization, more features can be confidently selected for prediction (Sub-fig. 4.10b). This is useful in the EMR setting because each patient typically has limited number of active features despite the huge number of features across the database. Having more confident features would make explanation for individual prediction easier.

With regards to performance, Laplacian regularization along with binary elastic net resulted in a model with a better fit against the validation cohort (as per Table 4.3). The marginal increase in sensitivity and classification accuracy in Laplacian regularization can be attributed to grouping of correlated features.

With regards to feature stability, the improvement upon the elastic net demonstrates that feature graph is complementary to ridge regression. This could be explained by the fact that while ridge regression tends to encourage all weights to be similar and regressed toward zero, graph regularization only requires pairwise smoothness.

Our EMR-derived model achieved a discriminatory capacity (AUC = 0.66 for 6 months) comparable with or better than existing prediction models for rehospitalization follow-

ing heart failure discharges (Ross et al., 2008). The model is derived from free available administrative and medical data, making it readily implementable into existing EMR systems. Interestingly, the top predictors discovered by our model are consistent with the existing clinical studies. Our model ranked male gender highest on the importance scale (Chin et al., 1997; Krumholz et al., 1997; Amarasingham et al., 2010). Looking at the medical factors, the strong predictors include prior history of hospitalization (past emergencies, past emergency attend time), which are consistent with those in (Chin et al., 1997; Krumholz et al., 2000, 1997; Felker et al., 2004; Amarasingham et al., 2010). The comorbidities observed were occurrence of coagulopathy in the past year and occurrence of complicated diabetes in the past three months. Other major predictors for heart failure rehospitalization are heart failure (Chin et al., 1997; Krumholz et al., 1997, 2000; Felker et al., 2004), lipoprotein metabolism disorders, angina pectoris, cataract, and chronic ischaemic heart diseases. Past number of procedures in a period of 3 months to 2 years was also ranked high.

The discrimination power, the automatic feature selection and stability control capacity suggest that the model can be used as a fast and inexpensive screening tool to select patients and risk factors for more in-depth clinical investigation. For example, through selected feature subgraphs, related risk factors can be collapsed to achieve more generality. It could serve as a first step in bridging the translational gap between bench and bedside (Amarasingham et al., 2010) . We wish to emphasize that the entire prediction process is transparent as the model is capable of explaining what risk factors are involved in a risk estimate.

4.6.1 Limitations

We acknowledge the following limitations in our study. First, since our main focus was on stabilizing a high dimensional model, we did not concentrate on improving the accuracy. In our experiments, graph regularization contributed very little to improving model discrimination. Second, we did not investigate more complex relationship between variables in EMR data when building feature graphs. The data could have rich structures and high-order regularities which can be exploited to model a more robust $R_{\mathcal{D}}(\boldsymbol{w})$, which may further enhance sharing of statistical strength between correlated features. Third, the model evaluation was not tested independently by other research-

ers. However, we have used temporal validation on unique patients, and it matches the common practice of learning models using past patients and predicting outcomes for future patient. Fourth, clinical measurements had a high degree of missingness, and hence were discarded. In review of these limitations, we believe our derived model is conservative and may have underestimated the AUC of the validation cohort.

4.6.2 Conclusion

In this study, we tackle the seldom studied but notorious problem of feature instability in clinical prediction models. Stable model features translate to proper understanding of risk factors, and hence better confidence in prognosis. Our approach consists of a novel technique to mitigate the problem by utilizing feature graphs that link similar conditions/interventions and the same condition/intervention over multiple time periods. Our extensive experiments in predicting 6-month readmission in a heart failure cohort confirm that the application of feature graphs increases the stability of the selected feature subset and reduces the variation in feature weights. The performance of the readmission models derived from administrative hospital data is competitive against existing models developed on clinical data. Further, since our approach is based on commonly available administrative attributes, models can be readily implemented on top of existing EMR systems and portable across cohorts and institutions using similar EMR databases. *We believe our stabilizing framework provides the first proof of concept in utilizing feature graphs in clinical setting and numerically validating stability for a clinical prediction model.*

A key assumption in building the feature graphs was that diagnosis codes with the same prefix have similar contributions towards patient outcome. This may be a strong assumption, especially when the diagnosis codes are long and resolve to specific conditions like first encounter and subsequent encounters. We have to look beyond such semantic relations and explore statistical correlations and higher order regularities among patient features. Also, medical cohorts are characterized by small sample sizes with high dimensionality. In such scenarios, transfer learning principles can be applied to transfer domain knowledge among related cohorts. We will address these issues, one-by-one in the following chapters.

"After all, we are nothing more or less than what we choose to reveal"

Francis Underwood, "House of Cards"

Chapter 5

Stabilization II: Data-Driven

DATA DRIVEN methods are guided by statistical relationships in the given data and empirical evidence, rather than prior assumptions or hypothesis. In the previous chapter, we used the inherent structural and temporal relationships among diagnosis codes and events to stabilize model learning. Though we were able to significantly improve model stability, we ignored statistical relationships among patient features. In this chapter, we hypothesize that underlying statistical relationships in patient records can be efficiently exploited to stabilize high-dimensional clinical prediction. Why do we believe this statistical relationship would help? To answer this question, we need to analyse the source of instability.

Our data is characterized by large degrees of freedom while number of labels is limited. The statistical relationship offers an additional source of information, which is derived directly from the features, not the labels. Since there is much information hidden in the features, we propose to use it to limit the degrees of freedom, and proceed to do so by introducing a structural prior in the Bayesian framework. A good prior is known to reduced the variance of the posterior, which is the distribution of feature weights estimated by the model.

We begin this chapter by re-iterating our objective: stabilizing a high dimensional model derived from routinely collected EMR data. We focus on minimizing the variance in feature subsets and model estimation parameters by proposing a regularizer $R_{\mathcal{D}}(\boldsymbol{w})$ to sparse model learning as in (3.4). As in the preceding chapter, we propose to

build a feature graph with nodes as EMR features and edges representing feature relationship. But in difference with the previous approach, we now discover these feature relationships from the given data. Specifically, we look at popular statistical measures such as Euclidean similarity, Cosine and Jaccard similarity and even high dimensional RBF similarity in building feature graphs. Features with similarity values greater than a predefined threshold are connected with edge weight representing corresponding similarity measure. As in the previous chapter, the exact value of threshold was determined by maximising the F-measure from training data (Lipton et al., 2014). A random walk regularization of the proposed graphs is used to stabilize a sparse Cox model that predicts time to readmission.

To test our hypothesis, we use an additional diabetic cohort along with the heart failure cohort from the previous chapter. Diabetics and heart failure share many diagnosis and comorbidities (Barrett-Connor, 2003) – presenting an interesting opportunity to explore transfer learning of feature relationships. We argue that since the predictors are related, the feature graph can be transferable. We measure feature stability using the Consistency index and model estimation stability using signal-to-noise ratio (SNR).

In summary, the main contributions in this chapter are as follows:

1. Representation of medical domain knowledge as feature graphs that embed (i) statistical correlations between features using Jaccard index (ii) aggregate of statistical and semantic correlation among features (iii) correlations between features transferred from a related cohort.

2. A random walk regularizer based on the proposed feature graphs to stabilize a Cox model as opposed to the traditional Laplacian regularizer. While Laplacian regularizer focuses on pairwise similarity, the random walk regularizer encourages group-wise similarity.

3. Demonstration of improved feature stability as measured by the Consistency index and improved model stability as measured by signal-to-noise ratio (SNR) for model regularization using proposed feature graphs. The stability measures were compared with lasso, elastic net and Laplacian semantic EMR graph regularization (introduced in the previous chapter) on a cohort of $1,784$ index admissions in heart failure patients and $2,370$ index admissions in diabetic patients admitted to a regional hospital in Australia.

4. Demonstration of improved stability, using transfer learning on related cohorts. Related cohorts like heart failure and diabetes share comorbidities and predictors. Hence, the feature graph constructed from one cohort is used to stabilize the model derived from related cohort. Stability is measured using the Consistency index and SNR.

We begin by specifying our model framework. We then proceed to expand on different methods for feature graph construction, formulate $R_{\mathcal{D}}(\boldsymbol{w})$ with the best feature graph, and finally explore transfer learning.

5.1 Model Framework

The framework of our proposed model is very similar to the setup used in the previous chapter (as in Figure. 4.6). The features from patient records in EMR database are extracted as detailed in Section 4.1.1. However, there are two key differences. First we model readmission due to heart failure as a hazard function using Cox regression (Section 5.2). Second, we model $R_{\mathcal{D}}(\boldsymbol{w})$ using a random walk regularization or locally linear embedding of a statistical graph derived from Jaccard similarity among features in EMR data. We expand on these differences in the following sections.

5.2 Sparse Cox Model

We use Cox regression to model risk of readmission (hazard function) at a future time instance, based on data from EMR. Unlike logistic regression where each patient is assigned a nominal label, Cox regression models the readmission time directly (Vinzamuri and Reddy, 2013). The proportional hazards assumption in Cox regression assumes a constant relationship between readmission time and EMR-derived explanatory variables. The formulation for Cox regression incorporates censoring information. In our data, we only had patients who were right-censored – these patients did not experience the event (re-hospitalization) for the duration of our study. Hence the survival time of these patients can be considered to be at least as long as our study. To incorporate censoring information, we slightly modify our notations for model specification in

this chapter as follows. Let $\mathcal{D} = \{\boldsymbol{x}_m, y_m\}_{m=1}^{M}$ be the training dataset with M observations, ordered on increasing y_m, where $\boldsymbol{x}_m \in \mathbb{R}^N$ denotes the feature vector for m^{th} index admission and y_m is the time to next unplanned readmission. When a patient withdraws from the hospital or does not encounter readmission in our data during the follow-up period, the observation is treated as right censored. Let k observations be uncensored and $R(t_i)$ be the remaining events at readmission time t_i.

Since the data $\mathcal{D}$ is high dimensional (possibly $N \gg M$), we apply lasso regularization for sparsity induction (Tibshirani, 1997). The feature weights $\boldsymbol{w} \in \mathbb{R}^N$ are estimated by maximizing the ℓ_1-penalized partial likelihood:

$$\mathcal{L}_{\text{sparse-cox}} = \frac{1}{M}\mathcal{L}\left(\boldsymbol{w}; \mathcal{D}\right) - \alpha \sum_{p=1}^{N} |w_j| \tag{5.1}$$

where $\|\boldsymbol{w}\|_1 = \sum_p |w_p|$, $\alpha > 0$ is the regularizing constant, and $\mathcal{L}\left(\mathbf{w}; \mathcal{D}\right)$ is the log partial likelihood (Cox, 1975) computed as:

$$\mathcal{L}\left(\mathbf{w}; \mathcal{D}\right) = \sum_{i=1}^{k} \left\{ \boldsymbol{w}^\top \boldsymbol{x}_i - \log \left[\sum_{l \in R(t_i)} \exp\left(\boldsymbol{w}^\top \boldsymbol{x}_l\right) \right] \right\}$$

We propose to stabilize the sparse model in (5.1) using $R_{\mathcal{D}}(\boldsymbol{w})$ from statistical correlations in given data. The following section describes the various statistical relations considered.

5.3 Formulating $R_{\mathcal{D}}(\boldsymbol{w})$ using RBF Kernel

Radial basis function (RBF) kernels are quite popular in support vector classification problems. In statistical learning theory, kernel functions are used to calculate the similarity between given input. Given two vectors x_i and x_j, a radial basis function kernel (K) calculates similarity as:

$$K(x_i,\, x_j) = \exp \frac{-||x_i - x_j||_2^2}{2\sigma^2} \tag{5.2}$$

where σ is a user defined parameter for controlling the "spread" of the kernel. When x_i and x_j are similar, $||x_i - x_j||$ is small. When $\sigma > 0$, vectors that are close will result

in larger RBF kernel similarity than farther vectors. The RBF values take a bell-shaped curve, where the width is controlled by the choice of σ. These values range between zero and one (when $x_i = x_j$), and hence can be interpreted as a similarity measure (Vert et al., 2004). The RBF kernel in (5.2) projects the given input vectors into an infinite dimensional space. Model learning can be regularized using the RBF kernel as in (5.2) (Smola et al., 1998). Hence the final model becomes:

$$\mathcal{L}_{\text{loss}} = \frac{1}{M}\mathcal{L}_{\text{cox}}\left(\boldsymbol{w}|\mathcal{D}\right) + \alpha\sum_{i}^{N}|w_i| + \frac{\beta}{2}\boldsymbol{w}K\boldsymbol{w}' \tag{5.3}$$

Here, $R_{\mathcal{D}}(\boldsymbol{w}) = \frac{\beta}{2}\boldsymbol{w}K\boldsymbol{w}'$, which is the RBF regularization. This approach was recently used by Vinzamuri and Reddy (2013) to improve feature selection from EMR data. Unlike traditional SVMs where the kernel calculates similarity between data points, here we calculate similarity between feature vectors.

5.4 Formulating $R_{\mathcal{D}}(\boldsymbol{w})$ using Structural Regularization

To ensure selection of correlated features, Sandler et al. (2008) proposed regularization using a network of features. We use a graph with nodes as features and edges corresponding to the statistical similarity between features. Hence the edges are non-negative with large edge weights signifying greater similarity. An edge weight of zero signifies the features have zero correlation. Let the adjacency matrix of the feature graph be G, where $G_{pq} = g \in (0, 1)$ represents the weighted similarity score between features p and q. We ensure all features have equal prominence by constraining the out-links of each node to sum to one. The medical events linked together in the feature graph should have similar weights.We introduce a random walk regularizer (Sandler et al., 2008):

$$\begin{aligned}\Omega(\boldsymbol{w};\,\mathbf{G}) &= \sum_{p}\left(w_p - \sum_{q}G_{pq}w_q\right)^2 \\ &= \boldsymbol{w}^{\top}\left(I - \mathbf{G}\right)^{\top}\left(I - \mathbf{G}\right)\boldsymbol{w}\end{aligned} \tag{5.4}$$

where I is the identity matrix. The graph stabilized model likelihood can be written as:

$$\mathcal{L}_{\text{graph}} = \mathcal{L}_{\text{sparse-cox}} - \frac{1}{2}\beta \boldsymbol{w}^{\top} \left(I - \mathbf{G}\right)^{\top} \left(I - \mathbf{G}\right) \boldsymbol{w} \tag{5.5}$$

The stabilization parameter $R_{\mathcal{D}}(\boldsymbol{w})$ becomes: $R_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{2}\beta \boldsymbol{w}^{\top} \left(I - \mathbf{G}\right)^{\top} \left(I - \mathbf{G}\right) \boldsymbol{w}$. Here the $\ell_1$regularizer introduces sparsity by pushing weak features towards zero, while the random walk regularizer distributes smoothness equally among correlated features. The gradient of (5.5) becomes:

$$\frac{\partial \mathcal{L}_{\text{graph}}}{\partial w} = \sum_{i=1}^{k} \left\{ \boldsymbol{x}_i - \frac{\sum\limits_{\ell \in R(t_i)} \boldsymbol{x}_l \exp(\boldsymbol{w}^{\top} \boldsymbol{x}_{\ell})}{\sum\limits_{\ell \in R(t_i)} \exp(\boldsymbol{w}^{\top} \boldsymbol{x}_{\ell})} \right\} - \alpha \, \text{sign}(w) - \beta \left(I - \mathbf{G}\right)^{\top} \left(I - \mathbf{G}\right) \boldsymbol{w} \tag{5.6}$$

Parameter estimation is done by maximizing the likelihood in (5.5) using L-BFGS algorithm (Liu and Nocedal, 1989).

The only remaining question becomes: how to build the feature graph $\boldsymbol{G}$. The following sections describe statistical similarity measures used to construct $\boldsymbol{G}$.

Euclidean Similarity Graph The Euclidean distance is the simplest and most widely used similarity measure in applications like clustering. To build the edges of the feature graph, we use the measure as suggested by Frey and Dueck (2007) as: $\boldsymbol{G}_{ij} = -||F_i - F_j||_2^2$, where $||F_i - F_j||_2^2$ is the squared Euclidean distance between feature vectors $F_i,\ F_j$.

Cosine Similarity Graph If we assume that the features in patient records form a network, then cosine similarity among features is a measure of structural equivalence. However, we do not have the underlying network structure or feature relationships. We propose to take the cosine similarity between feature vectors in the given data matrix. For feature vectors F_i and F_j, graph adjacency matrix using cosine similarity becomes

$$\boldsymbol{G}_{ij} = \frac{F_i \bullet F_j}{||F_i||\,||F_j||}$$

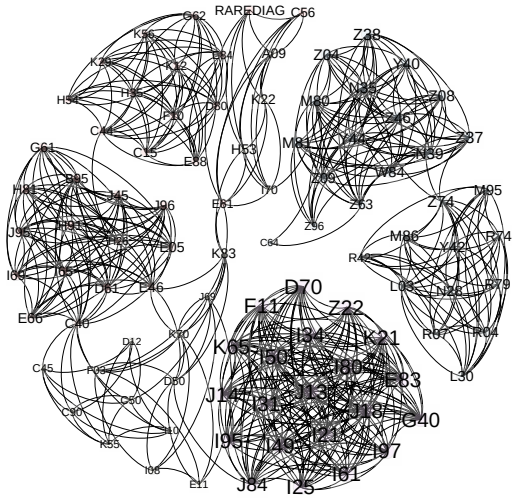


Figure 5.1: A portion of cosine graph derived from 1885 HF patients. Nodes are diagnosis codes and edges represent cosine similarity between the nodes. Size of the nodes convey prevalence.

where $F_i \bullet F_j$ is the Euclidean dot product of F_i and F_j. A portion of such graph derived from a heart failure cohort is illustrated in Figure 5.1.

Jaccard Similarity Graph Another popular statistical measure of similarity is the Jaccard index. The *Jaccard index* is the measure of the percentage of agreement between components among feature vectors. Given two feature vectors F_i and F_j, the pairwise Jaccard score reads:

$$J_{ij} = \frac{a}{a+b+c} \tag{5.7}$$

where a is the number of non-zero components in F_i and F_j, b is the number of non-zero components in F_i but not in F_j and c is number of non-zero components in F_j but not in F_i. We construct an undirected graph with nodes as features and edges representing the Jaccard score between features. Figure 5.2 shows a portion of the Jaccard graph derived from the heart failure cohort.

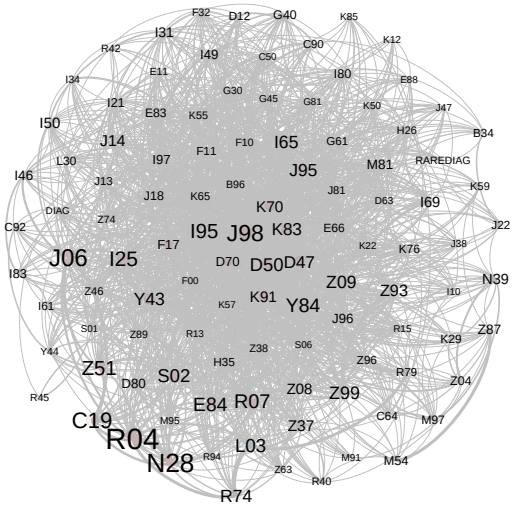


Figure 5.2: A portion of Jaccard graph derived from 1885 HF patients. Nodes are diagnosis codes and edges represent Jaccard similarity between the nodes. Size of the nodes convey prevalence.

5.4.1 Graph aggregation.

The semantic EMR graph used in the previous chapter captures the temporal relationship between features and the general relationship between diagnostic codes based on the ICD-10 structures. The statistical graphs listed above are cohort specific and derive correlations directly from given data. Combining domain knowledge with insights derived from data could yield better stabilization. Hence, we propose aggregating the two graphs: semantic EMR graph from Section 4.2, denoted as G_{EMR} and the statistical graph G_{Stat}.We use a simple aggregation technique to construct the final $\langle \text{EMR}; \text{Stat} \rangle$ graph as:

$$G_{\langle \text{EMR;Stat} \rangle} = \max(G_{\text{EMR}}, G_{\text{Stat}}) \tag{5.8}$$

5.4.2 Transferred Graphs

Finally, we examine the capability of our proposed method in transfer learning. Knowledge from one domain can be transferred to a related domain when data is scarce or expensive to collect (Pan and Yang, 2010). Getting high quality training data is often difficult, particularly in a medical setting. Cohorts that share comorbidities and diagnoses, as in diabetes and cardiovascular diseases, are likely to have similar correlations

among features. Accordingly, we propose to stabilize a Cox model derived from one cohort using the statistical similarity graph constructed from a related cohort. We denote the transferred graph as: TL-Stat graph. Further, we use TL-Stat graph to construct the aggregated graph:

$$G_{\langle \text{EMR;TL-Stat} \rangle} = \max(G_{\text{EMR}}, G_{\text{TL-Stat}})$$

Here, the temporal and hierarchical feature relations in the cohort are captured by the EMR graph. The statistical relations among features, which can be expensive to calculate, are transferred from the related cohort using TL-Stat graph.

5.5 Experiments

In this section, we evaluate feature and model stability of our framework. The results are reported on two cohorts: heart failure (HF) and diabetes (DB), provided by Barwon Health (as detailed in Section 4.1). We collect retrospective data for heart failure and diabetes patients from the hospital EMR database. The heart failure cohort contains all patients with at least one ICD-10 diagnosis code I50, while the diabetes cohort includes all patients with at least one diagnosis code between E10-E14. This resulted in $1,885$ heart failure admissions and $2,840$ diabetes admissions between January 2007 and December 2011. Patients of all age groups were included whilst inpatient deaths were excluded. We focus our study on emergency attendances and unplanned admissions of patients. The heart failure cohort was introduced in the previous chapter (see Table 4.1). The characteristics of diabetic cohort are listed in Table 5.1.

We perform temporal validation for both cohorts as described in Section 4.4. Feature selection stability is measured using Consistency index (Section 2.2.3.2) and model stability was evaluated using signal-to-noise ratio (Section 2.2.3.2).

We use the one-sided convolutional filter bank, as detailed in Section 4.1.1, to extract a large pool of features from EMR databases. The filter bank summarizes event statistics over multiple time periods and granularities. The feature extraction process resulted in $3,338$ features for heart failure cohort and $7,641$ features for diabetes cohort. The extracted features are used to derive a sparse Cox model. Our proposed feature graphs

	Diabetes	
	Training set	Testing set
Checkpoint	Dec 2008	
Number of admissions	1,341	1,029
Unique patients	951	765
Gender		
Male	501 (52.68%)	407 (53.20%)
Female	450 (47.32%)	358 (46.80%)
Mean age (years)	57.8	56.4

Table 5.1: Characteristics of training and validation cohorts.

capture correlations between these features to stabilize model learning.

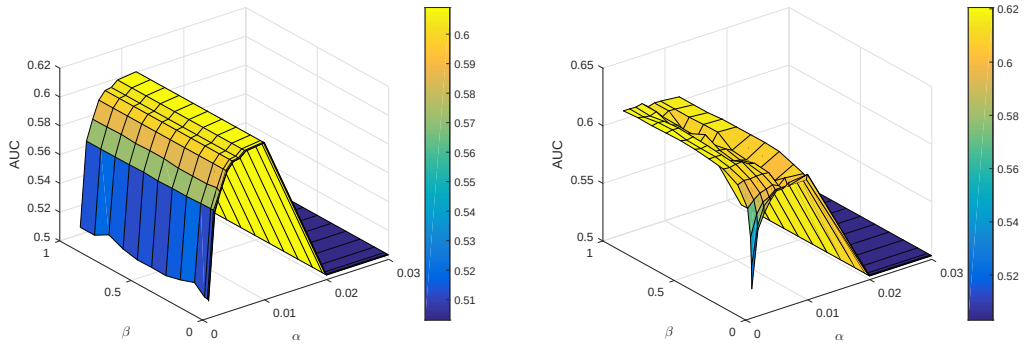
5.5.1 Results

Our models are designed using two hyper-parameters: lasso regularization parameter α and graph regularization parameter β. We empirically tune these parameters to improve feature stability without hurting model discrimination. Overall, feature stability depended more on graph parameter β, while model discrimination was more sensitive to α. A good trade-off was achieved at $\alpha = 0.003$ and $\beta = 0.8$ for cosine and Jaccard graph models, while RBF regularized mode resulted in best parameters of $\alpha = 0.01$ and $\beta = 0.3$. Figure 5.3 illustrates the variation in model discrimination (as measured using area under ROC curve) for different settings of hyperparameters on the heart failure cohort. The models exhibited similar behaviour on diabetes cohort.

All models are externally validated against (i) heart failure cohort with a 6-month horizon (ii) diabetes cohort with a 12-month horizon. Table 5.2 reports the AUC scores with confidence intervals for our proposed models. The predictive performance is comparable with the baselines.

5.5.1.1 Stability against Data Re-sampling

We now compare the effects of our proposed stabilization strategies. In these experiments, we fix the lasso regularization parameter α and graph regularization parameter β

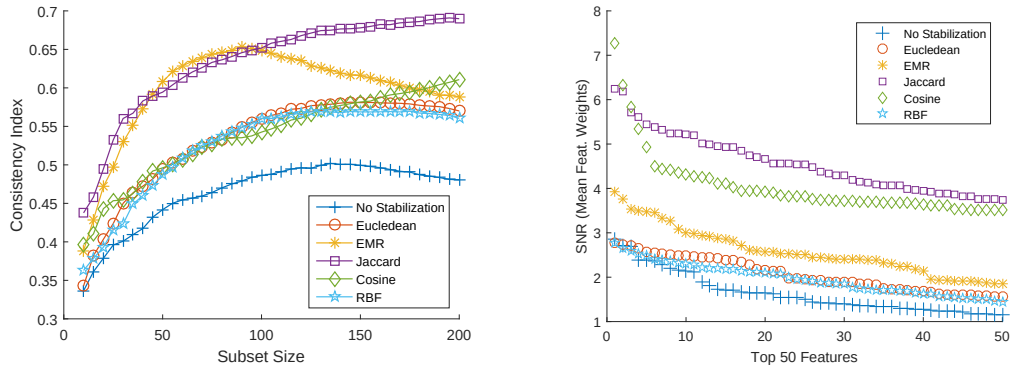


(a) RBF regularized model (b) Statistical graph regularized model

Figure 5.3: Effect of lasso regularization α and graph regularization β for different stabilization models on heart failure cohort.

Stabilization	**AUC**	
	HF	DB
No stabilization (lasso)	0.60 [0.55,0.66]	0.74 [0.70, 0.77]
RBF	0.61 [0.55,0.66]	0.75[0.72, 0.79]
Cosine	0.62 [0.55,0.67]	0.76 [0.73, 0.79]
Jaccard	0.62 [0.56,0.68]	0.76 [0.73, 0.79]

Table 5.2: Performance comparison of different graph stabilization mechanisms on heart failure and diabetes cohort



(a) Feature subset stability (b) Model estimation stability

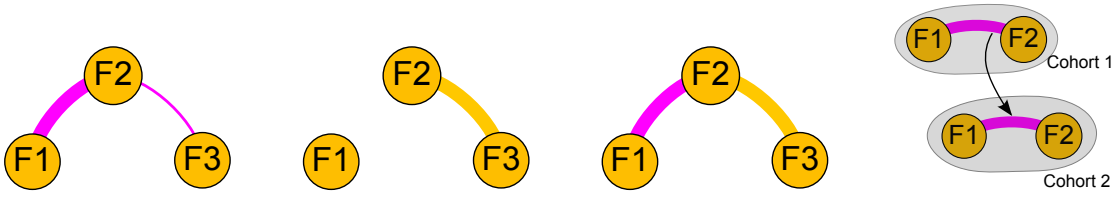
Figure 5.4: Comparing feature subset stability and model estimation stability of our proposed methods on heart failure cohort. Our proposed models are compared with lasso (no stabilization) and EMR Graph proposed in the previous chapter.

to the values corresponding to maximum AUC. Each model regularization is subjected to 100 bootstraps. The feature subsets returned from the bootstraps are compared using Consistency index, and variation in model weights is measured using Signal-to-Noise ratio. Among all statistical methods, Jaccard graph performed the best for stabilizing feature selection in heart failure cohort, as illustrated in Figure 5.4. Similar results were obtained for diabetic cohort.

For increasing feature subset sizes (>100), Jaccard graphs proved effective. The temporal and structural relations of diagnosis codes have stronger effect for small set of features, while Jaccard index was effectual on larger sets. This behaviour suggests aggregating statistical and semantic structures. In the following section we examine the results of graph aggregation and transfer learning using Jaccard graphs.

5.5.1.2 Graph Aggregations and Transfer Learning

For this set of experiments, we investigate the effect of aggregating knowledge driven graph (G_{EMR}) and statistical graph, particularly the Jaccard graph G_{Jaccard}. We re-iterate our graph construction process for clarity. We construct $\mathbf{G}$ using the following methods. First, we represent the edges using the Jaccard index between features, as in Fig. 5.5.(a). Second, we aggregate the baseline semantic EMR graph and the Jaccard graph. Here, each edge is the maximum of Jaccard and semantic scores between the



(a) Jaccard (b) Knowledge-driven semantic (c) Aggregate (d) TL Jaccard

Figure 5.5: Feature correlation captured by constructing feature graph with nodes as features and edges as: (a) statistical correlation measured using Jaccard score (b) semantic relations derived from temporal and ICD-10 structures (c) aggregation of Jaccard and Semantic graphs. (d) transfer of Jaccard similarity between features from a related cohort.

features (Fig. 5.5.(c)). Finally, we investigate transferring the adjacency matrix between related cohorts. Specifically, the Jaccard similarity scores between features in one cohort is transferred to a related cohort (Fig. 5.5.(d)).

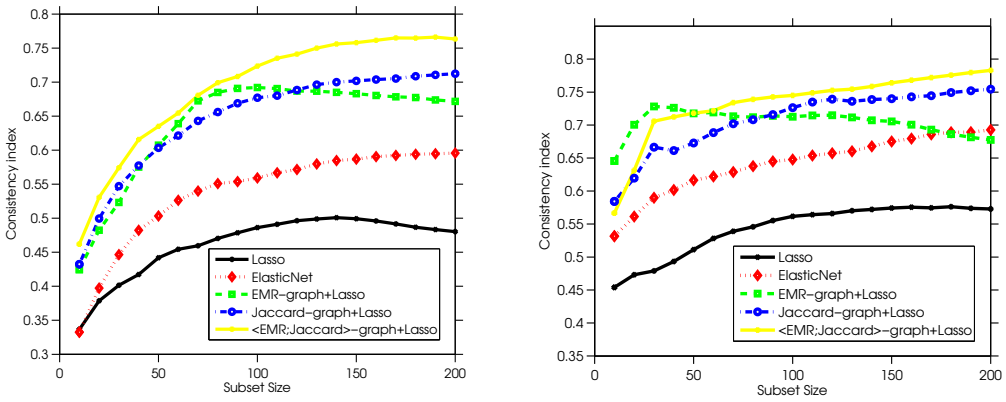
The baseline regularization methods for the readmission model are chosen to be (i) lasso (ii) elastic net (iii) knowledge-driven EMR graph (as in Chapter 4). Based on the construction of the feature graph, we arrive at four different models: (i) Jaccard graph regularized model: feature graph is the Jaccard similarity graph among features in the given cohort (ii) EMRJaccard regularized model: feature graph is the aggregation of Jaccard graph with semantic EMR graph, as in ((5.8)) in the given cohort (iii) TL Jaccard regularized model: feature graph is the Jaccard similarity graph transferred from a related cohort (iv) EMR; TL Jaccard regularized model: feature graph is the aggregation of semantic EMR graph from the given cohort and Jaccard graph transferred from a related cohort.

The maximum AUC scores (along with confidence intervals) for models and baselines are reported in Table 5.3. We observe that knowledge-driven and data-driven regularization offers very little in terms of improving performance.

For the top 100 predictors, EMRJaccard graph stabilization demonstrated the highest feature stability in both cohorts (see Fig. 5.6). Next, we compare variance in parameter weights using SNR measures. In Fig. 5.8, each model is represented by average of its 20 highest SNR values. The Jaccard graph regularized model proved to be most robust in both cohorts. Interestingly, model stability using EMRJaccard graph was similar to elastic net and was not able to improve upon semantic EMR graph or Jaccard graph.

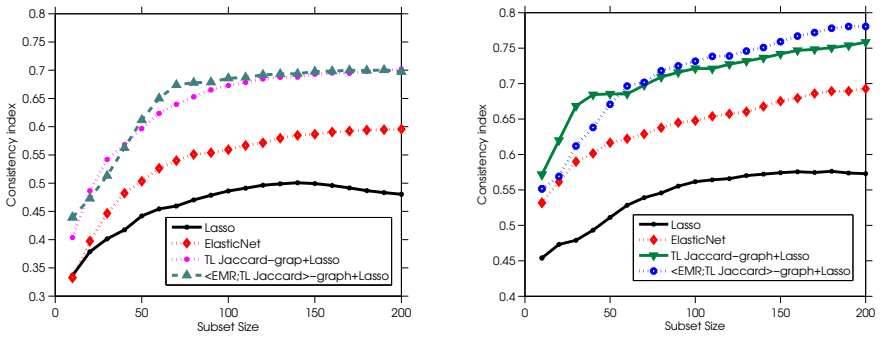
	Heart failure	Diabetes
Lasso	0.60 [0.55; 0.66]	0.74 [0.70; 0.77]
Elastic net	0.61 [0.55; 0.67]	0.75 [0.72; 0.79]
EMR-graph+Lasso	0.61 [0.56; 0.67]	0.76 [0.72; 0.79]
Jaccard-graph+Lasso	0.62 [0.56; 0.67]	0.76 [0.73; 0.79]
⟨EMR; Jaccard⟩-graph+Lasso	0.62 [0.56; 0.67]	0.76 [0.73; 0.79]
Stabilization using Transfer Learning		
TL_Jaccard-graph+Lasso	0.62 [0.56; 0.67]	—
⟨HF_EMR; TL_Jaccard⟩-graph+Lasso	0.62 [0.57; 0.68]	—
TL_Jaccard-graph+Lasso	—	0.76 [0.73, 0.79]
⟨DB_EMR; TL_Jaccard⟩-graph+Lasso	—	0.75 [0.72, 0.79]

Table 5.3: AUC scores with confidence intervals for readmission prediction within 6 months for heart failure and 12 months for diabetes patients. Model performance on individual cohorts and on cohorts with Jaccard graph transferred from the other cohort is shown in separate sections.



(a) Heart failure. (b) Diabetes.

Figure 5.6: Stabilization using statistical and semantic structures. Feature stability measured by the consistency index as functions of the subset size for readmission prediction within 6 months for heart failure (Fig. 5.6a) and 12 months for diabetes patients (Fig. 5.6b). Larger indices imply more stability.



(a) Heart failure. (b) Diabetes.

Figure 5.7: Stabilization using transfer of Jaccard graph (TL Jaccard). Stabilization using statistical and semantic structures. Feature stability measured by the consistency index as functions of the subset size for readmission prediction within 6 months for heart failure (Fig. 5.7a) and 12 months for diabetes patients (Fig. 5.7b). Larger indices imply more stability.

Stabilization using transfer learning. We investigate transfer of feature graphs between related cohorts. For the heart failure cohort, TL Jaccard graph represents the Jaccard scores transferred from diabetes cohort, while EMR;TL Jaccard graph is the aggregation of the semantic EMR graph of heart failure cohort and Jaccard graph transferred from diabetes cohort. The same technique is applied to stabilize diabetes cohort, where the Jaccard scores are transferred from heart failure cohort. We compare the transferred graph stabilization with lasso and elastic net. Our experiments confirm that cross-domain graphs also help the stability of feature selections (see Figure. 5.7) and model estimation (see Figure. 5.8).

5.6 Discussion

Integrating domain knowledge to improve learning has been gaining much attention recently (Sandler et al., 2008). Biological understanding of gene-disease networks, for example, has enabled discovery of what genes contribute to a disease, and what proteins would bind with a particular chemical compound (Barabási et al., 2011). However, little has been explored in networks derived from the healthcare processes and their contribution to prediction models.

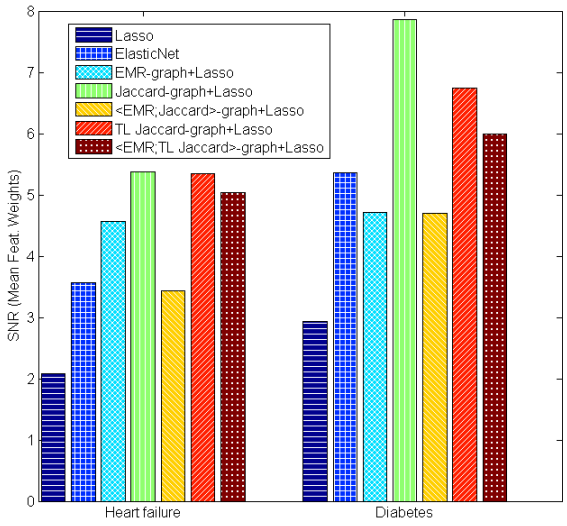
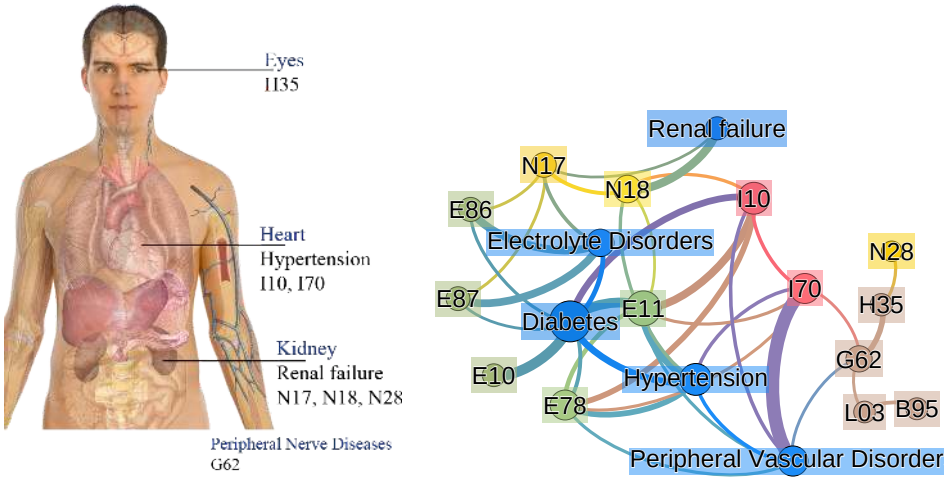


Figure 5.8: Model estimation stability measured by signal-to-noise ratios (SNR) of feature weights. High value of SNR indicates more stability. *"TL Jaccard"* means the Jaccard-graph used in transfer learning settings: heart failure Jaccard graph for stabilizing diabetes data and diabetes Jaccard graph for stabilizing heart failure cohort.

In this chapter, we extend our previous work on stabilizing high-dimensional clinical prediction using statistical relations automatically discovered from the given data. We explored kernel based and structural based regularizations, and concluded that regularizer $R_{\mathcal{D}}(\boldsymbol{w})$ based on random walk transformation of Jaccard similarity graph (G_{Jaccard})from EMR features demonstrated better stabilization. Model stability was further enhanced when data-driven Jaccard graph was combined with knowledge-driven semantic graph built from structural relations in diagnosis codes ($G_{<\text{EMR;Jaccard}>}$). This suggests structural relations and statistical relations among features is complementary and combining such relations during model regularization results in enhanced stability. The statistical graphs improved feature stability by 66% and model stability by 50% when compared to elastic nets.

5.6.1 Transfer Learning by Identifying Comorbidity relations

Medical events often co-occur, especially in aged cohorts. For example, the presence of comorbidities causes multiple diagnoses at the same time. Comorbidity is the presence of multiple co-existing diseases in a patient. Feinstein (1970) defined comorbidity as:



(a) (b)

Figure 5.9: Extracting disease correlations in diabetic cohort. Common comorbidities and diagnosis codes are shown in (a). A portion of the disease graph constructed using Jaccard similarity between EMR features in a diabetic cohort is shown in (b). The nodes represent EMR features, and links represent interaction strength, measured using Jaccard index. Blue nodes are co-occurring diseases, green nodes are diagnosis codes for diabetes, orange nodes for heart diseases, and yellow nodes for urinary diseases.

"*any distinct clinical entity that has co-existed or that may occur during the clinical course of a patient who has the index disease under study*". For example, a patient with diabetes frequently has hypertension (high blood pressure), dyslipidemia (Abnormal LDL, HDL, or triglycerides, increasing risk for heart attack), liver complications, cardiovascular disease, kidney disease and obesity (Pantalone et al., 2015). Such domain knowledge should ideally be integrated into feature selection process during clinical prediction.

In this chapter, we captured feature correlation in a knowledge network, with features as nodes and relations between features as edges. We examined popular statistical similarity measures to build such network from the data and concluded that Jaccard similarity is better suited for our cohort. Our Jaccard graph was able to capture the common complications in a diabetic cohort (as in Figure 5.9(a)[1]) as a feature graph shown in Figure 5.9(b).

This ability to automatically find implicit correlations could be the reason for improved stability during transfer learning. Our transfer learning process uses statistical correlations among features in one cohort to construct the feature graph in a related cohort.

[1]Image courtesy: http://www.clker.com/clipart-human-body-anatomy-basics-no-lines.html

In this chapter, we conducted experiments on heart failure and diabetes cohort. Heart failure is a chronic condition that also affects several other organs, most importantly, the kidney (Amaral et al., 2013; Moukarzel et al., 2013). Diabetic patients are also at increased risk of kidney diseases (Johnson et al., 2007; Fox et al., 2012). Further, diabetes was found to be an independent risk factor for heart failure (Widmer, 2011; van Deursen et al., 2014; Lüscher, 2015), with several studies including the 18 year Framingham study establishing strong correlation between diabetes and heart failure conditions (Kannel et al., 1974; Huo et al., 2016; Gregg et al., 2016). Our graph in Figure 5.9 was able to discover significant correlation between diabetics, heart problems and kidney diseases (including renal failure), among other relationships. Transferring this feature graph between heart failure and diabetes cohort resulted in enhanced stability.

5.7 Conclusion

Novel methods in feature selection often concentrate on model performance and overlook stability (Vinzamuri et al., 2014; Bilal et al., 2013). Stable predictors inspire confidence in prognosis, as they are often subjected to further examinations. In this chapter, we utilize statistical and semantic relations in EMR data to stabilize a sparse Cox model for predicting readmission. The model is validated on two different retrospective cohorts. When compared with similar studies, the model AUC is competitive and the top predictors were found to be common with related studies (Ross et al., 2008). On two stability measures, the proposed method has demonstrated to largely improved stability. In related cohorts, when collecting data becomes expensive, transferring domain knowledge using TL-Jaccard graph was also found to improve stability. The practical significance is that our proposed model is hypothesis free, fully automated and derived from freely available administrative and medical data.

The data driven methods in this chapter address the concerns over strong assumptions of EMR feature graphs built from ICD-10 structures. The Jaccard similarity graph is derived directly from given data with no prior hypothesis. Our experiments demonstrated such graphs are complementary to EMR feature graphs.The random walk regularization of the aggregated feature graph promotes group level selection and rare-but-important features. However, we have overlooked higher-order correlations in our data. In the next chapter, use higher-order correlations in data to formulate $R_{\mathcal{D}}(\boldsymbol{w})$.

"To understand is to perceive patterns."

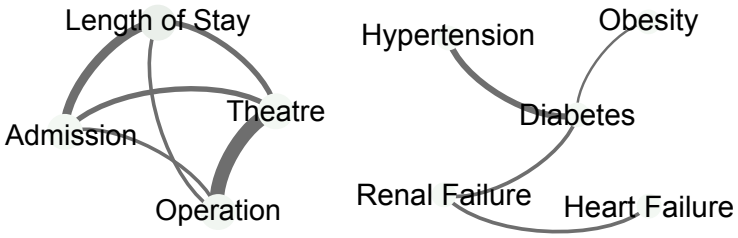
Isaiah Berlin

Chapter 6

Stabilisation III: Pattern Discovery

GRAPH based regularizations from previous chapters investigated pairwise and groupwise constraints using semantic relations in diagnosis codes and statistical relations derived from data. Yet, the graphs we derived captured only first order correlation in data. These first order correlations may be in hospital events or patient diagnosis, as example in Figure 6.1. Often, high dimensional data may contain linear, as well as non-linear correlations among features, as demonstrated in Figure 6.2. Feature transformation methods can be applied to uncover low-dimensional embeddings from such data.

Low dimensional embedding or patterns can reveal interesting feature relationships that were invisible to the graph based methods used in the previous chapters. The question now becomes how to exploit patterns and formulate $R_{\mathcal{D}}(\boldsymbol{w})$ for stabilizing model



(a) Correlation in events (b) Correlation in comorbidities

Figure 6.1: An example of first-order feature correlations in heart failure cohort. Nodes represent events and edges represent correlation strength.

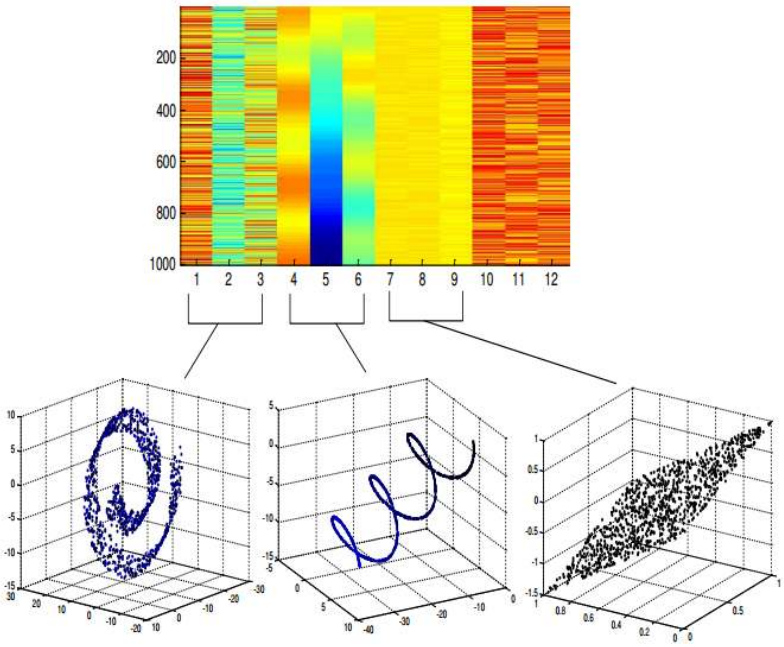


Figure 6.2: Linear and non-linear local correlations in example data used in Zhang et al. (2010).

learning as in (3.4). In this chapter, we resort to recent advances in deep learning and self-taught learning (Raina et al., 2007). Specifically, we use an autoencoder network to learn higher order correlations in our data. This is done by examining the encoding weights of the neural network. When an autoencoder has significantly lesser nodes in the hidden layer, the encoding weights capture all correlations in data, which can be used to stabilize model learning.

We list the main contributions of this chapter:

1. To capture higher order correlations in data, we reformulate our learning model by factorizing the linear parameter as a combination of a lower dimensional vector u and a high dimensional matrix W. By modelling W as the encoding weights of an autoencoder network, we capture higher order feature correlations in data.

2. We model the stabilizing factor: $R_{\mathcal{D}}(\boldsymbol{w})$ using the encoding weights W, which capture all levels of feature correlation. We extend this model to use semantic relations by further regularization with feature graph derived in Chapter 4.

3. We propose a more robust estimation of higher order correlation matrix W by augmenting the training data with an external cohort. This is in accordance with

the principles of self-taught learning (Raina et al., 2007).

4. We demonstrate improved feature subset stability and model estimation stability for each sparse linear model regularized with: 1) autoencoder derived from training cohort, 2) combination of autoencoder and feature graph derived from training cohort, 3) combination of feature graph derived from training cohort and autoencoder derived from augmenting an external cohort to training data. We conducted our experiments on $1,885$ heart failure admissions from an Australian hospital. The augmented external data consisted of $2,840$ diabetic admissions. Feature stability was measured using Consistency index. Parameter estimation stability was measured using Signal-to-Noise Ratio (SNR).

The rest of the chapter is organized as follows. We begin by describing our model specification that uses factorization of the learning parameter. We then explain autoencoder learning and regularization to formulate the modified $R_{\mathcal{D}}(W)$. Two extensions of this technique are proposed that ensures tighter constraints. Finally we examine the results of our proposed stabilization techniques and compare it with previous approaches.

6.1 Framework

Our model is built on patient records vectorized using the feature extraction process detailed in Section 4.1.1. However, the general framework now differs from the previous chapters in the following ways. To model higher order correlations in data, we begin by decomposing model parameter $\boldsymbol{w}$ in (3.4) into a lower order vector u and a high dimensional matrix W as: $\boldsymbol{w}_{\mathrm{N}\times 1} = W^T_{k\times \mathrm{N}}\, u_{k\times 1}$, where $k \ll \mathrm{N}$. This factorization offers several advantages. The lower dimensionality of u makes it more easier to learn and more stable to data variations. The W captures higher order correlations that be modelled using different auxiliary tasks. Greater number of tasks ensure better solution, since there are more constraints. Hence, we can rewrite 3.4 as:

$$\mathcal{L}_{\text{loss}} = \frac{1}{M}\,\mathcal{L}\left(W^T\,u|\mathcal{D}\right) + \alpha\left|W^T\,u\right| + R_{\mathcal{D}}(W) \tag{6.1}$$

In this modified formulation, the stability component $R_{\mathcal{D}}(.)$ focuses only on higher order component W, since the lower dimensional u is assumed to be stable. We now

need a model to learn the parameters u and W, and a feature transformation method to derive $R_{\mathcal{D}}(W)$.

As a concrete example for generalized linear models, we work on binary prediction using logistic regression. The modified logistic loss function $\mathcal{L}(\boldsymbol{w}|\mathcal{D})$ using u and W becomes:

$$\begin{aligned} \mathcal{L}_{\text{logit}}(u, W\,|\mathcal{D}) &= \log(1+\exp(-yu^T W\boldsymbol{x})) \\ &= \log(1+\exp(-yu^T z)) \end{aligned} \tag{6.2}$$

where $y \in \pm 1$ represents the data label[1]. Notice that $z = W\boldsymbol{x}$ is a data transformation from N dimensions to the smaller k dimension. To learn W, we need to choose a competent auxiliary task. The primary objective of this task involves feature transformation to a lower dimensional subspace or manifold that is embedded in the full-dimensional space.

One of the most popular methods for such transformation is principal components analysis or PCA (Jolliffe, 2002). Using PCA, high-dimensional data can often be represented using a lower dimensional linear representation, when there is a linear manifold near the original high-dimensional data. PCA projects the data onto this linear manifold without losing much information. This process can also be done using a neural-net with linear units and a single hidden layer (Hinton and Salakhutdinov, 2006). Interestingly, we can generalize such feature transformation to include lower dimensional non-linear manifolds by using non-linear activation functions and deep neural nets.

In our experiments, we model W as the encoding weights of a classical autoencoder with sigmoid activation function, derived from the same data $\mathcal{D}$. The workflow diagram of our method is illustrated in Figure 6.3.

6.1.1 Learning Higher Order Correlations using Autoencoder

An autoencoder is a neural network that learns by minimizing the reconstruction error using back-propagation (Bengio, 2009). The learning process is unsupervised, wherein the model learns useful properties of data. An autoencoder network consists of two

[1]We ignore the bias parameter for simplicity

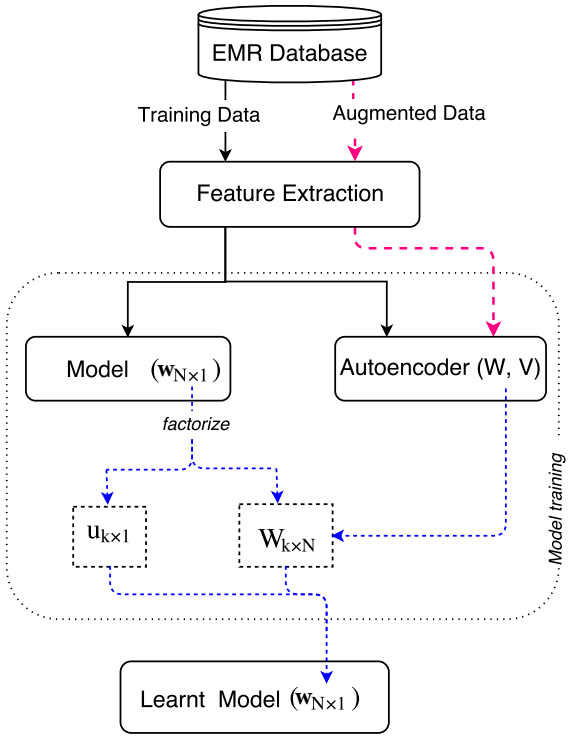


Figure 6.3: The work-flow diagram of our framework for deriving autoencoder stabilized prediction model from EMR. The model parameter $\boldsymbol{w}$ is factorized into a lower dimensional vector u and high dimensional matrix W. The W matrix is jointly modelled as encoding weights in an autoencoder network and is used to regularize the prediction model.

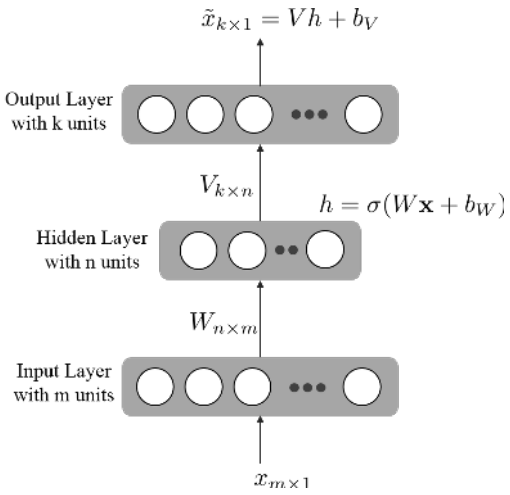


Figure 6.4: General framework of an autoencoder with one hidden layer

components: (1) An *encoder function* that maps the input data $\boldsymbol{x} \in \mathbb{R}^N$ as: $h(x) = \sigma(W\boldsymbol{x} + b_W)$, where σ can be any non-linear function (for e.g., the sigmoid function) and W, b_W are the weights and bias of the hidden layer (2) A *decoder function* that attempts to reconstruct the input data as: $\tilde{x} = Vh + b_V$, where V, b_V are the weights and bias of the output layer. This is illustrated in Figure 6.4. The loss function is modelled as the reconstruction error:

$$\mathcal{L}_{\text{AE}}(W, V, b_{W,} b_V|\mathcal{D}) = \frac{1}{2N}||x - b_V - V\sigma(Wx + b_W)||_2^2 \tag{6.3}$$

Once trained, evaluating a feed forward mapping using the encoder function gives a latent representation of the data. When the number of hidden units is significantly lesser than the input layer, W encapsulates the higher order correlations among features.

We propose to regularize our sparse linear model in (6.2) using the autoencoder framework in (6.3). The joint loss function becomes:

$$\begin{aligned}\mathcal{L}_{\text{model}}(u, W, V, b_W, b_V|\mathcal{D}) &= \mathcal{L}_{\text{logit}}(u, W\,|\mathcal{D}) + \\ &+ \alpha \sum_i |\sum_k W_{ik}^T u_k| \\ &+ \lambda_{\text{AE}}\, \mathcal{L}_{\text{AE}}(W, V, b_W, b_V|\mathcal{D}) \\ &+ \lambda_{\ell 2} \left(W^2 + V^2 + b_W^2 + b_V^2\right)\end{aligned} \tag{6.4}$$

where $\alpha > 0$ is the lasso regularization parameter which ensures weak $\boldsymbol{w}_i = \sum_k W_{ik}^T u_k$ are driven to zero. While λ_{AE} controls the amount of regularization due to higher order correlation, $\lambda_{\ell 2}$ controls overfitting in autoencoder. The loss function in (6.4) is non-convex. Referring to (6.1), the stabilization parameter now becomes:

$$R_{\mathcal{D}}(W) = \lambda_{\text{AE}}\, \mathcal{L}_{\text{AE}}(W, V, b_W, b_V|\mathcal{D}) + \lambda_{\ell 2} \left(W^2 + V^2 + b_W^2 + b_V^2\right)$$

The number of nodes in the hidden layer was chosen to be around 20% of the total number of features. We now propose two extensions to the model in (6.4) by adding further constraints on $R_{\mathcal{D}}(W)$.

6.1.1.1 Augmenting Feature Graph regularization

For the first constraint, we revisit predefined associations in patient medical records. Autoencoders can be used to find automatic feature grouping, but as we saw in Chapter 4, we can enforce two additional associations from domain knowledge – diseases or conditions reoccurring over multiple time-horizons (as detailed in Section 4.2.1), and hierarchical nature of ICD-10 diagnosis and procedure codes (as described in Section 4.2.2).

We build a feature graph regularizer using these associations, as in Section 4.2.3, and use it to further regularize our model in (6.4) as:

$$\begin{aligned} \mathcal{L}_{\text{model-fg}}(u, W, V, b_W, b_V|\mathcal{D}) &= \mathcal{L}_{\text{model}}(u, W, V, b_W, b_V|\mathcal{D}) \\ &+ \frac{1}{2}\lambda_{\text{fg}}\left[(u^T W)\,\mathbf{L}\,(W^T u)\right] \end{aligned} \tag{6.5}$$

In this case, our modified $R_{\mathcal{D}}(W)$ now becomes:

$$\begin{aligned} R_{\mathcal{D}}(W) =& \lambda_{\text{AE}}\,\mathcal{L}_{\text{AE}}(W, V, b_W, b_V|\mathcal{D}) \\ &+ \lambda_{\ell 2}\left(W^2 + V^2 + b_W^2 + b_V^2\right) \\ &+ \frac{1}{2}\lambda_{\text{fg}}\left[(u^T W)\,\mathbf{L}\,(W^T u)\right] \end{aligned}$$

6.1.1.2 Augmenting External data for Autoencoder learning

The encoding weights W in (6.3) can be estimated from multiple sources. For example, we propose to augment the current training data $\mathcal{D}$ (for example: heart failure cohort) with another cohort containing the same features (say, diabetic cohort). Training the autoencoder network on this augmented data will result in more robust estimation of W. Both cohorts are from the same hospital. Hence the common features in these cohorts share the same feature space. Augmenting additional data from same feature space would aid in finding latent correlations in data. The augmented data is provided only for autoencoder learning, and not for the linear model. If the augmented dataset

is $\mathcal{D}_{\text{AUG}}$, we can specify our model as:

$$\begin{aligned}\mathcal{L}_{\text{model}}(u, W, V, b_W, b_V|\mathcal{D}) = \mathcal{L}_{\text{logit}}(u, W\,|\mathcal{D}) + R_{\mathcal{D}_{\text{AUG}}}(W) \text{ where,}\\ R_{\mathcal{D}_{\text{AUG}}}(W) =& \lambda_{\text{AE}}\,\mathcal{L}_{\text{AE}}(W, V, b_W, b_V|\mathcal{D}_{\text{AUG}})\\ &+ \lambda_{\ell 2}\left(W^2 + V^2 + b_W^2 + b_V^2\right)\\ &+ \frac{1}{2}\,\lambda_{\text{fg}}\left[(u^T W)\,\mathbf{L}\,(W^T u)\right] \end{aligned} \tag{6.6}$$

6.2 Experiments

We now evaluate our proposed stabilization strategies using heart failure (HF) and diabetes (DB) cohorts. The cohort details and setting is similar to the previous chapter, as described in Section 5.5

As with previous experiments, all models undergo temporal validation for both cohorts as described in Section 4.4. Feature selection stability is measured using Consistency index (Section 2.2.3.2) and model stability was evaluated using signal-to-noise ratio (Section 2.2.3.2).

6.2.1 Models and Baselines

On HF and DB data, we derive a lasso regularized logistic regression model to predict heart failure readmissions in 6 months. We force lasso to consider higher order correlations in data by using the following three regularization schemes:

1. ***Lasso-Autoencoder:*** The linear model is regularized by encoding weights of an autoencoder derived from HF cohort as described in (6.4). This becomes our initial model. We now extend this model with two regularization schemes.

2. ***Lasso-Autoencoder-Graph:*** For our first extension, we use the feature graph regularization as in (6.5). We construct a feature graph from $3,338$ features in HF cohort as in Section 4.2, and use it to further regularize the Lasso-Autoencoder model as in (6.5).

$R_{\mathcal{D}}(\boldsymbol{w})$	Sensitivity	Specificity	Precision	F-Measure	AUC
Pure Lasso	0.38	0.77	0.62	0.47	0.60
Elastic Net	0.38	0.80	0.65	0.48	0.62
EMR Graph	0.42	0.73	0.59	0.49	0.64
AE	0.39	0.76	0.61	0.47	0.64
AE-Graph	0.39	0.76	0.61	0.47	0.64
AG-AE-Graph	0.39	0.76	0.61	0.47	0.64

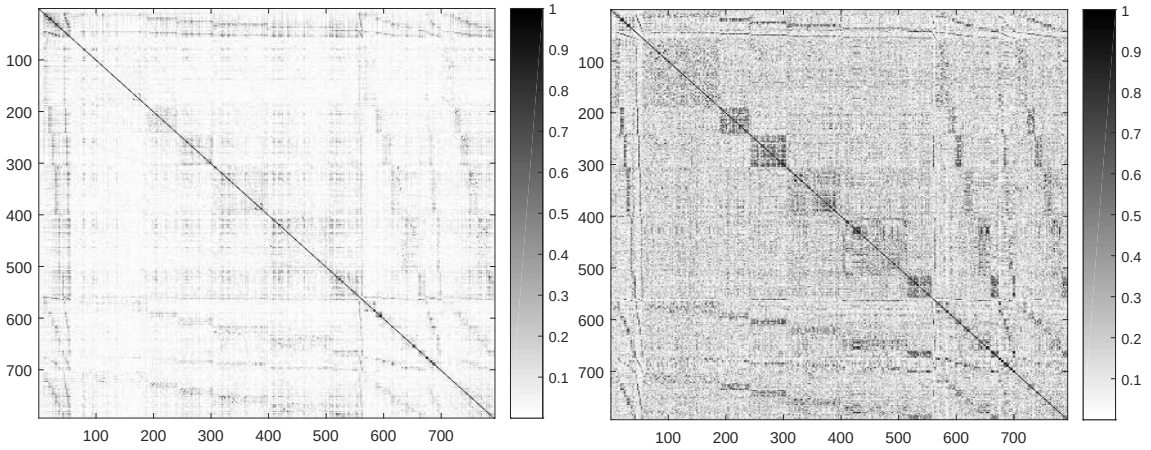
Table 6.1: Comparing model performance for different $R_{\mathcal{D}}(\boldsymbol{w})$ settings. AE denotes autoencoder regularization. AG denotes augmented data.

3. ***AG-Lasso-Autoencoder-Graph:*** Our third and final extension to autoencoder regularization consists of using augmented data to train the autoencoder. We use the notation AG to denote augmented data. To estimate W, we used DB cohort augmented to the HF cohort. Training data consisted of 558 features common to both HF and DB. The sparse prediction model was built from common features in HF cohort, and regularized using a HF-based feature graph and autoencoder from augmented data as in (6.6).

We compare the stability of our proposed regularization methods with the following baselines: 1) pure lasso 2) elastic net and 3) semantic EMR feature graph regularization from Chapter 4.

6.2.2 Results

In this section, we demonstrate the effect of autoencoder regularization on model performance and stability, and compare with our baselines. The prediction models for heart failure readmission were derived from $3,338$ features extracted from hospital database. The self taught learning stage during autoencoder training used an augmented $2,840$ diabetic admissions with 558 features that were common in both cohorts. A grid search for the best hyper-parameter setting resulted in $\alpha = .001$, $\lambda_{\text{en}} = .01$, $\lambda_{\text{graph}} = .03$ for the baseline models, and $\alpha = .005$, $\lambda_{\text{AE}} = 3000$, $\lambda_{\text{graph}} = 0.3$ for our autoencoder regularized models. Table 6.1 compares the model performance with different stabilizations schemes.



(a) Feature correlations from raw data

(b) Feature correlations discovered using autoencoder

Figure 6.5: Visualizing data correlation: Correlation matrix is calculated using absolute values of Pearson's correlation among EMR features. Denser matrix indicates higher correlation.

6.2.2.1 Capturing Higher Order Correlations

The efficacy of autoencoder network to model higher order correlations was verified by comparing the correlation matrices of raw data and data from the encoding layer. As illustrated in Fig. 6.5, the autoencoder derived correlation matrix was denser (matrix mean = 0.19) than the correlation matrix for raw data (matrix mean = 0.05). Further, comparing the entropy of these matrices, we found autoencoder derived correlation matrix had significantly higher entropy of $I = 2.45$, when compared to correlation matrix for raw data with $I = 0.22$. Hence the autoencoder was able to capture higher levels of correlation resulting in more information.

6.2.2.2 Effect on Model Sparsity

Table 6.2 provides a summary of the effects of stabilization schemes on model sparsity. Autoencoder regularization resulted in sparser models with no loss in performance. Model performance was measured using area under the ROC curve (AUC). For autoencoder regularization, AUC critically depended on the choice of autoencoder penalty (λ_{AE}) and number of hidden units (see Fig. 6.6). A maximum AUC of 0.65 was obtained for AG-Lasso-Autoencoder-Graph model with 20 hidden units and hyper-parameters as

Regularization	Features Selected (%)
Lasso	550 (16.5 %)
Elastic Net	753 (22.6 %)
Lasso-Graph	699 (20.9 %)
Lasso-Autoencoder	513 (15.4 %)
Lasso-Autoencoder-Graph	503 (15.1 %)
AG-Lasso-Autoencoder-Graph	412 (12.3 %)

Table 6.2: Effect of stabilization methods on model sparsity

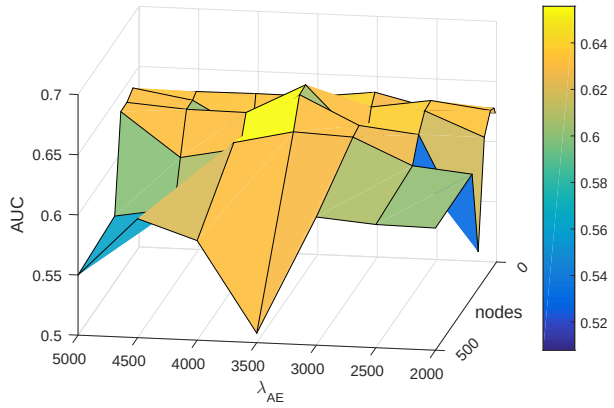


Figure 6.6: Effect of number of hidden units (nodes) and autoencoder penalty (λ_{AE}) on AUC. Lasso parameter fixed at $\alpha = .005$

$\alpha = .005$, $\lambda_{\text{en}} = .03$, $\lambda_{\text{graph}} = .3$, $\lambda_{\text{AE}} = 3000$.

The top predictors identified by our model are given in Table 6.3. The features were ranked based on importance measure calculated as described in Section 4.4.1.

6.2.3 Effect on Stability

When compared to λ_{AE}, the choice of hidden units had more influence on feature stability (see Fig. 6.7). Consistency index measurements for feature selection stability is reported in Fig. 6.8. In general, capturing higher order correlations using autoencoder improved feature stability when compared to baselines. Even though pure autoencoder regularization proved to be more effective for larger feature sets (> 120), the

Top predictors	Feature Importance
Age > 80	21
Past Emergency visits in 0-3 months	17.4
Past heart failures in 0-3 months	17.1
Past hospital admissions in 0-6 months	12.7
Past occurrence: congestive heart failure	11.7
Past occurrence: renal failure	10.3
Past occurrence: hypertension	10.3
Past occurrence: Acute kidney failure	10
Male	9.6
Past diagnosis: Angina pectoris	7.7
Past diagnosis: Pleural effusion	7.5
Past diagnosis: Type 2 diabetes mellitus	7.4
Personal history of certain other diseases	7.2

Table 6.3: Top predictors for 6-month unplanned re-hospitalization following heart failure discharges as identified by our autoencoder regularized linear model. Feature importance was calculated as product of feature weight and feature standard deviation in the training data set.

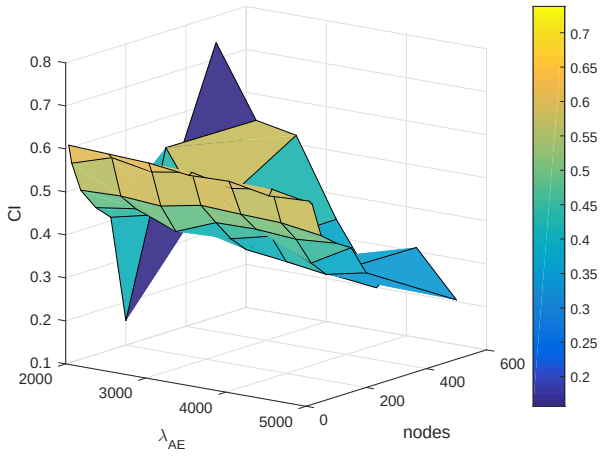


Figure 6.7: Effect of number of hidden units (nodes) and autoencoder penalty (λ_{AE})feature stability measured by consistency of top 100 features. Lasso parameter fixed at $\alpha = .005$

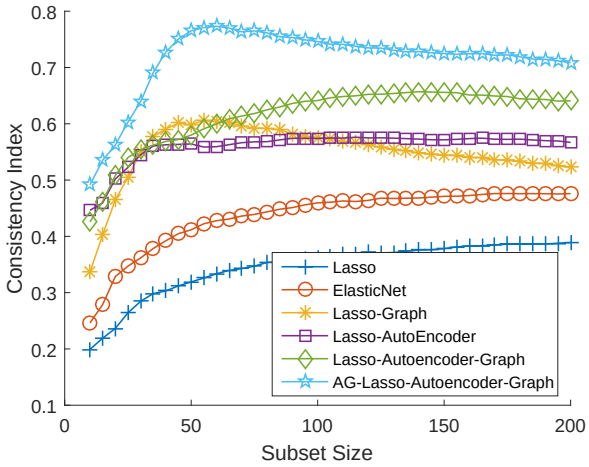


Figure 6.8: Feature stability as measured using Consistency Index. The plot compares similarity in feature subsets generated by our proposed models and baselines. Higher values indicate more stability.

combination of autoencoder and graph regularization consistently outperformed the baselines. Further, augmenting external cohort to autoencoder learning resulted in the most stable features. Similar observations were made when measuring model estimation stability. Fig. 6.9 reports the signal-to-noise ratios of the top 50 individual features. At 95% CI (approximately 1.96 std), lasso regularization identifies 3 features, elastic net identifies: 21, Graph regularization: 24, while the autoencoder regularized models identify all the 50 features. The variance in feature weight is greatly reduced by AG-Lasso-Autoencoder-Graph regularization.

6.3 Discussion and Conclusion

Higher order correlations in data have been studied in the past for pattern classification (Taylor and Coombes, 1993), finding associations among gene networks (Qian et al., 2009; Zhang et al., 2010), and distributed programming (Kannan et al., 2014). In this chapter, we utilize higher order correlations to stabilize a high dimensional sparse clinical model. Sparsity and stability are two important characteristics of interpretable healthcare. Sparsity promotes interpretability and stability inspires confidence in the prediction model.

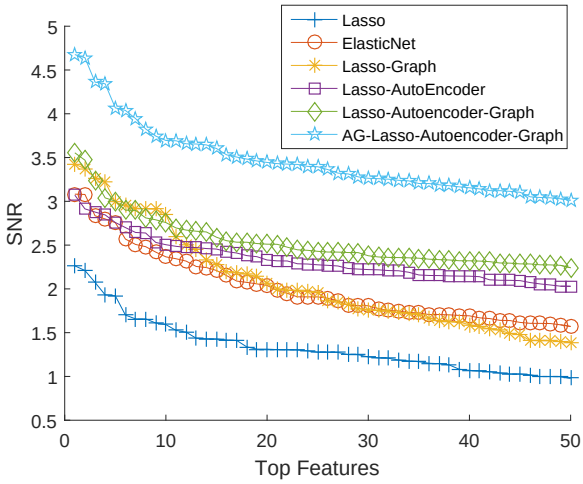


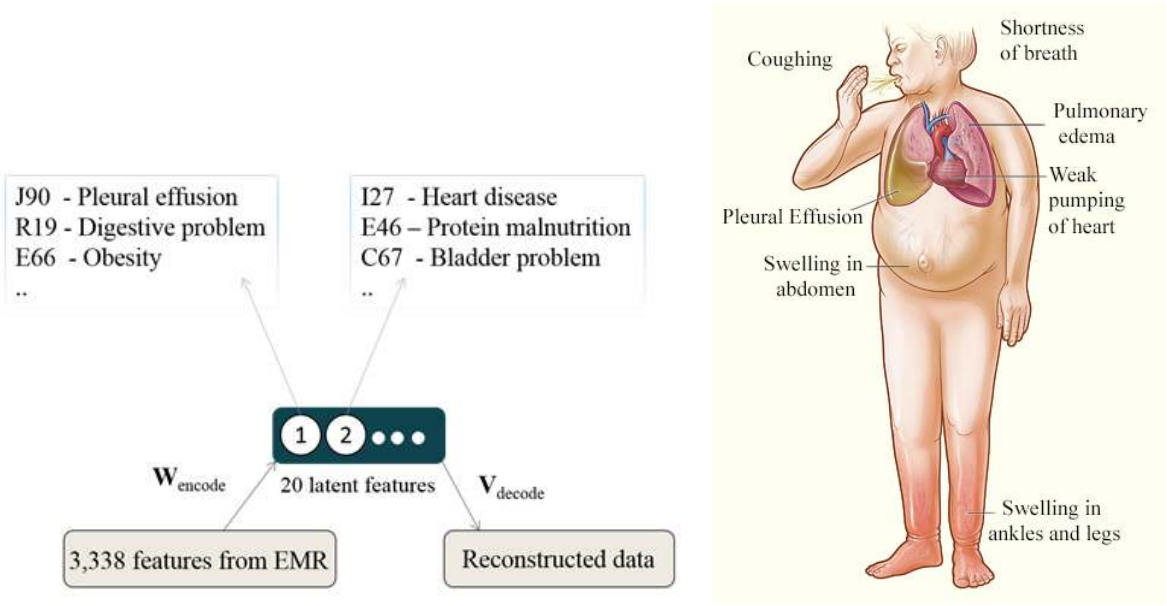
Figure 6.9: Model stability as measured using signal-to-noise ratio (SNR) of feature weights. Higher values indicate more stability.

Traditionally, autoencoder variants are used to improve prediction/classification accuracy. We have demonstrated a novel use of autoencoders – to stabilize high dimensional clinical prediction. We have achieved this by factorising the linear model parameter into a lower dimensional stable component, and a higher dimensional matrix that encapsulates all orders of correlation in data. The autoencoder was used to model these correlations using the encoding weights in the neural net. During network training, the interaction between the layers result in higher-order couplings, resulting in a low dimensional representation (Köster et al., 2014).

The encoding weights are responsible for mapping data from original space in the input layer to a reduced space in the hidden layer. Our input data contained 3338 features and we used a hidden layer with 20 nodes. An illustration of top features that were found to be correlated in the first two hidden nodes is shown in Figure 6.10(a). The final predictors chosen by our model are consistent with current medical literature, as shown in Figure 6.10(b) (adapted from website of National, Heart, Lung and Blood Institute[2]).

When looking at the top predictors (Table 6.3), our model selected old age (>80) as one of the most important predictors. This is consistent with a recent study by Muzzarelli et al. (2010), who concluded that elderly patients required frequent readmissions. Strong predictors also included past history of hospitalizations (past emergency visits,

[2]https://www.nhlbi.nih.gov/health/health-topics/topics/hf/signs



(a) (b)

Figure 6.10: Extracting higher order correlations in heart failure cohort. Top features associated in the first 2 hidden nodes is shown in (a). Common symptoms of heart failure is shown in (b).

ward transfers and direct admissions), consistent with the findings of (Chin et al., 1997; Krumholz et al., 2000, 1997; Felker et al., 2004; Amarasingham et al., 2010). Males were also found to be at increased risk of readmission, which is corroborated by recent studies (Chin et al., 1997; Krumholz et al., 1997; Amarasingham et al., 2010). Among comorbidities, renal failure, hypertension and kidney failure were strong predictors. Personal history of diseases were also ranked high by our model.

The predictive performance of our proposed model (as measured by AUC) is comparable with existing studies (Ross et al., 2008; Betihavas et al., 2012). We have demonstrated that the encoding process of an autoencoder, though intrinsically unstable, can be applied to regularize sparse linear prediction resulting in more stable features. The encoding weights capture higher level correlations in EMR data. When collecting data becomes expensive, augmenting another cohort during autoencoder training resulted in a more robust estimation of encoding weights, translating to better stability. This approach belongs to the emerging learning paradigm of self-taught learning (Raina et al., 2007). We believe this work presents interesting possibilities in the application of deep nets for model stability.

"For even the very wise cannot see all ends."

The Fellowship of the Ring, J.R.R.Tolkien

Chapter 7

Conclusion

STABLE prediction is often overlooked in favour of performance. Yet, stability prevails as key when adopting models in critical areas as healthcare. Stability facilitates reproducibility between model updates and generalization across medical studies. Stable models also aid meta-analysis, which combines analytic results from similar studies (Haidich, 2011).

This thesis set out to investigate model stability, characterized as feature subset stability and parameter estimation stability, for linear models derived from EMR data. Our contribution is in understanding the need for stable prediction, when much research has been dedicated to improving performance. For critical applications like healthcare, where data is sparse and redundant, stable features and estimates are necessary to lend credence to the model and its performance.

In this thesis, we have proposed three broad stabilization techniques that are fully automated. All models were derived from freely available administrative and medical data. This makes them portable to other cohorts or institutions using similar EMR systems. In the following sections, we summarize and discuss our scientific contributions of Chapters 3 through 6. We then provide directions for future research.

In Chapter 3, we presented a case study on instability using patient flow forecasting. To the best of our knowledge, this is the first work in forecasting next day discharges from a ward with no real-time clinical data. When comparing 3 linear and 4 non-linear

prediction models, we demonstrated that better performance may not guarantee better generalization. The instability in model parameters (we ignore hyper-parameters) poses a serious challenge for clinical acceptance of the model. The models in this chapter were derived purely from administrative data in hospital records. However, for prognosis, we need to include clinical, procedural and pathological information. This adds thousands of features to training data, which in turn necessitates parsimonious models to aid interpretation, visualization, faster computation and efficient storage. We resort to the efficient lasso regularization for simultaneous overfitting control and feature selection. The application of lasso guarantees sparsity irrespective of the number of irrelevant features in the training data. Since inducing sparsity invites instability (Xu et al., 2012), we propose three approaches to stabilize high-dimensional clinical prediction.

We presented our first approach in Chapter 4. Our approach consists of a novel technique to mitigate the stability problem by utilizing feature graphs that link similar conditions and the same condition of multiple time periods. To reduce variance in the selected features that are predictive, we introduced Laplacian-based regularization into a regression model. The Laplacian is derived on a feature graph that captures both the temporal and hierarchic relations between hospital events, diseases and interventions. Laplacian feature graphs have been used in bioinformatics , but not in healthcare. The model can be widely applicable and readily deployed in existing health information systems. We believe our framework provides the first proof of concept in utilizing feature graphs and numerically validating stability for a clinical prediction model.

In Chapter 5, we examined inherent statistical and structural relationships in routinely collected electronic medical records to propose two stabilization schemes.Using a sparse Cox model as basis for prediction, we achieved stability using random walk transformation of a feature graph. The feature graph was constructed with nodes as EMR features and edges as relationship between the features. We focussed on two types of feature relationships: (i) statistical similarity (ii) aggregate of statistical and semantic similarity. The Jaccard index was used to measure statistical similarity, while semantic similarity was derived from temporal and structural ICD-10 diagnosis tree relations among EMR features. Our experiments were conducted on 2 real world hospital datasets: a heart failure cohort ($1,784$ index admissions) and a diabetes cohort ($2,370$ index admissions). The Jaccard graph regularization proved to be the best for stabilizing parameter weights, whereas aggregate Jaccard scores and semantic EMR graph was superior in stabilizing feature subsets. Transferring Jaccard scores from a related cohort also improved

stability when compared with lasso and elastic net.

Finally we exploited higher order correlations for model stabilization in Chapter 6. Here, we demonstrated a novel formulation for the linear model parameter by factorizing it into (i) a lower order vector, which is stable and easy to learn, and (ii) a higher dimensional matrix that encapsulates all orders of correlation in data. By modelling this higher order component as the encoding weights of a classical autoencoder, and using it to regularize model learning, we achieved stability in feature subsets and weights.

All our regularization methods were posed as constrained optimization problems, and were solved using L-BFGS method (Liu and Nocedal, 1989). Hence the time complexity is linear with respect to the number data points. Our proposed schemes improved feature stability by over 20% when compared to the traditional methods, and offered marginal improvement in classification performance. A similar observation was made by Kalousis et al. (2007) in their study on high dimensional feature selection stability. This study compared the stability of five popular feature selection algorithms on 11 datasets taken from three different application domains and came to the following conclusion: "*Stability provides an objective criterion on which we can base our choice of feature selection algorithm in the absence of any significant difference in classification performance. Selecting the most stable algorithm, we have a higher confidence in the quality of the features that it selects but also a higher confidence in the corresponding classification performance*"(Kalousis et al. (2007), pp.113).

A general question often asked is "*What would happen with these methods if we have more data*"? With more data, model instability becomes less severe, as instability is partly caused by lack of data (more precisely, labels, other sources may be because of redundancies or noise in the labels & data collection). However, finer feature extraction rules could result in growing number of features with data size, inviting instability. In such cases, our proposed methods continue to remain effective.

7.1 Future Work

The stabilization techniques in this thesis open up interesting avenues for future work. The simplest extension of our work would be external validation. We have validated

our methods on heart failure and diabetic cohort from a single regional hospital. External validation of these methods on other cohorts possibly from other care centres would reveal interesting generalization properties in terms of stability and performance. This could be combined with more theoretical analysis on the effect of stabilization on confidence intervals of performance measures.

In Chapters 4 and 5, feature graph regularization has proven to be effective. The knowledge driven graph in Chapter 4 was constructed from two types of relations. It would be interesting to see which of the two relations – temporal or hierarchical – has more influence over feature stability. Further, in Chapter 5, Jaccard similarity graph proved to be more effective when compared to other statistical measures. A possible extension would be to investigate the different properties of the constructed feature graphs with its effects on stabilization.

Chapter 6 used autoencoder regularization to capture higher order correlations in patient data. It would be interesting to see other autoencoder variants to this purpose, opening interesting avenues in the application of deep learning.

This thesis primarily addresses the problem of model instability due to data correlation. Model instability can also happen due to missing data, which is quite common in data from EMRs. In patient records, data may be missing due to unknown past history, or due to error in documentation (Wells et al., 2013). Preprocessing the data using available popular techniques (Wells et al., 2013; Saha et al., 2015) could help to reduce instability.

Finally, we conclude by proposing two more approaches to counter instability due to data redundancy. Data redundancy can happen due to duplicate entries, or recording of the same clinical events in different formats. We hypothesize that artificially perturbing the features would help weed out weak features during model learning process. Specifically we propose the following experiments: (1) randomly drop features (dropout noise Section 7.1.1) during each bootstrap run (2) directly derive a learning model from data artificially corrupted using blankout noise (Section 7.1.2). We briefly expand on these techniques in the following sections.

7.1.1 Dropout

Dropout is a feature noising scheme that promotes model generalization by artificially corrupting the training data. When first introduced, dropout promoted model generalization by randomly omitting feature subsets during each iteration in training (Hinton et al., 2012). Since it prevents overfitting, dropout can be considered as a regularization technique along with parameter shrinkage methods and model averaging. In dropout training, the training data x_i is converted to $\tilde{x}_i$ using dropout noise by setting $x_{ij} = 0$ with a probability δ, and $x_{ij} = x_{ij}/(1-\delta)$ with a probability $1-\delta$ (Wager et al., 2013). For linear regression, dropout training is found to be equivalent to ridge regression (Wang and Manning, 2013). For logistic regression, dropout regularization favoured strong predictors and parameter shrinkage (Wager et al., 2013).

7.1.2 Learning with Marginalized Corrupted Features

Ideally, a good training set should represent all the variations in the data. However, this is not possible all the time. An interesting development (Maaten et al., 2013) is to artificially corrupt the training data to introduce variations, thereby enhancing the generalization of the model. Since EMR data is characterized by correlations, redundancy and high degree of missingness, we use blankout feature corruption characterized as:

$$p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \prod_{d=1}^{D} p(\tilde{x}_d|x_d;\, q_d) \tag{7.1}$$

where the d^{th} feature is randomly set to zero with the probability q_d. We assume that the corrupting distribution is unbiased with $\mathbb{E}(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \boldsymbol{x}$. Each element x_n in the training set, $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$is corrupted M times using (7.1) to obtain $\tilde{x}_{nm}$observations (where $m = 1, \cdots, M$). The loss function of the extended dataset $\tilde{\mathcal{D}}$ becomes:

$$\mathcal{L}(\tilde{\mathcal{D}}, \Theta) = \sum_{n=1}^{N} \frac{1}{M} \sum_{m=1}^{M} L\left(\tilde{x}_{nm}, y_n; \Theta\right),$$

where: $\tilde{x}_{nm} \sim p(\tilde{x}_{nm}|x_n)$, Θ is the model parameters, and $L(x, y; \Theta)$ is the loss function of the model. When $M \to \infty$, we have:

$$\mathcal{L}(\mathcal{D}, \Theta) = \sum_{n=1}^{N} \mathbb{E}\left(L\left(\tilde{x}_n, y_n; \Theta\right)\right) \tag{7.2}$$

Logistic Loss and Blankout Noise Given $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ be the training dataset in which $\boldsymbol{x}_n \in \mathbb{R}^p$ denotes the high-dimensional feature vector of data instance n and $y_n \in \{+1, -1\}$ is the outcome label. The logistic loss function becomes:

$$Loss = \sum_{n=1}^{N} \log\left(1 + \exp\left(-y_n(w^T x_n)\right)\right) \tag{7.3}$$

where $w \in \mathbb{R}^p$ are the feature weights. Using logistic loss (7.3)) in (7.2), we have:

$$\mathcal{L}(\mathcal{D}, w) \leq \sum_{n=1}^{N} \log\left\{1 + \prod_{d=1}^{D} \mathbb{E}\left(\exp(-y_n w_d x_{nd})\right)_{p(\tilde{x}_{nd}|x_{nd})}\right\} \tag{7.4}$$

Here, in (7.4), we can think of $\mathbb{E}\left(\exp(-y_n w_d x_{nd})\right)$ as a moment generating function $\mathbb{E}\left(\exp(t_{nd} x_{nd})\right)$ with $t_{nd} = -y_n w_d$ (Maaten et al., 2013). The moment-generating function (MGF) of corrupting distribution can be used here. For blankout noise, we can write the probability mass function (PMF) as:

$$f_X(\tilde{x}_{nd}) \quad = \quad \begin{cases} q_d & \text{when } \tilde{x}_{nd} = 0 \\ 1 - q_d & \text{when } \tilde{x}_{nd} = \frac{1}{1-q_d} x_{nd} \end{cases}$$

Hence, moment generating function using blankout noise becomes:

$$\mathrm{M}_x(\text{blankout}) = q_d + (1 - q_d)\exp(-y_n w_d \frac{1}{1 - q_d} x_{nd}) \tag{7.5}$$

The logistic regression model can be derived from data marginally corrupted using blankout noise by plugging (7.5) in (7.4) to give:

$$\mathcal{L}(\mathcal{D}, w) \leq \sum_{n=1}^{N} \log\left\{1 + \prod_{d=1}^{D} \left(q_d + (1 - q_d)\exp(\frac{-y_n x_{nd}}{1 - q_d} w_d)\right)\right\} \tag{7.6}$$

Survival Loss and Blankout Noise Cox regression models the readmission time directly and takes censored information into account. Let k observations be uncensored and $N-k$ be right censored. Let $t_{(i)}$ be the i^{th} ordered unique failure time. Let $R(t_{(i)})$ is the risk at time $t_{(i)}$. Hence $R(t_{(i)})$ consists of all persons surviving up to $t_{(i)}$. Using Jensen's inequality, we can write the Cox loss as:

$$\mathcal{L}(\tilde{\mathcal{D}}, \boldsymbol{w}) \leq \sum_{n=1}^{k} \log \left\{ \sum_{l \epsilon R(t_{(n)})} \prod_{d=1}^{D} \mathbb{E}\left[\exp(w_d \tilde{x}_{ld})\right] \right\} - \sum_{n=1}^{k} \sum_{d=1}^{D} \mathbb{E}\left(w_d \tilde{x}_{(n)d}\right)$$

The Cox model can be derived from data marginally corrupted using blankout noise (7.5). The modified loss function becomes:

$$\mathcal{L}(\tilde{\mathcal{D}}, \boldsymbol{w}) \leq \sum_{n=1}^{k} \log \left\{ \sum_{l \epsilon R(t_{(n)})} \prod_{d=1}^{D} \left[q_d + (1-q_d) \exp(w_d \frac{1}{1-q_d} x_{ld}) \right] \right\} - \sum_{n=1}^{k} \sum_{d=1}^{D} w_d x_{(n)d}$$

We detail the derivations of loss functions corrupted with blankout noise and their gradients in Appendix Section A.2.1.

Appendix A

Supplementary Material

A.1 Parameter Estimation for Logistic regression

Given a list of input variables $X \in \mathbb{R}^{M \times N}$ with corresponding output labels $y \in \{0, 1\}$, the logistic regression model with parameter $\boldsymbol{w} \in \mathbb{R}^N$ becomes:

$$J(\boldsymbol{w}) = \sum_{i=1}^{M} - y^{(i)} \log h_{\boldsymbol{w}}(x^{(i)}) - (1 - y^{(i)}) \log \left(1 - h_{\boldsymbol{w}}(x^{(i)})\right) \tag{A.1}$$

Thus, $J(\boldsymbol{w})$ becomes the cost function. The maximum likelihood estimate for $\boldsymbol{w}$ is obtained by minimizing the cost function with respect to $\boldsymbol{w}$. Hence we have:

$$\min_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{d}{d\boldsymbol{w}} J(\boldsymbol{w}) = \frac{d}{d\boldsymbol{w}} \sum_{i=1}^{M} - y^{(i)} \log h_{\boldsymbol{w}}(x^{(i)}) - (1 - y^{(i)}) \log \left(1 - h_{\boldsymbol{w}}(x^{(i)})\right)$$

If we consider only a single training example (x, y), we have:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{w}_j} \ell(\boldsymbol{w}) &= \frac{\partial}{\partial \boldsymbol{w}_j} \left[-y \log h_{\boldsymbol{w}}(x)\right] - \frac{\partial}{\partial \boldsymbol{w}_j} \left[(1 - y) \log \left(1 - h_{\boldsymbol{w}}(x)\right)\right] \\ &= \frac{\partial}{\partial \boldsymbol{w}_j} \left[-y \log g(\boldsymbol{w}^T x)\right] - \frac{\partial}{\partial \boldsymbol{w}_j} \left[(1 - y) \log \left(1 - g(\boldsymbol{w}^T x)\right)\right] \\ &= \left[-y \frac{1}{g(\boldsymbol{w}^T x)} + (1 - y) \frac{1}{(1 - g(\boldsymbol{w}^T x))}\right] \frac{\partial g(\boldsymbol{w}^T x)}{\partial \boldsymbol{w}_j} \end{aligned} \tag{A.2}$$

The derivative of the sigmoid function $g(z)$ can be written as:

$$\begin{aligned}\frac{d}{dz}g(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\ &= \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right) \\ &= g(z)\left(1-g(z)\right) \end{aligned} \tag{A.3}$$

Substituting (A.3) in (A.2), we have:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{w}_j}\ell(\boldsymbol{w}) &= \left[-y\frac{1}{g(\boldsymbol{w}^T x)}+(1-y)\frac{1}{(1-g(\boldsymbol{w}^T x))}\right] g(\boldsymbol{w}^T x)\left(1-g(\boldsymbol{w}^T x)\right)\frac{\partial g(\boldsymbol{w}^T x)}{\partial \boldsymbol{w}_j} \\ &= \left[h_{\boldsymbol{w}}(x)-y\right] x_j \end{aligned} \tag{A.4}$$

To summarize, the logistic regression learns parameter $\boldsymbol{w}$, by minimizing the objective function in (A.1). The objective function is guaranteed to be convex and the optimum can be found with the help of (A.4) using gradient descent or conjugate gradient method.

A.2 Estimating parameters of a Cox proportional hazards model

Let the number of individuals be n. Let k observations be uncensored and $n-k$ be right censored. Let $t_{(i)}$ be the i^{th} ordered unique failure time. Let $R(t_{(i)})$ is the risk at time $t_{(i)}$. Hence $R(t_{(i)})$ consists of all persons surviving up to $t_{(i)}$. The partial likelihood for the model is calculated as:

$$\mathcal{L}(\beta) = \prod_{i=1}^{k}\frac{exp(\beta^T x_i)}{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l)} \tag{A.5}$$

and the log-partial likelihood becomes

$$\begin{aligned} l(\beta) &= \log \mathcal{L}(\beta) \\ &= \sum_{i=1}^{k} \left\{ \beta^T x_{(i)} - log \left[\sum_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right] \right\} \end{aligned} \tag{A.6}$$

Maximizing the likelihood, we have:

$$\begin{aligned} \frac{\partial l}{\partial \beta_u} &= \sum_{i=1}^{k} \left\{ \frac{\partial}{\partial \beta_u} \beta^T x_{(i)} - \frac{\partial l}{\partial \beta_u} log \left[\sum_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right] \right\} \\ &= \sum_{i=1}^{k} \left\{ x_{u(i)} - \frac{\sum\limits_{l \epsilon R(t_{(i)})} x_{ul}\, exp(\beta^T x_l)}{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l)} \right\} = 0 \end{aligned} \tag{A.7}$$

We can simplify (A.7) as:

$$\frac{\partial l}{\partial \beta_u} = \sum_{i=1}^{k} \Big\{ x_{u(i)} - A_{ui}(\beta) \Big\} \tag{A.8}$$

where:

$$A_{ui}(\beta) = \frac{\sum\limits_{l \epsilon R(t_{(i)})} x_{ul}\, exp(\beta^T x_l)}{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l)} \tag{A.9}$$

Taking the second derivative, we have :

$$\frac{\partial l}{\partial \beta_u \beta_v} = -\sum_{i=1}^{k} \left(\frac{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \, \frac{\partial l}{\partial \beta_v} \left[\sum\limits_{l \epsilon R(t_{(i)})} x_{ul} \, exp(\beta^T x_l) \right]}{\left[\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right]^2} \right.$$

$$\left. - \frac{\sum\limits_{l \epsilon R(t_{(i)})} x_{ul} \, exp(\beta^T x_l) \, \frac{\partial l}{\partial \beta_v} \left[\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right]}{\left[\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right]^2} \right)$$

$$= -\sum_{i=1}^{k} \left(\frac{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \left[\sum\limits_{l \epsilon R(t_{(i)})} x_{ul} x_{vl} \, exp(\beta^T x_l) \right]}{\left[\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right]^2} \right.$$

$$\left. - \frac{\sum\limits_{l \epsilon R(t_{(i)})} x_{ul} \, exp(\beta^T x_l) \left[\sum\limits_{l \epsilon R(t_{(i)})} x_{vl} exp(\beta^T x_l) \right]}{\left[\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right]^2} \right)$$

$$\frac{\partial l}{\partial \beta_u \beta_v} = -\sum_{i=1}^{k} \left(- \frac{\left[\sum\limits_{l \epsilon R(t_{(i)})} x_{ul} x_{vl} \, exp(\beta^T x_l) \right]}{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l)} + \right.$$

$$\left. \frac{\sum\limits_{l \epsilon R(t_{(i)})} x_{ul} \, exp(\beta^T x_l) \left[\sum\limits_{l \epsilon R(t_{(i)})} x_{vl} exp(\beta^T x_l) \right]}{\left[\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l) \right]^2} \right) \quad \text{(A.10)}$$

We can simplify this using notation in equation(A.9) as:

$$\frac{\partial l}{\partial \beta_u \beta_v} = \sum_{i=1}^{k} \left\{ -\frac{\sum\limits_{l \epsilon R(t_{(i)})} x_{ul} x_{vl}\, exp(\beta^T x_l)}{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta^T x_l)} + A_{ui}(\beta) A_{vi}(\beta) \right\} \tag{A.11}$$
$$= C_{uvi}(\beta)$$

A.2.0.1 Breslow's estimator for baseline cumulative hazard function

We need to estimate the cumulative baseline hazard function and also the baseline survival function. Assuming that the baseline hazard function is constant between successive observed failure times, the Breslow's estimator (ignoring tied survival times) is given by:

$$\widehat{H_0}(t) = \sum_{t_i \leq t} \frac{1}{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta x_l)}$$

where, t_i is unique ordered failure time. For a discrete distribution, we have:

$$\widehat{h_0}(t) = \frac{1}{\sum\limits_{l \epsilon R(t_{(i)})} exp(\beta x_l)}$$

where $\widehat{h_0}(t) = 0$ if t is not an event time.

The Survival Function Estimator

The estimator for the baseline survival function becomes:

$$\widehat{S_0}(t) = e^{-\widehat{H_0}(t)}$$

The survival probability $S(t|x_i)$ of the i^{th} patient at time t becomes:

$$S(t|x_i) = S_0(t)^{exp(\beta^T x_i)}$$

A.2.1 Learning with Marginalized Corrupted Features

A.2.1.1 Logistic Loss and Blankout Noise

Given $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$ be the training dataset in which $\boldsymbol{x}_n \in \mathbb{R}^p$ denotes the high-dimensional feature vector of data instance n and y_n is the outcome label. We can model the probability of label $y \in \{+1, -1\}$, given data $x \in \mathbb{R}^p$as:

$$P(y|x; w) = \frac{1}{1 + \exp\left(-y(w^T x)\right)}$$

where $w \in \mathbb{R}^p$ are the feature weights. Hence the likelihood becomes:

$$\begin{aligned} \mathcal{L}(w; x, y) &= \prod_{n=1}^{N} \frac{1}{1 + \exp\left(-y_n(w^T x_n)\right)} \\ \log\mathcal{L}(w; x, y) &= -\sum_{n=1}^{N} \log\Big(1 + \exp\Big(-y_n(w^T x_n)\Big)\Big) \end{aligned}$$

Maximizing the likelihood is equivalent to minimizing the negative log likelihood. Hence the logistic loss function becomes:

$$\text{Loss} = \sum_{n=1}^{N} \log\Big(1 + \exp\Big(-y_n(w^T x_n)\Big)\Big) \tag{A.12}$$

Using logistic loss (A.12) in(7.2), we have:

$$\mathcal{L}(\mathcal{D}, w) = \sum_{n=1}^{N} \mathbb{E}\left[\log\left(1 + \exp(-y_n w^T x_n)\right)\right]_{p(\tilde{x}_n|x_n)}$$
$$\mathcal{L}(\mathcal{D}, w) \leq \sum_{n=1}^{N} \log\left\{1 + \prod_{d=1}^{D} \mathbb{E}\left(\exp(-y_n w_d x_{nd})\right)_{p(\tilde{x}_{nd}|x_{nd})}\right\} \tag{A.13}$$

In (A.13), we can think of $\mathbb{E}\left(\exp(-y_n w_d x_{nd})\right)$ as a moment generating function $\mathbb{E}\left(\exp(t_{nd} x_{nd})\right)$ with $t_{nd} = -y_n w_d$ (Maaten et al., 2013). The MGF of corrupting distribution can be used here.

For blankout noise, we can write the PMF as:

$$f_X(\tilde{x}_{nd}) \quad = \quad \begin{cases} q_d & \text{when } \tilde{x}_{nd} = 0 \\ 1 - q_d & \text{when } \tilde{x}_{nd} = \frac{1}{1-q_d} x_{nd} \end{cases}$$

Hence, moment generating function using blankout noise becomes:

$$\begin{aligned} \mathrm{M}_x(t) &= \mathbb{E}\left[\exp(t\tilde{x}_{nd})\right], \text{ where t= } -y_n w_d \\ \mathrm{M}_x(t) &= \sum_{\tilde{x}_{nd}} \exp(t\tilde{x}_{nd}) \times f_X(\tilde{x}_{nd}) \\ &= \exp(0) \times q_d + \exp(t\frac{1}{1-q_d} x_{nd}) \times (1 - q_d) \end{aligned}$$

$$\mathrm{M}_x(\text{blankout}) = q_d + (1 - q_d)\exp(-y_n w_d \frac{1}{1-q_d} x_{nd}) \tag{A.14}$$

The logistic regression model can be derived from data marginally corrupted using blankout noise by plugging (A.14) in (A.13) to give:

$$\mathcal{L}(\mathcal{D}, w) \leq \sum_{n=1}^{N} \log\left\{1 + \prod_{d=1}^{D}\left(q_d + (1 - q_d)\exp(\frac{-y_n x_{nd}}{1-q_d} w_d)\right)\right\} \tag{A.15}$$

The gradient for loss function in (A.15) becomes:

$$\frac{\partial \mathcal{L}(\mathcal{D}, w)}{\partial w_d} = \frac{\partial}{\partial w_d}\left(\sum_{n=1}^{N} \log\left\{1 + \prod_{d=1}^{D}\left(q_d + (1 - q_d)\exp(\frac{-y_n x_{nd}}{1-q_d} w_d)\right)\right\}\right)$$

For sake of simplicity, let us assign:

$$B_{nd} = q_d + (1 - q_d) \exp(\frac{-y_n x_{nd}}{1 - q_d} w_d)$$

$$\partial w_d B_{nd} = -y_n x_{nd} \exp(\frac{-y_n x_{nd}}{1 - q_d} w_d)$$

$$\begin{aligned}
\frac{\partial \mathcal{L}(\mathcal{D}, w)}{\partial w_d} &= \frac{\partial}{\partial w_d} \left(\sum_{n=1}^{N} \log \left\{ 1 + \prod_{d=1}^{D} B_{nd} \right\} \right) \\
&= \sum_{n=1}^{N} \frac{-y_n x_{nd}}{1 + \prod\limits_{d=1}^{D} B_{nd}} \exp(\frac{-y_n x_{nd}}{1 - q_d} w_d) \prod_{\tilde{d} \neq d} B_{nd} \\
&= \sum_{n=1}^{N} \frac{-y_n x_{nd}}{B_{nd}} \frac{\prod\limits_{d=1}^{D} B_{nd}}{1 + \prod\limits_{d=1}^{D} B_{nd}} \exp(\frac{-y_n x_{nd}}{1 - q_d} w_d)
\end{aligned} \tag{A.16}$$

A.2.1.2 Cox Loss and Blankout Noise

Cox regression models the readmission time directly and takes censored information into account. Let k observations be uncensored and $N - k$ be right censored. Let $t_{(i)}$ be the i^{th} ordered unique failure time. Let $R(t_{(i)})$ is the risk at time $t_{(i)}$. Hence $R(t_{(i)})$ consists of all persons surviving up to $t_{(i)}$. The log-partial likelihood function for Cox regression model is:

$$\log \ell(w; \boldsymbol{x}) = \sum_{n=1}^{k} \left\{ w^T x_{(n)} - \log \left[\sum_{l \epsilon R(t_{(n)})} \exp(w^T x_l) \right] \right\}$$

Maximizing the likelihood is equivalent to minimizing the negative of log-likelihood. Hence the loss function becomes:

$$\mathcal{L}(\boldsymbol{x}; \boldsymbol{w}) = \sum_{n=1}^{k} \left\{ \log \left[\sum_{l \epsilon R(t_{(n)})} \exp(w^\top x_l) \right] - w^\top x_{(n)} \right\} \tag{A.17}$$

Using (A.17) in (7.2), we have:

$$\begin{aligned}
\mathcal{L}(\tilde{\mathcal{D}}, \boldsymbol{w}) &= \sum_{n=1}^{k} \mathbb{E}\left\{\log\left[\sum_{l \epsilon R(t_{(n)})} \exp(w^\top \tilde{x}_l)\right] - w^\top \tilde{x}_{(n)}\right\} \\
&= \sum_{n=1}^{k} \mathbb{E}\left\{\log\left[\sum_{l \epsilon R(t_{(n)})} \exp(w^\top \tilde{x}_l)\right]\right\} - \sum_{n=1}^{k} \mathbb{E}\left(w^\top \tilde{x}_{(n)}\right) \\
\mathcal{L}(\tilde{\mathcal{D}}, \boldsymbol{w}) &\leq \sum_{n=1}^{k} \log\left\{\mathbb{E}\left[\sum_{l \epsilon R(t_{(n)})} \prod_{d=1}^{D} \exp(w_d \tilde{x}_{ld})\right]\right\} - \sum_{n=1}^{k}\sum_{d=1}^{D} \mathbb{E}\left(w_d \tilde{x}_{(n)d}\right) \quad \text{(A.18)}
\end{aligned}$$

Here, (A.18) is the result of applying Jensen's inequality. We further simplify as:

$$\begin{aligned}
\mathcal{L}(\tilde{\mathcal{D}}, \boldsymbol{w}) &\leq \sum_{n=1}^{k} \log\left\{\sum_{l \epsilon R(t_{(n)})} \mathbb{E}\left[\prod_{d=1}^{D} \exp(w_d \tilde{x}_{ld})\right]\right\} - \sum_{n=1}^{k}\sum_{d=1}^{D} \mathbb{E}\left(w_d \tilde{x}_{(n)d}\right) \\
&\leq \sum_{n=1}^{k} \log\left\{\sum_{l \epsilon R(t_{(n)})} \prod_{d=1}^{D} \mathbb{E}\left[\exp(w_d \tilde{x}_{ld})\right]\right\} - \sum_{n=1}^{k}\sum_{d=1}^{D} \mathbb{E}\left(w_d \tilde{x}_{(n)d}\right) \quad \text{(A.19)}
\end{aligned}$$

Since we assume that the corrupting distribution is unbiased, we have:$\mathbb{E}[\tilde{x}_{nd}]_{p(\tilde{x}_{nd}|x_{nd})} = x_{nd}$

$$\begin{aligned}
\text{Hence, } \mathbb{E}\left(w_d \tilde{x}_{(n)d}\right) &= w_d \mathbb{E}\left(\tilde{x}_{(n)d}\right) \\
&= w_d \left[\left(\tilde{x}_{(n)d} \times \mathrm{P}(\tilde{x}_{(n)d} = 0)\right)_{\tilde{x}_{(n)d}=0}\right] \\
&\quad + w_d \left[\left(\tilde{x}_{(n)d} \times \mathrm{P}(\tilde{x}_{(n)d} = \frac{1}{1-q} x_{(n)d})\right)_{\tilde{x}_{(n)d}=\frac{1}{1-q}x_{(n)d}}\right] \\
&= w_d \left[0 + \left(\frac{1}{1-q} x_{(n)d} \times (1-q)\right)\right] \\
\mathbb{E}\left(w_d \tilde{x}_{(n)d}\right) &= w_d x_{(n)d} \quad \text{(A.20)}
\end{aligned}$$

Using (A.14) and (A.20) in (A.18) we have:

$$\mathcal{L}(\tilde{\mathcal{D}}, \boldsymbol{w}) \leq \sum_{n=1}^{k} \log\left\{\sum_{l \epsilon R(t_{(n)})} \prod_{d=1}^{D} \left[q_d + (1-q_d)\exp(w_d \frac{1}{1-q_d} x_{ld})\right]\right\} - \sum_{n=1}^{k}\sum_{d=1}^{D} w_d x_{(n)d}$$

The gradient of loss function becomes:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_d} &= \frac{\partial \mathcal{L}}{\partial w_d}\left(\sum_{n=1}^{k}\log\left\{\sum_{l\epsilon R(t_{(n)})}\prod_{d=1}^{D}\left[q_d+(1-q_d)\exp(w_d\frac{1}{1-q_d}x_{ld})\right]\right\}\right)\\ &\quad -\frac{\partial \mathcal{L}}{\partial w_d}\left(\sum_{n=1}^{k}\sum_{d=1}^{D}w_d x_{(n)d}\right)\\ &= \frac{\partial \mathcal{L}}{\partial w_d}(\text{Term}_1)-\frac{\partial \mathcal{L}}{\partial w_d}(\text{Term}_2)\end{aligned}$$

We now find the differentials of the two terms individually.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_d}(\text{Term}_2) &= \frac{\partial \mathcal{L}}{\partial w_d}\left(\sum_{n=1}^{k}\sum_{d=1}^{D}w_d x_{(n)d}\right)\\ &= \sum_{n=1}^{k}x_{(n)d}\end{aligned} \tag{A.21}$$

Let

$$B_{ld} = q_d + (1-q_d)\exp(\frac{x_{ld}}{1-q_d}w_d)$$

$$A_l = \prod_{d=1}^{D} B_{ld}$$

Here,

$$\begin{aligned}\partial w_d A_l &= \left(\prod_{\tilde{d}\neq d} B_{l\tilde{d}}\right)\exp(\frac{x_{ld}}{1-q_d}w_d)x_{ld}\\ &= \left(\prod_{d=1}^{D} B_{ld}\right)\frac{\exp(\frac{x_{ld}}{1-q_d}w_d)x_{ld}}{B_{ld}}\\ &= A_l\frac{\exp(\frac{x_{ld}}{1-q_d}w_d)x_{ld}}{B_{ld}}\end{aligned}$$

then:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_d}(\text{Term}_1) &= \frac{\partial \mathcal{L}}{\partial w_d}\left(\sum_{n=1}^{k} \log \sum_{l \epsilon R(t_{(n)})} A_l\right) \\
&= \sum_{n=1}^{k} \frac{1}{\sum\limits_{l \epsilon R(t_{(n)})} A_l} \partial w_d \left(\sum_{l \epsilon R(t_{(n)})} A_l\right) \\
&= \sum_{n=1}^{k} \frac{1}{\sum\limits_{l \epsilon R(t_{(n)})} A_l} \sum_{l \epsilon R(t_{(n)})} \partial w_d A_l \\
&= \sum_{n=1}^{k} \sum_{l \epsilon R(t_{(n)})} \frac{A_l}{\sum\limits_{l* \epsilon R(t_{(n)})} A_{l*}} \frac{\exp(\frac{x_{ld}}{1-q_d} w_d) x_{ld}}{B_{ld}} \\
&= \sum_{n=1}^{k} \sum_{l \epsilon R(t_{(n)})} Q(l) \frac{\exp(\frac{x_{ld}}{1-q_d} w_d)}{B_{ld}} x_{ld}
\end{aligned}$$

where,

$$Q(l) = \frac{A_l}{\sum\limits_{l* \epsilon R(t_{(n)})} A_{l*}}$$

Appendix B

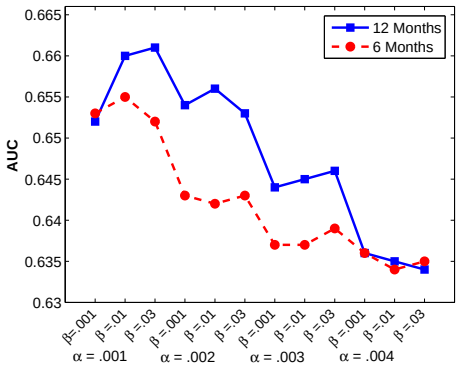
Additional Experiments

B.1 Effect of Knowledge-based Stabilization on heart failure readmission within 12 months

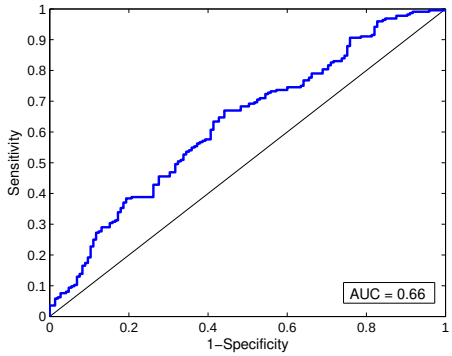
In Chapter 4, we applied knowledge-based feature graph (EMR graph) for stabilizing an individual patient predictive model for unplanned readmission within 6 months. We applied the same techniques for a 12 months readmission model. Feature extraction and graph generation techniques were as detailed in Chapter 4. We present the results on performance and stability for the 12 months model.

The AUC reaches the pick of 0.66 (95% CI: 0.60–0.71) when $\alpha = .001$ and $\beta = .01$. After a certain point, model discrimination gradually decreased with increasing regularization penalties (Figure B.1a). This suggests a trade-off between maintaining discriminative power, sparsity and stability. A good trade-off was achieved at $\alpha = .001$ and $\beta = .03$, where external validation resulted in an AUC 0.66 on 12-month prediction (Fig. B.1b).

The stability of the feature subset selected by our 12 month readmission model was numerically validated using measures of Consistency index and Jaccard index. The top ranked features of the model with and without Laplacian feature graph regularization were compared among each other for different bootstraps. The Laplacian-regularized



(a) Variation in AUC



(b) ROC curve for 12 months HF model

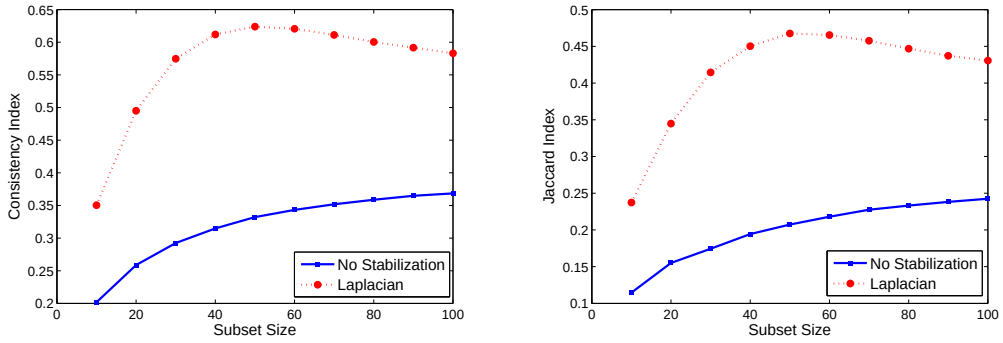
Figure B.1: Model performance for 12 months HF readmission. (a) Variation in AUC w.r.t different hyperparameter settings for models predicting HF specific re-hospitalization 6 months and 12 months. The X-axis represents ordinal values of increasing penalty in regularization variables. The Y-axis represents the AUC. (b) Receiver operating characteristic curve for our stabilized model predicting re-hospitalization in 6 months and 12 months with $\alpha = .001$ and $\beta = .03$.

model resulted in more stable feature subsets when measured using both indices, as demonstrated in Figures B.2a,b.

From a total of 3,338 features extracted from the EMR database, the lasso-regularized regression model resulted in 142 risk factors which are positively predictive of unplanned readmissions following heart failure discharges. We list the top predictors for 12-month re-hospitalizations in Table B.1.

B.2 Stabilization: Data driven experiments

In Chapter 5, we examined data driven stabilization schemes. We also experimented on stabilization using data perturbations. In this process, we introduce data perturbations to the model to weed out weak and inconsequential features. We had briefly mentioned two methods for data perturbations – dropout (Section 7.1.1) and learning with marginalized corrupted features (Section 7.1.2). Here, we introduce two more methods. We then present our initial results on stabilization with these methods.



(a) Consistency index. (b) Jaccard index.

Figure B.2: Stability of the model as measured by the Consistency index (see Figure B.2a) and Jaccard index (see Figure B.2b) for 12-month prediction. The plot compares the similarity in feature subsets generated by models with and without Laplacian EMR graph stabilization under data variations. The stability indices are evaluated at different sizes of the feature subsets. Larger value of indices implies more stability.

B.2.1 Augmenting training data

We apply the converse principle of dropout to our data. Instead of dropping out random data points, we now augment ("drop in") a random sub-sample of our original data during model learning. This augmenting process is similar to bootstrap, we try to see if such perturbations have any effect on model stability.

B.2.2 Adding Gaussian Noise

Training features by adding Gaussian noise is similar to L_2regularization (Bishop, 1995). Though adding noise to features may look counter-intuitive, this increases the variance in model parameters, thereby forcing feature selection algorithms to choose only the strong features.

B.2.3 Double Bootstrap

Breiman et al. (1996) studied instability in prediction and proposed bagging. Bagging is an ensemble technique based on bootstrapping (Efron and Tibshirani, 1994) and

Top predictors for 12 months readmission	Importance
Male	50.2
Public health insurance	47.2
Admissions past 2-4 years	31.6
Rare diagnoses past 0-3 months	30.8
Emergency visits past 0-3 months	29.3
Emergency attend time past 0-3 months	27.9
Emergency-to-ward transfers past 0-3 months	25.2
Procedures past 0-3 months	27.4
Occupations: pensioner, retired or home duties	25.8
Acute myocardial infarction	12.2
Disorders of lipoprotein metabolism	12.8
Angina pectoris	11.8
Personal history of certain other diseases	11.7
Complicated diabetes diagnoses past 3-6 months	14.2

Table B.1: Top predictors for 12-month unplanned re-hospitalization following heart failure discharges as identified by our model. Feature importance was calculated as product of feature weight and feature standard deviation in the training data set, normalized into the range [0–100].

aggregating. In theory, if a base classifier could perform better than random guessing, then aggregating results from an ensemble of classifiers would always outperform the base classifier. In bagging, B bootstrap samples are generated from the original training set. For each of the B samples, a learning model is derived, resulting in B independent models. A test instance is estimated using a majority vote from the B models or by averaging the model parameters. The bagging algorithm is described by Breiman in (Breiman et al., 1996). In our work, we implement bagging as a double bootstrap shown in algorithm B.1.

On average, a bootstrap sample will not contain around 37% of training data. Hence sampling with replacement could possibly avoid potential outliers, resulting in better classifiers (Skurichina and Duin, 2002). Aggregating results reduces their variance and hence increases stability. In our algorithm, we do two bootstrap aggregations. The inner bootstrap resamples from an already bootstrapped sample and derives the model parameters. Hence, each iteration of the outer for loop produces a $\bar{\boldsymbol{w}}_i$: the aggregated

Algorithm B.1 Double Bootstrap: variation of Bagging for predicting HF readmission

1. **for** i = 1,2,...,B_{outer}
2. Generate a bootstrap replicate D_i from $\mathcal{D}_{train}$
3. **for** j = 1,2,...,B_{inner}
4. Generate a bootstrap replicate D_j from D_i
5. Derive model parameters $\boldsymbol{w}_j$ using D_j
6. **end for**
7. Compute $\bar{\boldsymbol{w}}_i = \frac{\sum_j \boldsymbol{w}_j}{B_{inner}}$
8. Compute $\mathrm{P}(\boldsymbol{y}_i|D_{test}) = \mathtt{Model}(D_{test}, \bar{\boldsymbol{w}}_i)$
9. **end for**
10. Compute $\mathrm{P}(\boldsymbol{y}|\mathcal{D}_{test}) = \frac{\sum_i \mathrm{P}(\boldsymbol{y}_i|D_{test})}{B_{outer}}$

model parameters from B_{inner} bootstrapped samples.

B.2.4 Results for Data Perturbation Methods

We compared the predictive performance and stability of our proposed data perturbation methods with lasso, elastic net and EMR Feature graph introduced in Chapter 4. We also implemented a combination of these methods.

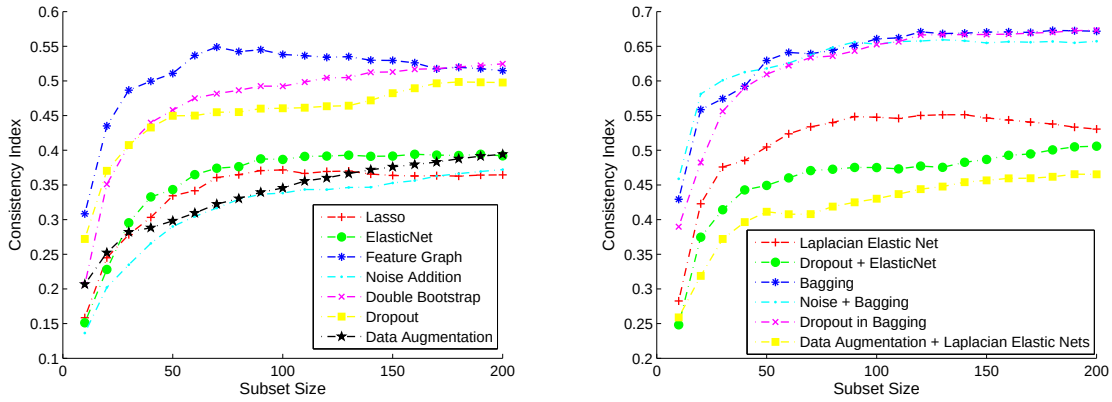
The discrimination of the model with respect to various stabilization techniques are shown in Table B.2. We can infer the following details. The best AUC achieved by the model is 0.65, competitive with existing systems. However, model performance is not improved by combining regularization techniques.

In our experiments, model AUC is least when artificially corrupting features using dropout and data augmentation. The best AUC achieved in this case is 0.63, which is still competitive with existing heart failure readmission models. We believe that this value is a better estimate of the true model AUC, since this resulted from artificially corrupting the features. Hence it is less optimistic.

We looked at various aspects of stability. The stability of the feature subset was numerically validated using Consistency index (Fig B.3).

Standard Techniques	**AUC**
No stabilization (lasso)	0.657 [0.601,0.713]
Elastic Net	0.651 [0.595,0.707]
Graph	0.650 [0.594,0.706]
Multiplicative Gaussian Noise	0.653[0.596, 0.708]
Double Bootstrap	0.648 [0.592,0.704]
Feature dropout	0.633 [0.577,0.690]
Data augmentation	0.623 [0.566,0.679]
Combined Techniques	**AUC**
Elastic Net + Graph	0.654 [0.598,0.710]
Feature dropout + Elastic Net	0.642 [0.586,0.698]
Double Bootstrap +Elastic Net + Graph (Bagging)	0.652 [0.596,0.707]
Gaussian Noise + Bagging	0.653[0.596, 0.708]
Dropout in Bagging	0.640 [0.583,0.696]
Data augmentation + Elastic Net + Graph	0.630 [0.573,0.687]

Table B.2: Model Performance measured as area under the ROC curve (AUC)

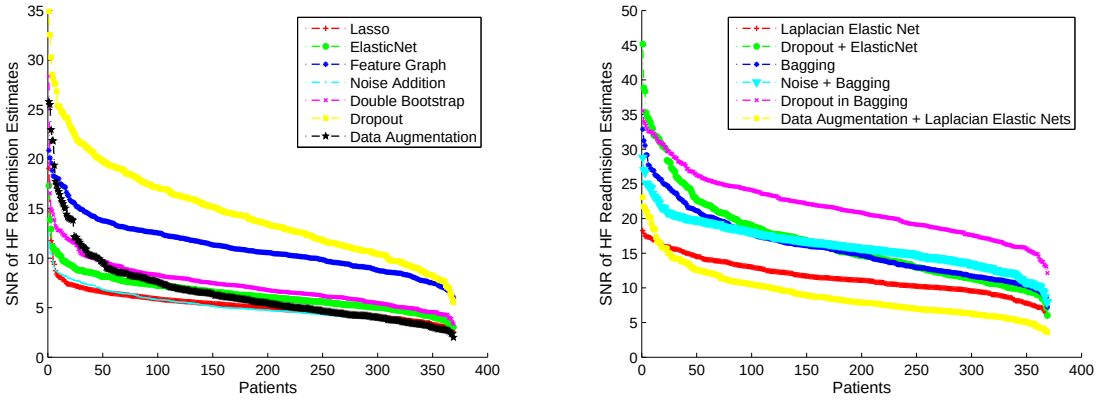


(a) Standard techniques

(b) Combination of techniques

Figure B.3: Feature Stability as measured by Consistency index

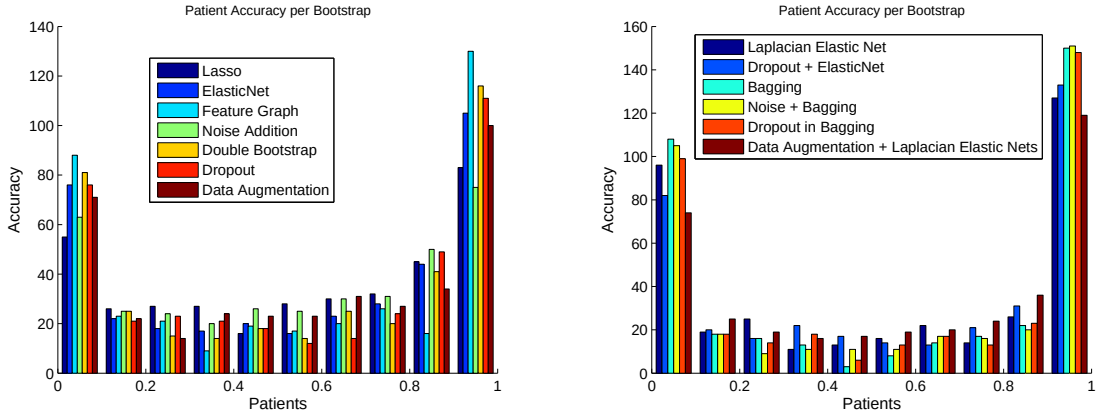
Stability in prediction for each patient is calculated using SNR values (Fig B.4). Probability stability is defined as the variance in probability for each example, and is calculated as $\text{SNR}_{\text{probs}} = \frac{\bar{p}_i}{\sigma_{p_i}}$.



(a) (b)

Figure B.4: Prediction Stability for each patient

Algorithmic stability was measured as the stability of accuracy in high risk patients and is illustrated in Figure B.5.



(a) (b)

Figure B.5: Algorithmic stability measured as Accuracy in high risk patients

We concluded that knowledge based feature graph regularization performed the best in terms of performance and overall stability of features. This regularization scheme was also easier to implement and was less time consuming compared to others.

Bibliography

Fahad H Al-Qahtani and Sven F Crone. Multivariate k-nearest neighbour regression for time series data - a novel algorithm for forecasting uk electricity demand. In *Proceedings of the 2013 International Joint Conference on Neural Networks,*, pages 1–8. IEEE, 2013.

Salem Alelyani, Zheng Zhao, and Huan Liu. A dilemma in assessing stability of feature selection algorithms. In *Proceedings of the 13th International Conference on High Performance Computing and Communications*, pages 701–707. IEEE, 2011.

Francisco Javier Algar, Antonio Alvarez, Angel Salvatierra, Carlos Baamonde, José Luis Aranda, and Francisco Javier López-Pujol. Predicting pulmonary complications after pneumonectomy for lung cancer. *European journal of cardio-thoracic surgery*, 23(2): 201–208, 2003.

Afshin Alijani, George B Hanna, Dorin Ziyaie, Suzanne L Burns, Kenneth L Campbell, Marion ET McMurdo, and Alfred Cuschieri. Instrument for objective assessment of appropriateness of surgical bed occupancy: validation study. *BMJ*, 326(7401):1243–1244, 2003.

Douglas G Altman, Yvonne Vergouwe, Patrick Royston, and Karel GM Moons. Prognosis and prognostic research: validating a prognostic model. *BMJ: British Medical Journal*, 338(7708):1432–1435, 2009.

Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

N Amaral, S Varkey, O Demir, P Sharma, N Varsani, M Kelshikir, K Norrington, HK Turner, MF Barakat, and DO Okonko. The cardiorenal volume index: a simple biochemical algorithm for the differentiation, assessment, and risk stratification of

patients hospitalised for heart failure. *European Heart Journal*, 34(suppl 1):P5060, 2013.

Ruben Amarasingham, Billy J Moore, Ying P Tabak, Mark H Drazner, Christopher A Clark, Song Zhang, W Gary Reed, Timothy S Swanson, Ying Ma, and Ethan A Halm. An automated model to identify heart failure patients at risk for 30-day readmission or death using electronic medical record data. *Medical care*, 48(11):981–988, 2010.

Javier Arroyo and Carlos Maté. Forecasting histogram time series with k-nearest neighbours methods. *International Journal of Forecasting*, 25(1):192–207, 2009.

Wai-Ho Au, Keith CC Chan, Andrew KC Wong, and Yang Wang. Attribute clustering for grouping, selection, and classification of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):83–101, 2005.

Peter C Austin and Jack V Tu. Automated variable selection methods for logistic regression produced unstable models for predicting acute myocardial infarction mortality. *Journal of clinical epidemiology*, 57(11):1138–1146, 2004.

Richard F Averill, John H Muldoon, James C Vertrees, Norbert I Goldfield, Robert L Mullin, Elizabeth C Fineran, Mona Z Zhang, Barbara Steinbeck, and Thelma Grant. The evolution of casemix measurement using diagnosis related groups (DRGs). *Wallingford: 3M Health Information Systems*, 1998.

Wael Awada, Taghi M Khoshgoftaar, David Dittman, Randall Wald, and Amri Napolitano. A review of the stability of feature selection techniques for bioinformatics data. In *Proceedings of the 13th International Conference on Information Reuse and Integration*, pages 356–363. IEEE, 2012.

Adrian Bagust, Michael Place, and John W Posnett. Dynamics of bed use in accommodating emergency admissions: stochastic simulation model. *BMJ*, 319(7203):155–158, 1999.

Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016.

Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.

Sean Barnes, Eric Hamrock, Matthew Toerper, Sauleh Siddiqui, and Scott Levin. Real-time prediction of inpatient length of stay for discharge prioritization. *Journal of the American Medical Informatics Association*, 23:e2–e10, 2016.

Elizabeth Barrett-Connor. Diabetes and heart disease. *Diabetes care*, 26(10):2947–2958, 2003.

Iyad Batal, Hamed Valizadegan, Gregory F. Cooper, and Milos Hauskrecht. A temporal pattern mining approach for classifying electronic health record data. *ACM Trans. Intell. Syst. Technol.*, 4(4):63:1–63:22, 2013.

Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Casey C Bennett. Utilizing RxNorm to support practical computing applications: Capturing medication history in live electronic health records. *Journal of biomedical informatics*, 45(4):634–641, 2012.

Vasiliki Betihavas, Patricia M Davidson, Phillip J Newton, Steven A Frost, Peter S Macdonald, and Simon Stewart. What are the factors in risk prediction models for rehospitalisation for adults with chronic heart failure? *Australian Critical Care*, 25 (1):31–40, 2012.

Viv Bewick, Liz Cheek, and Jonathan Ball. Statistics review 13: receiver operating characteristic curves. *Critical care*, 8(6):508–512, 2004.

W Dean Bidgood, Steven C Horii, Fred W Prior, and Donald E Van Syckle. Understanding and using DICOM, the data interchange standard for biomedical imaging. *Journal of the American Medical Informatics Association*, 4(3):199–212, 1997.

Erhan Bilal, Janusz Dutkowski, Justin Guinney, In Sock Jang, Benjamin A Logsdon, Gaurav Pandey, Benjamin A Sauerwine, Yishai Shimoni, Hans Kristian Moen Vollan, Brigham H Mecham, et al. Improving breast cancer survival analysis through competition-based multidimensional modeling. *PLoS computational biology*, 9(5): e1003047, 2013.

ZW Birnbaum et al. On a use of the mann-whitney statistic. In *Proceedings of the third Berkeley symposium on mathematical statistics and probability*, volume 1, pages 13–17. University of California Press Berkeley, CA, 1956.

Chris M Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

Christopher M. Bishop. *Pattern recognition and machine learning*, volume 1. Springer-Verlag New York, Inc., 2006. ISBN 0387310738.

J Frederic Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013. ISBN 978-0-387-98705-7.

Anne-Laure Boulesteix and Martin Slawski. Stability and aggregation of ranked gene lists. *Briefings in bioinformatics*, 10(5):556–568, 2009.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–526, 2002.

George Edward Pelham Box and Gwilym Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990. ISBN 0816211043.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

Justin Boyle, Marianne Wallis, Melanie Jessup, Julia Crilly, James Lind, Peter Miller, and Gerard Fitzgerald. Regression forecasting of patient admission data. In *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pages 3819–3822. IEEE, 2008.

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

Leo. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984. ISBN 9780412048418.

Leo Breiman et al. Heuristics of instability and stabilization in model selection. *The annals of statistics*, 24(6):2350–2383, 1996.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Peter Bühlmann. Bagging, boosting and ensemble methods. In *Handbook of Computational Statistics*, pages 985–1022. Springer, 2012.

Alex AT Bui and Ricky K Taira. *Medical imaging informatics*. Springer, 2009. ISBN 9781441903846.

Hui Cao, Marianthi Markatou, Genevieve B Melton, Michael F Chiang, and George Hripcsak. Mining a clinical data warehouse to discover disease-finding associations using co-occurrence statistics. In *AMIA Annual Symposium Proceedings*, volume 2005, pages 106–110. American Medical Informatics Association, 2005.

T Chai and RR Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? *Geoscientific Model Development Discussions*, 7:1525–1534, 2014.

Chris Chatfield. *The analysis of time series: an introduction*. CRC press, 2013. ISBN 9781584883173.

Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, 2006.

MD Chin, H Marshall, MD Goldman, et al. Correlates of early hospital readmission or death in patients with congestive heart failure. *The American journal of cardiology*, 79(12):1640–1644, 1997.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM, 2016.

Han-Yu Chuang, Eunjung Lee, Yu-Tsueng Liu, Doheon Lee, and Trey Ideker. Network-based classification of breast cancer metastasis. *Molecular systems biology*, 3(1):140, 2007.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. ISBN 0821803158.

David E Clark and Louise M Ryan. Concurrent prediction of hospital mortality and length of stay from risk factors on admission. *Health services research*, 37(3):631–645, 2002.

William S Cleveland, Eric Grosse, and William M Shyu. Local regression models. *Statistical models in S*, 2:309–376, 1992.

Catherine Combes, Farid Kadri, and Sondès Chaabane. Predicting hospital length of stay using regression models: Application to emergency department. In *Proceedings of 10ème Conférence Francophone de Modélisation, Optimisation et Simulation-MOSIMŠ14*, pages 1–11, 2014.

Michael Connolly, Jane Grimshaw, Mary Dodd, Julie Cawthorne, Tarnya Hulme, Sarah Everitt, Stephanie Tierney, and Christi Deaton. Systems and people under pressure: the discharge process in an acute hospital. *Journal of clinical nursing*, 18(4): 549–558, 2009.

Michael Connolly, Christi Deaton, Mary Dodd, Jane Grimshaw, Tarnya Hulme, Sarah Everitt, and Stephanie Tierney. Discharge preparation: Do healthcare professionals differ in their opinions? *Journal of interprofessional care*, 24(6):633–643, 2010.

Ronald Cornet and Nicolette de Keizer. Forty years of SNOMED: a literature review. *BMC medical informatics and decision making*, 8(Suppl 1):S2, 2008.

AX Costa, SA Ridley, AK Shahani, Paul Robert Harper, V De Senna, and MS Nielsen. Mathematical modelling and simulation for planning critical care capacity. *Anaesthesia*, 58(4):320–327, 2003.

Murray J Cote and Stephen L Tucker. Four methodologies to improve healthcare demand forecasting. *Healthcare Financial Management*, 55(5):54–54, 2001.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

David R Cox. Partial likelihood. *Biometrika*, 62(2):269–276, 1975.

Yupeng Cun and Holger Fröhlich. Network and data integration for biomarker signature discovery via network smoothed t-statistics. *PloS one*, 8(9):e73074, 2013.

Matthew J Daniels, Michael E Kuhl, and Elisabeth Hager. Forecasting hospital bed availability using simulation and neural networks. In *Proceedings of IIE Annual Conference and Exposition*, pages 1–6. Institute of Industrial Engineers-Publisher, 2005.

Gary A Davis and Nancy L Nihan. Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering*, 117(2):178–188, 1991.

Koen W De Bock and Dirk Van den Poel. Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Systems with Applications*, 39(8):6816–6826, 2012.

Luc Devroye and T Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979.

Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000a.

Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000b.

David Dittman, Taghi M Khoshgoftaar, Randall Wald, and Huanjing Wang. Stability analysis of feature ranking techniques on biological datasets. In *Proceedings of 2011 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 252–256. IEEE, 2011.

Carsten F Dormann, Jane Elith, Sven Bacher, Carsten Buchmann, Gudrun Carl, Gabriel Carré, Jaime R García Marquéz, Bernd Gruber, Bruno Lafourcade, Pedro J Leitão, et al. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36(1):27–46, 2013.

Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133 (1):e54–e63, 2014.

Kevin Dunne, Padraig Cunningham, and Francisco Azuaje. Solutions to instability problems with sequential wrapper-based approaches to feature selection. *Journal of Machine Learning Research*, pages 1–22, 2002.

Kenneth Dwyer and Robert Holte. Decision tree instability and active learning. In *Proceedings of European Conference on Machine Learning*, pages 128–139. Springer, 2007.

Arul Earnest, Mark I Chen, Donald Ng, and Leo Yee Sin. 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, 2005.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994. ISBN 9781489945419.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Liat Ein-Dor, Or Zuk, and Eytan Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences*, 103(15):5923–5928, 2006.

E El-Darzi, C Vasilakis, T Chaussalet, and PH Millard. A simulation modelling approach to evaluating length of stay, occupancy, emptiness and bed blocking in a hospital geriatric department. *Health Care Management Science*, 1(2):143–149, 1998.

Richard H Epstein, Paul St Jacques, Michael Stockin, Brian Rothman, Jesse M Ehrenfeld, and Joshua C Denny. Automated identification of drug and food allergies entered using non-standard terminology. *Journal of the American Medical Informatics Association*, 20(5):962–968, 2013.

A Famili and Peter Turney. Intelligently helping the human planner in industrial process planning. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 5(02):109–124, 1991.

Alvan R Feinstein. The pre-therapeutic classification of co-morbidity in chronic disease. *Journal of chronic diseases*, 23(7):455–468, 1970.

G Michael Felker, Jeffrey D Leimberger, Robert M Califf, Michael S Cuffe, Barry M Massie, Kirkwood F Adams Jr, Mihai Gheorghiade, and Christopher M O'Connor. Risk stratification after hospitalization for decompensated heart failure. *Journal of cardiac failure*, 10(6):460–466, 2004.

Caroline S Fox, Kunihiro Matsushita, Mark Woodward, Henk JG Bilo, John Chalmers, Hiddo J Lambers Heerspink, Brian J Lee, Robert M Perkins, Peter Rossing, Toshimi Sairenchi, et al. Associations of kidney disease measures with mortality and end-stage renal disease in individuals with and without diabetes: a meta-analysis. *The Lancet*, 380(9854):1662–1673, 2012.

Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–374, April 2000.

Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, pages 916–954, 2008.

Kleber A Garcia and Philip K Chan. Estimating hospital admissions with a randomized regression approach. In *Proceedings of 11th International Conference on Machine Learning and Applications*, volume 1, pages 179–184. IEEE, 2012.

Shivapratap Gopakumar, TuDinh Nguyen, Truyen Tran, Dinh Phung, and Svetha Venkatesh. Stabilizing sparse cox model using statistic and semantic structures in electronic medical records. In *Advances in Knowledge Discovery and Data Mining*, volume 9078, pages 331–343. Springer International Publishing, 2015a.

Shivapratap Gopakumar, Truyen Tran, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. Stabilizing high-dimensional prediction models using feature graphs. *IEEE Journal of Biomedical and Health Informatics*, 19(3):1044–1052, 2015b.

Florin Gorunescu, Sally I McClean, and Peter H Millard. Using a queueing model to help plan bed allocation in a department of geriatric medicine. *Health Care Management Science*, 5(4):307–312, 2002.

Edward W Gregg, Naveed Sattar, and Mohammed K Ali. The changing face of diabetes complications. *The Lancet Diabetes & Endocrinology*, 4(6):537–547, 2016.

Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

Gokhan Gulgezen, Zehra Cataltepe, and Lei Yu. Stable and accurate feature selection. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 455–468. Springer, 2009.

Sunil Gupta, Truyen Tran, Wei Luo, Dinh Phung, Richard Lee Kennedy, Adam Broad, David Campbell, David Kipp, Madhu Singh, Mustafa Khasraw, et al. Machine-learning prediction of cancer survival: a retrospective study using electronic administrative records and a cancer registry. *BMJ open*, 4(3):e004007, 2014.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

Isabelle Guyon and André Elisseeff. An introduction to feature extraction. In *Feature Extraction: Foundations and Applications*, pages 1–25. Springer Berlin Heidelberg, 2006.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3): 389–422, 2002.

AB Haidich. Meta-analysis in medical research. *Hippokratia*, 14(1):29–37, 2011.

Yoni Halpern, Youngduck Choi, Steven Horng, and David Sontag. Using anchors to estimate clinical state without labeled data. In *AMIA Annual Symposium Proceedings*, volume 2014, pages 606–615. American Medical Informatics Association, 2014.

Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.

Yue Han and Lei Yu. A variance reduction framework for stable feature selection. *Statistical Analysis and Data Mining*, 5(5):428–445, 2012.

Rave Harpaz, Santiago Vilar, William DuMouchel, Hojjat Salmasian, Krystl Haerian, Nigam H Shah, Herbert S Chase, and Carol Friedman. Combing signals from spontaneous reports and electronic health records for detection of adverse drug reactions. *Journal of the American Medical Informatics Association*, 20(3):413–419, 2013.

Paul Robert Harper and AK Shahani. Modelling for the planning and management of bed capacities in hospitals. *Journal of the Operational Research Society*, 53(1):11–18, 2002.

FE Harrell, Kerry L Lee, and Daniel B Mark. Tutorial in biostatistics multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15:361–387, 1996.

Frank Harrell. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer, 2015. ISBN 9781475734621.

Trevor Hastie, Robert Tibshirani, David Botstein, and Patrick Brown. Supervised harvesting of expression trees. *Genome Biology*, 2(1):1, 2001a.

Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001b. ISBN 9780387216065.

Anne-Claire Haury, Pierre Gestraud, and Jean-Philippe Vert. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one*, 6(12):e28210, 2011.

Kristiina Häyrinen, Kaija Saranto, and Pirkko Nykänen. Definition, structure, content, use and impacts of electronic health records: a review of the research literature. *International journal of medical informatics*, 77(5):291–304, 2008.

Danning He, Simon C Mathews, Anthony N Kalloo, and Susan Hutfless. Mining high-dimensional administrative claims data to predict early hospital readmissions. *Journal of the American Medical Informatics Association*, pages amiajnl–2013, 2013.

Zengyou He and Weichuan Yu. Stable feature selection for biomarker discovery. *Computational biology and chemistry*, 34(4):215–225, 2010.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Antony B Holmes, Alexander Hawson, Feng Liu, Carol Friedman, Hossein Khiabanian, and Raul Rabadan. Discovering disease associations by integrating electronic clinical data and medical literature. *PloS one*, 6(6):e21132, 2011.

Nathan R Hoot, Larry J LeBlanc, Ian Jones, Scott R Levin, Chuan Zhou, Cynthia S Gadd, and Dominik Aronsky. Forecasting emergency department crowding: a discrete event simulation. *Annals of emergency medicine*, 52(2):116–125, 2008.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013. ISBN 9780470582473.

Shuai Huang, Jing Li, Liang Sun, Jun Liu, Teresa Wu, Kewei Chen, Adam Fleisher, Eric Reiman, and Jieping Ye. Learning brain connectivity of alzheimer's disease from neuroimaging data. In *Advances in Neural Information Processing Systems*, pages 808–816, 2009.

Xiaoxu Huo, Leili Gao, Lixin Guo, Wen Xu, Wenbo Wang, Xinyue Zhi, Ling Li, Yanfeng Ren, Xiuying Qi, Zhong Sun, et al. Risk of non-fatal cardiovascular diseases in early-onset versus late-onset type 2 diabetes in china: a cross-sectional study. *The Lancet Diabetes & Endocrinology*, 4(2):115–124, 2016.

Rob J Hyndman. Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4(4):43–46, 2006.

Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008.

Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning: with applications in R*, volume 103 of *Springer Texts in Statistics*. Springer-Verlag New York, 1 edition, 2013. ISBN 978-1-4614-7138-7.

Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12(Oct):2777–2824, 2011.

Anders Boeck Jensen, Pope L Moseley, Tudor I Oprea, Sabrina Gade Ellesøe, Robert Eriksson, Henriette Schmock, Peter Bjødstrup Jensen, Lars Juhl Jensen, and Søren Brunak. Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. *Nature communications*, 5(4022), 2014.

Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13 (6):395–405, 2012.

Ulf Johansson, Cecilia Sönströd, Ulf Norinder, and Henrik Boström. Trade-off between accuracy and interpretability for predictive in silico modeling. *Future medicinal chemistry*, 3(6):647–663, 2011.

Richard J Johnson, Mark S Segal, Yuri Sautin, Takahiko Nakagawa, Daniel I Feig, Duk-Hee Kang, Michael S Gersch, Steven Benner, and Laura G Sánchez-Lozada. Potential role of sugar (fructose) in the epidemic of hypertension, obesity and the metabolic syndrome, diabetes, kidney disease, and cardiovascular disease. *The American journal of clinical nutrition*, 86(4):899–906, 2007.

Ian Jolliffe. *Principal component analysis*. Springer Series in Statistics. Springer-Verlag New York, 2002. ISBN 9780387224404.

Simon Andrew Jones, Mark Patrick Joy, and Jon Pearson. Forecasting demand of emergency care. *Health care management science*, 5(4):297–305, 2002.

Spencer S Jones, Alun Thomas, R Scott Evans, Shari J Welch, Peter J Haug, and Gregory L Snow. Forecasting daily patient volumes in the emergency department. *Academic Emergency Medicine*, 15(2):159–170, 2008.

Giuseppe Jurman, Stefano Merler, Annalisa Barla, Silvano Paoli, Antonio Galea, and Cesare Furlanello. Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics*, 24(2):258–264, 2008.

Farid Kadri, Fouzi Harrou, Sondès Chaabane, and Christian Tahon. Time series modelling and forecasting of emergency department overcrowding. *Journal of medical systems*, 38(9):1–20, 2014.

A Kalache and A Gatti. Active ageing: a policy framework. *Advances in gerontology*, 11:7–18, 2002.

Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and information systems*, 12(1):95–116, 2007.

Iman Kamkar, Sunil Kumar Gupta, Dinh Phung, and Svetha Venkatesh. Exploiting feature relationships towards stable feature selection. In *Proceedings of IEEE International Conference on Data Science and Advanced Analytics*, pages 1–10. IEEE, 2015.

Michael J Kane, Natalie Price, Matthew Scotch, and Peter Rabinowitz. Comparison of ARIMA and random forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC bioinformatics*, 15(1):276, 2014.

Ravi Kannan, Santosh Vempala, and David P Woodruff. Principal component analysis and higher correlations for distributed data. In *Proceedings of 27th Annual Conference on Computational Learning Theory (COLT)*, pages 1040–1057, 2014.

William B Kannel, Marthana Hjortland, and William P Castelli. Role of diabetes in congestive heart failure: the Framingham study. *The American Journal of Cardiology*, 34(1):29–34, 1974.

Devan Kansagara, Honora Englander, Amanda Salanitro, David Kagen, Cecelia Theobald, Michele Freeman, and Sunil Kripalani. Risk prediction models for hospital readmission. *JAMA*, 306(15):1688–1698, 2011.

Robert E Kass, Brian S Caffo, Marie Davidian, Xiao-Li Meng, Bin Yu, and Nancy Reid. Ten simple rules for effective statistical practice. *PLOS Comput Biol*, 12(6):e1004961, 2016.

Taghi M Khoshgoftaar, Alireza Fazelpour, Huanjing Wang, and Randall Wald. A survey of stability analysis of feature subset selection techniques. In *Proceedings of 14th International Conference on Information Reuse and Integration*, pages 424–431. IEEE, 2013.

Seon-Young Kim. Effects of sample size on robustness and prediction accuracy of a prognostic gene signature. *BMC bioinformatics*, 10(1):1, 2009.

Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning*, pages 543–550, 2010.

Jean Klastersky, Marianne Paesmans, Aspasia Georgala, Frédérique Muanza, Barbara Plehiers, Laurent Dubreucq, Yassine Lalami, Michel Aoun, and Martine Barette. Outpatient oral antibiotics for febrile neutropenic cancer patients using a score predictive for complications. *Journal of Clinical Oncology*, 24(25):4129–4134, 2006.

David G Kleinbaum and Mitchel Klein. *Survival analysis: a self-learning text*. Statistics for Biology and Health. Springer-Verlag, New York, 3rd edition, 2006. ISBN 9781441966452.

Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 14, pages 1137–1145. Morgan Kaufmann, 1995.

Inke R König, JD Malley, C Weimar, H-C Diener, and A Ziegler. Practical experiences on the necessity of external validation. *Statistics in medicine*, 26(30):5499–5511, 2007.

Igor Kononenko. Estimating attributes: analysis and extensions of RELIEF. In *Machine Learning: ECML-94: European Conference on Machine Learning Proceedings*, pages 171–182. Springer, 1994.

Urs Köster, Jascha Sohl-Dickstein, Charles M Gray, and Bruno A Olshausen. Modeling higher-order correlations within cortical microcolumns. *PLoS computational biology*, 10(7):e1003684, 2014.

Karen Kostick. SNOMED CT integral part of quality EHR documentation. *Journal of American Health Information Management Association*, 83(10):72–75, 2012.

Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, and Dimitrios I Fotiadis. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, 13:8–17, 2015.

Pavel Křížek, Josef Kittler, and Václav Hlaváč. Improving stability of feature selection methods. In *Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns*, 4673, pages 929–936. Springer-Verlag Berlin Heidelberg, 2007.

Harlan M Krumholz, Eugene M Parent, Nora Tu, Viola Vaccarino, Yun Wang, Martha J Radford, and John Hennen. Readmission after hospitalization for congestive heart failure among medicare beneficiaries. *Archives of Internal Medicine*, 157 (1):99, 1997.

Harlan M Krumholz, Ya-Ting Chen, Yun Wang, Viola Vaccarino, Martha J Radford, and Ralph I Horwitz. Predictors of readmission among elderly survivors of admission with heart failure. *American Heart Journal*, 139(1):72–77, 2000.

Eyal Krupka and Naftali Tishby. Incorporating prior knowledge on features into learning. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS-07)*, pages 227–234. MIT Press, 2007.

Elena Kulinskaya, Diana Kornbrot, and Haiyan Gao. Length of stay as a performance indicator: robust statistical methodology. *IMA Journal of Management Mathematics*, 16(4):369–381, 2005.

Ludmila I Kuncheva. A stability index for feature selection. In *Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, pages 421–427. ACTA Press, 2007.

Thomas A Lasko, Joshua C Denny, and Mia A Levy. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS one*, 8(6):e66341, 2013.

Malcolm R Law, Nicholas J Wald, and AR Rudnicka. Quantifying effect of statins on low density lipoprotein cholesterol, ischaemic heart disease, and stroke: systematic review and meta-analysis. *BMJ*, 326(7404):1423, 2003.

Elisa T Lee and John Wang. *Statistical methods for survival data analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, 4th edition, 2013. ISBN 9781118095027.

Eunjung Lee, Han-Yu Chuang, Jong-Won Kim, Trey Ideker, and Doheon Lee. Inferring pathway activity toward precise disease classification. *PLoS comput biol*, 4(11): e1000217, 2008.

Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient l˜ 1 regularized logistic regression. In *AAAI*, volume 6, pages 401–408, 2006.

Scott R Levin, Eric T Harley, James C Fackler, Christoph U Lehmann, Jason W Custer, Daniel France, and Scott L Zeger. Real-time forecasting of pediatric intensive care unit length of stay using computerized provider orders. *Critical care medicine*, 40 (11):3058–3064, 2012.

Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.

Ruey-Hsia Li and Geneva G Belford. Instability of decision tree classification algorithms. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 570–575. ACM, 2002.

Katherine P Liao, Tianxi Cai, Vivian Gainer, Sergey Goryachev, Qing Zeng-treitler, Soumya Raychaudhuri, Peter Szolovits, Susanne Churchill, Shawn Murphy, Isaac Kohane, et al. Electronic medical records for discovery research in rheumatoid arthritis. *Arthritis Care & Research*, 62(8):1120–1127, 2010.

Rung-Chuan Lin, Kalyan S Pasupathy, and Mustafa Y Sir. Estimating admissions and discharges for planning purposes–case of an academic health system. *Advances in Business and Management Forecasting*, 8:115, 2011.

Wei Lin and Jinchi Lv. High-dimensional sparse additive hazards regression. *Journal of the American Statistical Association*, 108(501):247–264, 2013.

Winston T Lin. Modeling and forecasting hospital patient movements: Univariate and multiple time series approaches. *International Journal of Forecasting*, 5(2):195–208, 1989.

Patrice Lindsay, Michael Schull, Susan Bronskill, and Geoffrey Anderson. The development of indicators to measure the quality of clinical care in emergency departments following a modified-delphi approach. *Academic Emergency Medicine*, 9(11):1131–1139, 2002.

Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize f1 measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–239. Springer, 2014.

Steven J Littig and Mark W Isken. Short term hospital occupancy prediction. *Health care management science*, 10(1):47–66, 2007.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Steven Loscalzo, Lei Yu, and Chris Ding. Consensus group stable feature selection. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 567–576. ACM, 2009.

Thomas F Lüscher. Heart failure and comorbidities: renal failure, diabetes, atrial fibrillation, and inflammation. *European heart journal*, 36(23):1415–1417, 2015.

Shuangge Ma and Jian Huang. Penalized feature selection and classification in bioinformatics. *Briefings in bioinformatics*, 9(5):392–403, 2008.

Shuangge Ma, Xiao Song, and Jian Huang. Supervised group lasso with applications to microarray data analysis. *BMC bioinformatics*, 8(1):1–17, 2007.

Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Q Weinberger. Learning with marginalized corrupted features. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 410–418, 2013.

Mark Mackay. Practical experience with bed occupancy management and planning systems: an australian view. *Health Care Management Science*, 4(1):47–56, 2001.

Mark Mackay and Michael Lee. Choice of models for the analysis and forecasting of hospital beds. *Health Care Management Science*, 8(3):221–230, 2005.

Adele Marshall, Christos Vasilakis, and Elia El-Darzi. Length of stay-based patient flow models: recent developments and future directions. *Health Care Management Science*, 8(3):213–220, 2005.

Sally McClean and Peter H Millard. A decision support system for bed-occupancy management and planning hospitals. *Mathematical Medicine and Biology*, 12(3-4): 249–257, 1995.

Sally I McClean and Peter H Millard. A three compartment model of the patient flows in a geriatric department: a decision support approach. *Health care management science*, 1(2):159–163, 1998.

Stéphane M Meystre, Vikrant G Deshmukh, and Joyce Mitchell. A clinical use case to evaluate the i2b2 Hive: predicting asthma exacerbations. In *AMIA Annual Symposium Proceedings*, volume 2009, pages 442–446. American Medical Informatics Association, 2009.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

TM Mills. A mathematician goes to hospital. *Australian Mathematical Society Gazette*, 31(5):320–327, 2004.

Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6:26094, 2016.

Karel GM Moons, Douglas G Altman, Yvonne Vergouwe, and Patrick Royston. Prognosis and prognostic research: application and impact of prognostic models in clinical practice. *BMJ: British Medical Journal*, 338(7709):1487–1490, 2009.

JA Moukarzel, P Klin, C Zambrano, MP Duczynski, JP Ochoa, A Bilbao, and F Klein. Clinical relevance of dynamic changes in renal function in patients admitted for acute heart failure. *European Heart Journal*, 34(suppl 1):P2733, 2013.

Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1-3): 161–193, 2006.

Fliss EM Murtagh, Emma Murphy, and Neil S Sheerin. Illness trajectories: an important concept in the management of kidney failure. *Nephrology Dialysis Transplantation*, 23(12):3746–3748, 2008.

Stefano Muzzarelli, Gregor Leibundgut, Micha T Maeder, Hans Rickli, Rolf Handschin, Marc Gutmann, Urs Jeker, Peter Buser, Matthias Pfisterer, Hans-Peter Brunner-La Rocca, et al. Predictors of early readmission or death in elderly patients with heart failure. *American heart journal*, 160(2):308–314, 2010.

Senthil K Nachimuthu, Anthony Wong, and Peter J Haug. Modeling glucose homeostasis and insulin dosing in an intensive care unit using dynamic bayesian networks. In *AMIA Annual Symposium Proceedings*, volume 2010, pages 532–536. American Medical Informatics Association, 2010.

Stuart J Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs: RxNorm at 6 years. *Journal of the American Medical Informatics Association*, 18(4):441–448, 2011.

Andrew Y Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the21st International Conference on Machine learning*, page 78. ACM, 2004.

Kenney Ng, Amol Ghoting, Steven R Steinhubl, Walter F Stewart, Bradley Malin, and Jimeng Sun. Paramo: A parallel predictive modeling platform for healthcare analytic research using electronic health records. *Journal of biomedical informatics*, 48:160–170, 2014.

Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: A convolutional net for medical records. *arXiv preprint arXiv:1607.07519*, 2016.

Tu Dinh Nguyen, Truyen Tran, Dinh Phung, and Svetha Venkatesh. Latent patient profile modelling and applications with mixed-variate Restricted Boltzmann Machine. In *Proceedings of 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, April 2013.

Ziad Obermeyer and Ezekiel J Emanuel. Predicting the future - big data, machine learning, and clinical medicine. *The New England Journal of Medicine*, 375(13):1216, 2016.

OECD. A disease-based comparison of health systems. May 2003. doi: http://dx.doi.org/10.1787/9789264100053-en.

Murat Y Ozkalkanli, Dila Tuna Ozkalkanli, Kaan Katircioglu, and Serdar Savaci. Comparison of tools for nutrition assessment and screening for predicting the development of complications in orthopedic surgery. *Nutrition in Clinical Practice*, 24(2): 274–280, 2009.

Fred C Pampel. *Logistic regression: A primer*, volume 132 of *Quantitative Applications in the Social Sciences*. Sage, 2000. ISBN 9780761920106.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

Kevin M Pantalone, Todd M Hobbs, Brian J Wells, Sheldon X Kong, Michael W Kattan, Jonathan Bouchard, Changhong Yu, Brian Sakurada, Alex Milinovich, Wayne Weng, et al. Clinical characteristics, complications, comorbidities and treatment patterns among patients with type 2 diabetes mellitus in a large integrated health system. *BMJ open diabetes research & care*, 3(1):e000093, 2015.

Mee Young Park, Trevor Hastie, and Robert Tibshirani. Averaged gene expressions for regression. *Biostatistics*, 8(2):212–227, 2007.

Jordan S Peck, James C Benneyan, Deborah J Nightingale, and Stephan A Gaehde. Predicting emergency department inpatient admissions to improve same-day patient flow. *Academic Emergency Medicine*, 19(9):E1045–E1054, 2012.

Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1): 3–14, 2002.

Diego Peteiro-Barral, Veronica Bolon-Canedo, Amparo Alonso-Betanzos, Bertha Guijarro-Berdinas, and Noelia Sanchez-Marono. Scalability analysis of filter-based methods for feature selection. *Advances in Smart Systems Research*, 2(1):21–26, 2012.

Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. DeepCare: a deep dynamic memory model for predictive medicine. In *Proceedings of 20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 30–41. Springer, 2016.

Edward F Philbin and Thomas G DiSalvo. Prediction of hospital readmission for heart failure: development of a simple risk score based on administrative data. *Journal of the American College of Cardiology*, 33(6):1560–1566, 1999.

T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428(6981):419–422, 2004.

Karl Popper. *The logic of scientific discovery*. Philosophy of science. Hutchinson & Co, 1959. ISBN 0415278449.

William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes in C*, volume 2 of *The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1996. ISBN 0521431085.

Judith J Prochaska, Wayne F Velicer, Claudio R Nigg, and James O Prochaska. Methods of quantifying change in multiple risk factor interventions. *Preventive medicine*, 46(3):260–265, 2008.

Anthony M Propst, Rebecca F Liberman, Bernard L Harlow, and Elizabeth S Ginsburg. Complications of hysteroscopic surgery: predicting patients at risk. *Obstetrics & Gynecology*, 96(4):517–520, 2000.

Xian Qian, Xiaoqian Jiang, Qi Zhang, Xuanjing Huang, and Lide Wu. Sparse higher order conditional random fields for improved sequence labeling. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 849–856. ACM, 2009.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 759–766. ACM, 2007.

Franck Rapaport, Andrei Zinovyev, Marie Dutreix, Emmanuel Barillot, and Jean-Philippe Vert. Classification of microarray data using gene networks. *BMC bioinformatics*, 8(1):35, 2007.

Raimundo Real and Juan M Vargas. The probabilistic basis of jaccard's index of similarity. *Systematic biology*, 45(3):380–385, 1996.

Chandan K Reddy and Charu C Aggarwal. *Healthcare data analytics*, volume 36 of *Chapman & Hall/CRC Data Mining and Knowledge Discovery Series*. CRC Press, 2015. ISBN 9781482232110.

Chandan K. Reddy and Rajiur Rahman. *Electronic Health Records: A Survey in Healthcare Data Analytics*. Healthcare Data Analytics. CRC Press, 2014.

Lior Rokach and Oded Maimon. *Data mining with decision trees: theory and applications*, volume 69 of *Machine Perception and Artificial Intelligence*. World scientific, 2014. ISBN 9789812771728.

Francisco S Roque, Peter B Jensen, Henriette Schmock, Marlene Dalgaard, Massimo Andreatta, Thomas Hansen, Karen Søeby, Søren Bredkjær, Anders Juul, Thomas Werge, et al. Using electronic patient records to discover disease correlations and stratify patient cohorts. *PLoS Comput Biol*, 7(8):e1002141, 2011.

Cédric Rose, Cherif Smaili, and Francois Charpillet. A dynamic Bayesian network for handling uncertainty in a decision support system adapted to the monitoring of patients treated by hemodialysis. In *17th IEEE International Conference on Tools with Artificial Intelligence*, pages 594–598. IEEE, 2005.

Joseph S Ross, Gregory K Mulvey, Brett Stauffer, Vishnu Patlolla, Susannah M Bernheim, Patricia S Keenan, and Harlan M Krumholz. Statistical models and patient predictors of readmission for heart failure: A systematic review. *Archives of Internal Medicine*, 168(13):1371–1386, 2008.

Lucia Sacchi, Cristiana Larizza, Carlo Combi, and Riccardo Bellazzi. Data mining with temporal abstractions: learning rules from time series. *Data Mining and Knowledge Discovery*, 15(2):217–247, 2007.

Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Procedings of Joint European Conference on*

Machine Learning and Knowledge Discovery in Databases, pages 313–325. Springer, 2008.

Charles Safran, Meryl Bloomrosen, W Edward Hammond, Steven Labkoff, Suzanne Markel-Fox, Paul C Tang, and Don E Detmer. Toward a national framework for the secondary use of health data: an american medical informatics association white paper. *Journal of the American Medical Informatics Association*, 14(1):1–9, 2007.

Budhaditya Saha, Sunil Gupta, and Svetha Venkatesh. Improved risk predictions via sparse imputation of patient conditions in electronic medical records. In *Proceedings of IEEE International Conference on Data Science and Advanced Analytics*, pages 1–10. IEEE, 2015.

Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and María Tombilla-Sanromán. Filter methods for feature selection–a comparative study. In *Proceedings of 16th International Conference on Intelligent Data Engineering and Automated Learning*, pages 178–187. Springer, 2007.

Ted Sandler, John Blitzer, Partha P Talukdar, and Lyle H Ungar. Regularized learning with networks of features. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pages 1401–1408, 2008.

Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, 2009.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

Rainer Schmidt and Lothar Gierl. A prognostic model for temporal courses that combines temporal abstraction and case-based reasoning. *International journal of medical informatics*, 74(2):307–315, 2005.

Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.

John R Schrom, Pedro J Caraballo, M Regina Castro, and György J Simon. Quantifying the effect of statin use in pre-diabetic phenotypes discovered through association rule mining. In *AMIA Annual Symposium Proceedings*, volume 2013, page 1249. American Medical Informatics Association, 2013.

Lisa M Schweigler, Jeffrey S Desmond, Melissa L McCarthy, Kyle J Bukowski, Edward L Ionides, and John G Younger. Forecasting models of emergency department crowding. *Academic Emergency Medicine*, 16(4):301–308, 2009.

Yuval Shahar and Mark A Musen. A temporal-abstraction system for patient monitoring. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 121. American Medical Informatics Association, 1992.

Yuval Shahar and Mark A Musen. Knowledge-based temporal abstraction in clinical domains. *Artificial intelligence in medicine*, 8(3):267–298, 1996.

Maxim Vladimirovich Shcherbakov, Adriaan Brebels, Nataliya Lvovna Shcherbakova, Anton Pavlovich Tyukov, Timur Alexandrovich Janovsky, and Valeriy Anatolevich Kamaev. A survey of forecast error measures. *World Applied Sciences Journal*, 24: 171–176, 2013.

A Mi Shin, In Hee Lee, Gyeong Ho Lee, Hee Joon Park, Hyung Seop Park, Kyung Il Yoon, Jung Jeung Lee, and Yoon Nyun Kim. Diagnostic analysis of patients with essential hypertension using association rule mining. *Healthcare informatics research*, 16(2):77–81, 2010.

Gyorgy J Simon, John Schrom, M Regina Castro, Peter W Li, and Pedro J Caraballo. Survival association rule mining towards type 2 diabetes risk assessment. In *AMIA Annual Symposium Proceedings*, volume 2013, page 1293. American Medical Informatics Association, 2013.

David Sinreich and Yariv Marmor. Emergency department operations: the basis for developing a simulation tool. *IIE Transactions*, 37(3):233–245, 2005.

Marina Skurichina and Robert PW Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135, 2002.

Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

Alex J Smola, Bernhard Schölkopf, and Klaus-Robert Müller. The connection between regularization operators and support vector kernels. *Neural networks*, 11(4):637–649, 1998.

Nelson F SooHoo, Jay R Lieberman, Clifford Y Ko, and David S Zingmond. Factors predicting complication rates following total knee replacement. *J Bone Joint Surg Am*, 88(3):480–485, 2006.

Robert GD Steel and H James. Principles and procedures of statistics: with special reference to the biological sciences. Technical report, New York, US: McGraw-Hill, 1960.

Ewout W Steyerberg. *Clinical prediction models: A Practical Approach to Development, Validation, and Updating*. Statistics for Biology and Health. Springer-Verlag New York, 1st edition, 2009. ISBN 9780387772448.

Ewout W Steyerberg, Frank E Harrell Jr, Gerard JJM Borsboom, MJC Eijkemans, Yvonne Vergouwe, and J Dik F Habbema. Internal validation of predictive models: efficiency of some procedures for logistic regression analysis. *Journal of clinical epidemiology*, 54(8):774–781, 2001.

Ewout W Steyerberg, Andrew J Vickers, Nancy R Cook, Thomas Gerds, Mithat Gonen, Nancy Obuchowski, Michael J Pencina, and Michael W Kattan. Assessing the performance of prediction models: a framework for some traditional and novel measures. *Epidemiology (Cambridge, Mass.)*, 21(1):128, 2010.

Hokeun Sun, Wei Lin, Rui Feng, and Hongzhe Li. Network-regularized high-dimensional cox regression for analysis of genomic data. *Statistica Sinica*, 24(3): 1433–1459, 2014.

Feng Tai and Wei Pan. Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms. *Bioinformatics*, 23(14):1775–1782, 2007.

John G Taylor and Stephen Coombes. Learning higher order correlations. *Neural Networks*, 6(3):423–427, 1993.

Joan M Teno, Sherry Weitzen, Mary L Fennell, and Vincent Mor. Dying trajectory in the last year of life: does cancer trajectory fit other diseases? *Journal of palliative medicine*, 4(4):457–464, 2001.

Krish Thiru, Alan Hassey, and Frank Sullivan. Systematic review of scope and quality of electronic patient record data in primary care. *BMJ*, 326(7398):1070, 2003.

Paul Thottakkara, Tezcan Ozrazgat-Baslanti, Bradley B Hupf, Parisa Rashidi, Panos Pardalos, Petar Momcilovic, and Azra Bihorac. Application of machine learning techniques to high-dimensional clinical data to forecast postoperative complications. *PloS one*, 11(5):e0155705, 2016.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Robert Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

Truyen Tran, Dinh Phung, Wei Luo, Richard Harvey, Michael Berk, and Svetha Venkatesh. An integrated framework for suicide risk prediction. In *19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1410–1418. ACM, 2013.

Truyen Tran, Wei Luo, Dinh Phung, Sunil Gupta, Santu Rana, Richard L Kennedy, Ann Larkins, and Svetha Venkatesh. A framework for feature extraction from hospital medical data with applications in risk prediction. *BMC bioinformatics*, 15(1): 6596, 2014.

Truyen Tran, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines (eNRBM). *Journal of biomedical informatics*, 54:96–105, 2015a.

Truyen Tran, Dinh Phung, Wei Luo, and Svetha Venkatesh. Stabilized sparse ordinal regression for medical risk stratification. *Knowledge and Information Systems*, 43(3): 555–582, 2015b. ISSN 0219-1377.

AC Tsakoumis, SS Vladov, and VM Mladenov. Daily load forecasting based on previous day load. In *Proceedings of 6th Seminar on Neural Network Applications in Electrical Engineering*, pages 83–86. IEEE, 2002.

Peter Turney. Technical note: Bias and the quantification of stability. *Journal of Machine Learning*, 20(1-2):23–33, 1995.

Frans Van de Werf, Jeroen Bax, Amadeo Betriu, Carina Blomstrom-Lundqvist, Filippo Crea, Volkmar Falk, Gerasimos Filippatos, Keith Fox, Kurt Huber, Adnan Kastrati, et al. Management of acute myocardial infarction in patients presenting with persistent st-segment elevation. *European heart journal*, 29(23):2909–2945, 2008.

Vincent M van Deursen, Renato Urso, Cecile Laroche, Kevin Damman, Ulf Dahlström, Luigi Tavazzi, Aldo P Maggioni, and Adriaan A Voors. Co-morbidities in patients with heart failure: an analysis of the european heart failure pilot survey. *European journal of heart failure*, 16(1):103–111, 2014.

Jason Van Hulse, Taghi M Khoshgoftaar, Amri Napolitano, and Randall Wald. Feature selection with high-dimensional imbalanced data. In *2009 IEEE International Conference on Data Mining Workshops*, pages 507–514. IEEE, 2009.

Carl van Walraven and Chaim M Bell. Risk of death or readmission among people discharged from hospital on fridays. *Canadian Medical Association Journal*, 166(13): 1672–1673, 2002.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer-Verlag New York, 2nd edition, 2013. ISBN 9781475732641.

Pratibha Vellanki, Thi Duong, Svetha Venkatesh, and Dinh Phung. Nonparametric discovery of learning patterns and autism subgroups from therapeutic data. In *Proceedings of 22nd International Conference on Pattern Recognition*, pages 1828–1833. IEEE, 2014.

Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. *A Primer on Kernel Methods*, pages 35–70. MIT Press, Cambridge, MA, USA, 2004.

Santiago Vilar, Rave Harpaz, Lourdes Santana, Eugenio Uriarte, and Carol Friedman. Enhancing adverse drug event detection in electronic health records using molecular structure similarity: application to pancreatitis. *PloS one*, 7(7):e41471, 2012.

Bhanukiran Vinzamuri and Chandan K Reddy. Cox regression with correlation based regularization for electronic health records. In *Proceedings of the 13th International Conference on Data Mining*, pages 757–766. IEEE, 2013.

Bhanukiran Vinzamuri, Yan Li, and Chandan K. Reddy. Active learning based survival regression for censored data. In *Proceedings of the 23rd ACM International Confer-*

ence on Conference on Information and Knowledge Management, pages 241–250. ACM, 2014. ISBN 978-1-4503-2598-1.

Stefan Wager, Sida Wang, and Percy Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 351–359, 2013.

Martin J Wainwright, John D Lafferty, and Pradeep K Ravikumar. High-dimensional graphical model selection using l1-regularized logistic regression. In *Advances in neural information processing systems*, pages 1465–1472, 2007.

Matt P Wand and M Chris Jones. *Kernel smoothing*, volume 60 of *Monographs on Statistics and Applied Probability*. CRC Press, 1st edition, 1994. ISBN 9780412552700.

F. Wang, N. Lee, J. Hu, J. Sun, and S. Ebadollahi. Towards heterogeneous temporal clinical event pattern discovery: a convolutional approach. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 453–461. ACM, 2012a.

Huanjing Wang, Taghi M Khoshgoftaar, and Randall Wald. Measuring robustness of feature selection techniques on software engineering datasets. In *Proceedings of the 2011 IEEE International Conference on Information Reuse and Integration*, pages 309–314. IEEE, 2011.

Huanjing Wang, Taghi M Khoshgoftaar, Randall Wald, and Amri Napolitano. A novel dataset-similarity-aware approach for evaluating stability of software metric selection techniques. In *Proceedings of the 13th International Conference on Information Reuse and Integration (IRI)*, pages 1–8. IEEE, 2012b.

Sida Wang and Christopher Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 118–126, 2013.

Brian J Wells, Amy S Nowacki, Kevin Chagin, and Michael W Kattan. Strategies for handling missing data in electronic health record derived data. *eGEMs (Generating Evidence & Methods to improve patient outcomes)*, 1(3):1035–1035, 2013.

World Health Organization (WHO). ICD Revision Timelines, Department of Health Statistics and Information Systems, World Health Organization (WHO). http://www.who.int/classifications/icd/revision/timeline/en/, 2017. Accessed: 2017-02-13.

Joanna L Whyte, Nicole M Engel-Nitz, April Teitelbaum, Gabriel Gomez Rey, and Joel D Kallich. An evaluation of algorithms for identifying metastatic breast, lung, or colorectal cancer in administrative claims data. *Medical care*, 53(7):e49–e57, 2015.

F Widmer. Comorbidity in heart failure. *Therapeutische Umschau. Revue therapeutique*, 68(2):103–106, 2011.

Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research*, 30(1):79, 2005.

Hannah Wong, Robert C Wu, George Tomlinson, Michael Caesar, Howard Abrams, Michael W Carter, and Dante Morra. How much do operational processes affect hospital inpatient discharge rates? *Journal of public health*, 31(4):546–553, 2009.

Hannah J Wong, Robert C Wu, Michael Caesar, Howard Abrams, and Dante Morra. Real-time operational feedback: daily discharge rate as a novel hospital efficiency metric. *Quality and Safety in Health Care*, 19(6):1–5, 2010.

Hua Xu, Zhenming Fu, Anushi Shah, Yukun Chen, Neeraja B Peterson, Qingxia Chen, Subramani Mani, Mia A Levy, Qi Dai, and Josh C Denny. Extracting and integrating data from entire electronic health records for detecting colorectal cancer cases. In *AMIA Annu Symp Proc*, volume 2011, pages 1564–1572, 2011.

Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):187–193, 2012.

Lisiane Pranjul Yadav, Sanjoy Andrew, Bonnie Katherine, Vipin Connie, and Gyorgy Simon Michael. Modelling trajectories for diabetes complications. In *Proceedings of the SIAM Workshop on Data Mining for Medicine and Healthcare*, pages 1–6. SDM, 2015a.

Pranjul Yadav, Michael Steinbach, Vipin Kumar, and Gyorgy Simon. Mining electronic health records (EHR): A survey. Technical report, Department of Computer Science and Engineering, University of Minnesota, 2015b.

Laura M Yamokoski, Vic Hasselblad, Debra K Moser, Cynthia Binanay, Ginger A Conway, Jana M Glotzer, Karen A Hartman, Lynne W Stevenson, and Carl V Leier.

Prediction of rehospitalization and death in severe heart failure by physicians and nurses of the escape trial. *Journal of cardiac failure*, 13(1):8–13, 2007.

Jieping Ye and Jun Liu. Sparse methods for biomedical data. *ACM SIGKDD Explorations Newsletter*, 14(1):4–15, 2012.

Jinn-Yi Yeh and Wen-Shan Lin. Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department. *Expert Systems with Applications*, 32(4):1073–1083, 2007.

Bin Yu. Stability. *Bernoulli*, 19(4):1484–1500, 2013.

Lei Yu, Chris Ding, and Steven Loscalzo. Stable feature selection via dense feature groups. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 803–811. ACM, 2008.

Lei Yu, Yue Han, and Michael E Berens. Stable gene selection from microarray data via sample weighting. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(1):262–272, 2012.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006.

Ming Yuan, V Roshan Joseph, and Hui Zou. Structured variable selection and estimation. *The Annals of Applied Statistics*, pages 1738–1757, 2009.

Lun Zhang, Qiuchen Liu, Wenchen Yang, Nai Wei, and Decun Dong. An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia-Social and Behavioral Sciences*, 96:653–662, 2013.

Min Zhang, Chen Yao, Zheng Guo, Jinfeng Zou, Lin Zhang, Hui Xiao, Dong Wang, Da Yang, Xue Gong, Jing Zhu, et al. Apparently low reproducibility of true differential expression discoveries in microarray studies. *Bioinformatics*, 24(18):2057–2063, 2008.

Xiang Zhang, Feng Pan, and Wei Wang. Finding high-order correlations in high-dimensional biological data. In *Link Mining: Models, Algorithms, and Applications*, pages 505–534. Springer, 2010.

Di Zhao and Chunhua Weng. Combining PubMed knowledge and EHR data to develop a weighted Bayesian network for pancreatic cancer prediction. *Journal of biomedical informatics*, 44(5):859–868, 2011.

Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, pages 3468–3497, 2009.

Jiayu Zhou, Jimeng Sun, Yashu Liu, Jianying Hu, and Jieping Ye. Patient risk prediction model via top-k stability selection. In *Proceedings of the 13th SIAM International Conference on Data Mining*, pages 55–63. SIAM, 2013.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

Copyright Information