BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

MACHINE LEARNING FOR EFFECTIVE PREDICTIONS AND PRESCRIPTIONS IN HEALTH CARE

by

TINGTING XU

B.S., China University of Mining and Technology (Beijing), 2011
M.S., University of Chinese Academy of Sciences, 2014

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2020

ProQuest Number: 27835696

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 27835696

Published by ProQuest LLC (2020). Copyright of the Dissertation is held by the Author.

All Rights Reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

© 2020 by
TINGTING XU
All rights reserved

Approved by

First Reader

Ioannis Ch. Paschalidis, Ph.D.
Professor of Electrical and Computer Engineering
Professor of Systems Engineering
Professor of Biomedical Engineering
Professor of Computing & Data Sciences

Second Reader

Christos G. Cassandras, Ph.D.
Distinguished Professor of Engineering
Professor and Division Head of Systems Engineering
Professor of Electrical and Computer Engineering

Third Reader

Brian Kulis, Ph.D.
Associate Professor of Electrical and Computer Engineering
Associate Professor of Computer Science
Associate Professor of Systems Engineering

Fourth Reader

Francesco Orabona, Ph.D.
Assistant Professor of Electrical and Computer Engineering
Assistant Professor of Computer Science
Assistant Professor of Mathematics and Statistics
Assistant Professor of Systems Engineering

I should like to say two things, one intellectual and one moral.
The intellectual thing I should want to say to them, is this: When you are studying any matter, or considering any philosophy, ask yourself only what are the facts and what is the truth that the facts bear out. Never let yourself be diverted, either by what you wish to believe, or by what you think would have beneficent social effects if it were believed. But look only, and solely, at what are the facts. That is the intellectual thing that I should wish to say.
The moral thing I should wish to say to them is very simple: I should say, love is wise, hatred is foolish. In this world, which is getting more and more closely interconnected, we have to learn to tolerate each other, we have to learn to put up with the fact that some people say things that we don't like. We can only live together in that way – and if we are to live together and not die together – we must learn a kind of charity and a kind of tolerance, which is absolutely vital to the continuation of human life on this planet. Bertrand Arthur William Russell

Acknowledgements

Pursuing a Ph.D. has been an important and long journey in my life. I am very grateful for the help and support I received from my advisor, teachers, friends and family along the way, without whom this PhD thesis would not have been possible.

First of all, I would like to express my most sincere gratitude to my advisor, Prof. Ioannis Ch. Paschalidis, for his everlasting support and guidance. I have always been impressed by his vision and passion for the frontiers of scientific research, inspiring me to explore and solve challenging problems. His profound theoretical background and enthusiasm for solving practical problems have set a good example for me. He was also very patient and supportive, and gave me plenty of time to learn and explore new areas. I learned a lot from him, including research, presentation, communication, time management skills, etc. All in all, a big thank you to my advisor, I was very lucky to have such an amazing mentor and friend during my Ph.D. study.

I would like to thank my dissertation committee Prof. Christos Cassandras, Prof. Brian Kulis and Prof. Francesco Orabona for providing so many valuable suggestions. Their knowledge and insights have been a great inspiration to my work. In addition to my committee, I also want to thank Prof. Prakash Ishwar, Prof. W. Clem Karl, Prof. David Castañón, Prof. Michael C. Caramanis for their lectures, talks and conferences that have taught me so much.

Thanks to my collaborators for their time and efforts: Taiyao Wang, Henghui Zhu, Wuyang Dai, Theodora S. Brisimi, Alexis De La Veiga, Shruthi Mahalingaiah, William G. Adams, Elpida Velmahos, George Kasotakis. The discussions with them have benefited me greatly. Also, thanks to Bill Adams and Galina Lozinski at Boston Medical Center for their support in the diabetes-related projects, Victor Escott at eIVF/Practice Highway and Denny Sakkas at BIVF at the New England Fertility Society for their support in the IVF outcome prediction research.

I would like to thank the staff at SE and CISE, Elizabeth Flagg, Ruth Mason, Cheryl Stewart, Christina Polyzos, Denise Joseph and Maureen Stanton, for their kind assistance and service in making my studies and life run smoothly at BU.

Thanks also to the current and past lab mates of the Network Optimization & Control (NOC) Lab: Wuyang Dai, Jing Wang, Qi Zhao, Theodora Brisimi, Jing Zhang, Hao Liu, Shahrooz Zarbafian, Taiyao Wang, Henghui Zhu, Yanying Zhao, Ruidi Chen, Athar Roshandelpoor, Xiangyu Meng, Athanasios Tsiligkaridis, Salomon Wollenstein-Betech, Zhiyu Zhang, Shahabeddin Sotudian, Jimmy Queeney. It's an honor for me to be with these wonderful lab mates.

Interacting with other friends has also benefited me greatly, and the fun times I've had with them will stay in my memory forever: Nan Zhou, Baichuan Zhou, Xiaoxuan Wu, Qianqian Ma, Yue Zhang, Wanqing Yang, Xinmiao Sun, Ruiqi Li, Liangxiao Xin, Yuting Chen, Weiwei Tao, Guancun Qu, David Zhou, Zhenyu Liao, Ruizhe Zhang, Victor Cao, Feng Nan, Huanyu Ding, Xi Yu, Rui Liu, Ye Lin, Athar Roshandelpoor, Christy Lin, Xiao Wang, Siyue Wang, Ruizhao Zhu, Boran Hao, Yang Hu, Parisa Babaheidarian, Andrew Cutler, Noushin Mehdipour, F. Selin Yanikara, Rebecca Swaszek, Francisco Álvarez.

Last but not least, I would like to give special thanks to my parents Li and Zhenxuan, sister Yingying and brother Jiancheng, and especially my husband Qiaobin, for their unconditional, eternal support and love. This thesis is dedicated to them.

Tingting Xu

Division of Systems Engineering

MACHINE LEARNING FOR EFFECTIVE PREDICTIONS AND PRESCRIPTIONS IN HEALTH CARE

TINGTING XU

Boston University, College of Engineering, 2020

Major Professor: Ioannis Ch. Paschalidis, Ph.D.
Professor of Electrical and Computer Engineering
Professor of Systems Engineering
Professor of Biomedical Engineering
Professor of Computing & Data Sciences

ABSTRACT

Early detection of acute hospitalizations and enhancing treatment efficiency is important to improve patients' long-term life quality and reduce health care costs. This thesis develops data-driven methods to predict important health related events and optimize treatment options. Applications include predicting chronic-disease-related hospitalizations, predicting the effect of interventions, such as In Vitro Fertilization (IVF), and learning and improving upon physicians' prescription policies.

For a binary hospitalization classification problem, and to strike a balance between accuracy and interpretability of the prediction, a novel Alternating Clustering and Classification (ACC) method is proposed, which employs an alternating optimization approach that jointly identifies hidden patient clusters and adapts classifiers to each cluster. Convergence and out-of-sample guarantees for this algorithm are established. The algorithm is validated on large data sets from the Boston Medical Center, the largest safety-net hospital system in New England.

For the IVF outcome prediction problem, and for women who have difficulty conceiv-

ing, several predictive models that estimate IVF success rate are designed. For predicted non-pregnant subjects, an algorithm further predicts whether no embryos were implanted (due to embryo abnormalities) or pregnancy did not occur despite implantation. Results are presented to assess the sensitivity of the models to specific predictive variables.

The third problem considered amounts to modeling the patients' disease progression as a Markov Decision Process (MDP), and seeking to estimate the physicians' prescription policy and the disease state transition probabilities. Two regularized maximum likelihood estimation algorithms for learning the transition probability model and policy, respectively, are proposed. A sample complexity result that guarantees a low regret with a relatively small amount of training samples is established. The theoretical results are illustrated using a healthcare example.

Finally, the thesis develops a framework for learning and improving the pharmacological therapy algorithm used by physicians to treat type 2 diabetes, based on prescription data. First, the proposed approach predicts the outcomes of prescriptions using regression, and then a policy consistent with physicians' prescriptions using a parametric multi-class classification method is synthesized from data. Then, by optimizing over algorithm parameters in the prescription model, the algorithm is able to achieve better glucose control effects.

Contents

List of Tables

List of Figures

List of Abbreviations

ACC	.....	Alternating Clustering and Classication
AUC	.....	Area Under the Receiver Operating Characteristic Curve
AUCPR	.....	Area Under the Precision-Recall Curve
BMC	.....	Boston Medical Center
BMI	.....	Body Mass Index
CDC	.....	The Centers for Disease Control and Prevention
CI	.....	Confidence Interval
CPT	.....	Current Procedural Terminology
CT-LSVM	.....	Cluster Then Linear Support Vector Machines
CT-SLSVM	.....	Cluster Then Sparse Linear Support Vector Machines
E2	.....	Estradiol
EHR	.....	Electronic Health Record
EN	.....	Elastic Net Regression
ER	.....	Emergency Room
FSH	.....	Follicle Stimulation Hormone
GBDT	.....	Gradient-Boosted Decision Tree
HbA1c	.....	Glycated Haemoglobin
ICD9	.....	International Classification of Diseases - Ninth Revision
IVF	.....	In vitro Fertilization
KL-divergence	.....	Kullback-Leibler Divergence
KNN	.....	k-Nearest Neighbor
L1LR	.....	L1-regularized Logistic Regression
L2LR	.....	L2-regularized Logistic Regression
LASSO	.....	Least Absolute Shrinkage and Selection Operator
Ob/gyn	.....	Obstetrics/Gynecology
R^2	.....	Coefficient of Determination
RF	.....	Random Forests
RBF	.....	Radial Basis Function
ROC	.....	Receiver Operating Characteristic
SLSVM	.....	Sparse Linear Support Vector Machine
SVM	.....	Support Vector Machine
VC	.....	Vapnik-Chervonenkis
WkNN	.....	weighted k-Nearest Neighbor
XGBoost	.....	Extreme Gradient Boosting

Chapter 1

Introduction

Health care is unquestionably an important global issue. In 2013, the United States (U.S.) spent about $3 trillion on health care, which exceeded 17% of its GDP [Kayyali et al., 2013]. The World Health Organization estimates that healthcare costs will grow up to 20% of the U.S. GDP (nearly $5 trillion) by 2021 [Bipartisan Policy Center, 2012], especially with civilization diseases (or else called lifestyle diseases), like diabetes, coronary heart disease and obesity, growing.

Our work aims at enabling personalized interventions using an algorithmic data-driven approach, based on the patients' medical history represented in their *Electronic Health Records (EHRs)*. An important enabler of our works is the increasing availability of patients' EHRs. The digitization of patients' records started more than two decades ago. Widespread adoption of EHRs (also called Electronic Medical Records – EMRs), has generated massive datasets. 87% of U.S. office-based physicians were using EHRs by the end of 2015, up from 42% in 2008 [Office of the National Coordinator for Health Information Technology, 2016]. EHRs have found diverse uses [Ludwick and Doucette, 2009], e.g., in assisting hospital quality management [Takeda et al., 2003], in detecting adverse drug reactions [Hannan, 1999], and in general primary care [Wang et al., 2003].

Specifically, there are two primary aims of this study. The first objective is to enable *predictive analytics* by exploring and developing Machine Learning algorithms. Applications of predictive analytics to healthcare include predicting hospitalizations to identify high-risk patients and predicting the effects of interventions. Prediction, naturally, is an

important first step towards decision making. It allows health systems to target individuals most in need and to allocate limited health resources more effectively. The second topic of our work seeks to provide a framework of learning and improving upon physicians' prescription policies, which enables personalized treatment recommendations with better intervention effects. This can not only improve the efficiency of treatments, but also potentially reduce the pain and economic burden of patients. Specifically, this thesis contains the subsequent four studies.

1.1 Predictive Analytics of Patients' Hospitalizations

Our interest in hospitalizations is motivated by [Jiang et al., 2006], which found that nearly $30.8 billion in hospital care cost during 2006 was preventable. Diabetes was one of the leading contributors accounting for almost $6 billion, or about 20%. Clearly, even modest percentage reductions in the amount matter. We seek to predict diabetes-related hospitalizations based on the patients' EHRs within a year from the time we examine the EHRs, so as to allow enough lead time for prevention. What is also critical is that our methods provide an interpretation (or explanation) of the predictions. Interpretability will boost the confidence of patients and physicians in the results, hence, the chance they will act based on the predictions, and provide insight into potential preventive measures. Interpretability is being increasingly recognized as important; for instance, European Union legislation has recently enforced a citizen's right to receive an explanation for algorithmic decisions [Voigt and Von dem Bussche, 2017].

Predictive analytics targets predicting future outcomes based on existing historical variables and is attracting widespread interest due to the large volume of data created by technology advances [Khalifa and Zabani, 2016]. Among the predictive models, classification and regression models are applied to predicting discrete and continuous target outcomes, respectively. For example, regression models have been developed to improve the predic-

tion of death and length of stay in the ICU [Rouzbahman et al., 2017], as well of the subcutaneous glucose concentration in Type 1 diabetes patients [Georga et al., 2012]. Efficient classification algorithms have been designed to predict cancer prognosis [Kourou et al., 2015] and asthma-related emergency department visits [Ram et al., 2015]. There are a variety of classification methods, including kernelized and sparse support vector machines (SVMs), sparse logistic regression, and random forests, etc. Deep learning models are well known for their accuracy but their outputs are hard to interpret - a major shortcoming in medical research.

We consider the problem of predicting patients' hospitalization as a binary classification problem, where the negative class represents the healthy people and the positive class the opposite. It is intuitive that people may get sick for various reasons (viewed as different clusters) while healthy people are healthy in every aspect (forming only one cluster). Thus, patients naturally arise from different groups with various age, sex, race or diseases. Mathematically, we assume that the positive (hospitalized) class consists of multiple clusters, whereas the samples of the negative (non-hospitalized) class are drawn from a single distribution. From a learning perspective, if the hidden groups are not predefined and we would like to learn an optimal group partition in the process of training classifiers, the problem could be viewed as a combination of clustering and classification.

The key contributions of this work are:

1. For a specific class of binary classification problems, we design a novel alternating clustering and classification (ACC) algorithm, which achieves favorable classification accuracy compared to other alternatives. In addition, the theoretical performance guarantees on the convergence and sample complexity of the proposed algorithm are discussed.

2. The proposed ACC algorithm is able to identify hidden clusters in the positive (hospitalized) class, which can not only improve the classification accuracy, but also

increase the interpretability of experimental results and help better understand the heterogeneity of the hospitalized patients.

1.2 In-Cycle Success Rate Prediction During the First In Vitro Fertilization (IVF) Cycle

According to the Centers for Disease Control and Prevention (CDC), about 10% (6.1 million) of women aged 15-44 in the United States have impaired fertility [Centres for Disease Control and Prevention, 2019], which brings them enormous psychological burdens. Fortunately, human reproductive capacity has been significantly improved through the development of assisted reproductive technology (ART), which is a group of methods to surgically remove the egg from the ovary and fertilize it with sperm in order to produce embryos. In vitro fertilization (IVF) is one of the most popular ART therapies. The process of presenting for infertility evaluation and management, ultimately requiring IVF therapy may be time-consuming, emotionally burdensome, and may require significant financial resources in states with no mandated insurance coverage. Predicting the IVF success rate has traditionally fallen on the society for assisted reproductive technology (SART) national success rate data, which is important for both doctors and patients to make an informed decision about the best choice of treatment [The Society for Assisted Reproductive Technology (SART), 2016].

In this study, we aim to provide predictive models to estimate the in-cycle success rate prediction during the first IVF treatment cycle based on fertility-related variables such as age, egg-related variables, sperm-related variables, hormones and lifestyle-related variables, etc. The key contributions of this work are:

1. Our models can accurately predict the IVF success rate. We also provide a simple and sparse predictive model with high accuracy only based on the most important variables. For predicted non-pregnant subjects, we further predict whether no embryos

were implanted (due to embryo abnormalities) or pregnancy did not occur despite implantation with high accuracy.

2. Our methods assess the sensitivity of the models to specific predictive variables. Based on the analysis of variable importance in the predictive models, we find a number of informative variables consistent with the existing literature.

1.3 Learning Parametric Policies and Transition Probability Models of MDPs From Data

In the health care domain, disease progression can be modeled as a Markov Decision Processes (MDP) with states corresponding to the condition of the patient and actions associated with physician actions, drugs prescribed, etc. Typically, one does not have access to good models of state transitions and computing a good policy requires *exploration*, which can be done through expensive clinical trials. Yet, there is a wealth of information in *Electronic Health Records (EHRs)*, containing doctors' prescriptions and patients' relevant medical information (demographics, diagnoses, etc). Leveraging this information to learn a prescription policy can help homogenize care across multiple locations and reinforce best practices. In addition, it is also critical to estimate a disease progression model under various treatments. Given a policy and a model one could ask many what-if questions and use a variety of methods to improve the policy.

Markov Decision Processes (MDPs) provide a framework for solving dynamic optimization problems under uncertainty. When the cardinality of the state-action space is small and the transition probabilities are known, one can easily calculate an optimal policy using value iteration or policy iteration. However, in the health care applications, an MDP has a very large state-action space, and it is not even realistic to assume knowledge of the model, especially the transition probabilities for all state-action pairs.

In an increasing number of settings, it has become possible to collect large amounts

of data by observing (state, action, next state) tuples of an agent who uses an unknown policy. In such "data-rich" settings, the problem we consider is to learn the (unknown) original policy used by the agent and also obtain a good estimate of the transition probabilities. Clearly, and because collecting data is cumbersome and expensive, it is of interest to develop efficient learning methods that can handle limited data availability.

The key contributions of this work are:

1. To achieve the best estimates of (disease state) transition dynamics and the associated (prescription) policy in an MDP from data, we design two regularized logistic regression methods. We assume the policy and transition probabilities are "Boltzmann-type" parametric functions, and the parameters can be derived using maximum likelihood estimation.

2. Our proposed algorithm achieves a regret of $O(\sqrt{\varepsilon})$ with $\Omega(\log(n)\text{poly}(1/\varepsilon))$ samples, where n indicates the number of adopted features used to represent the policy and $\text{poly}(\cdot)$ denotes a polynomial function. This result implies that by utilizing only a relatively small amount of training data, the estimated policy performs similarly to the unknown original (prescription) policy whose state-action pairs we observe. The sample complexity guarantee is more valuable especially in health care applications where it is expensive to obtain training samples.

1.4 Learning Models for Writing Better Doctor Prescriptions

Urban living in modern large cities has significant adverse effects on health, increasing the risk of several chronic diseases, among which diabetes is the worldwide fastest growing chronic disease. On average, people with diagnosed diabetes have medical costs 2.3 times higher than expenditures of people without diabetes. The total costs of diagnosed diabetes have risen to $327 billion in 2017, up from $245 billion in 2012, representing a 26% increase over a five-year period. Treatment in the form of anti-hypoglycemic medications to

directly treat diabetes costs about $30.8 billion annually [American Diabetes Association, 2018b]. Personalizing the pharmacological therapy recommendations for patients can be a promising approach to boost treatment efficiency and reduce the medication expenditure.

For diabetic patients, the first and foremost task is to properly control glycemia. Glycated hemoglobin (HbA1c) is a widely used measure of long-term glycemia which reflects average glycemia over a 3-month period. The HbA1c values in our work are reported in "Diabetes Control and Complications Trial" (DCCT) units and indicated by percentage, representing the percentage of glycated hemoglobin to total hemoglobin [Sacks, 2012]. There is a close relationship between the risks of complications and glycemia [American Diabetes Association, 2003, Stratton et al., 2000]. Specifically, for every percentage point decrease in HbAlc (e.g., from 9.0% to 8.0%), there is a 35% reduction in the risk of microvascular complications (retinopathy, neuropathy and diabetic nephropathy), a 25% reduction in diabetes-related deaths, a 7% reduction in all-cause mortality, and an 18% reduction in combined fatal and nonfatal myocardial infarction [American Diabetes Association, 2003]. Among all patients with diabetes, type 2 diabetes patients account for $90\% - 95\%$. [Stratton et al., 2000] shows that people with type 2 diabetes who reduce their HbA1c level by one percentage point are 19%, 16%, and 43% less likely to suffer cataracts, heart failure, and amputation or death due to peripheral vascular disease, respectively.

Glycemic control can be achieved by lifestyle changes and pharmacotherapy that targets HbA1c, as well as postprandial and fasting glucose levels [American Diabetes Association, 2018a]. The effect of the same pharmacotherapy treatment to lower HbA1c may vary from individual to individual, therefore, understanding the patient-specific prescription effect is vital for the best treatment decision. The first challenge is to evaluate the patient-specific effect of a prescription treatment. This could be achieved through clinical trials, however, since a clinical trial can only involve a single treatment, it is impossible to quantize the effect of various prescription treatments on the same person under the same conditions, not

to mention that conducting clinical trials for a large amount of patients is both economically ineffective and time-consuming.

Our work aims at providing a framework to enable personalized prescriptions, recommending treatments with the best patient-specific effect. Specifically, we consider the problem of evaluating the patient-specific effect of prescriptions, learning the current physicians' prescription policy from data, and improving this policy by optimizing over its parameters. Our work can be seen as combining predictive and prescriptive analytics [Khalifa and Zabani, 2016]. Predictive analytics only focuses on outcome prediction, but are short of personalized prescription recommendations. Prescriptive analytics determines the most effective interventions to achieve desired outcomes. Such models not only predict consequences and future performance of the decisions, but also provide recommendations for effective actions [Khalifa and Zabani, 2016]. They have broad applications in the field of business, Internet technology and academic research, such as asset health management [Goyal et al., 2016] and automotive software architecture [Eliasson et al., 2015].

The key contributions of this work are:

1. We design a model which accurately predicts physicians' prescription policy using the patients' EHRs. For each patient, based on his/her EHR history, the data-based model is able to suggest a personalized treatment following the learned physicians' prescription policy.

2. We provide an approach which improves the current prescription policy, based on both the treatment effect regression model and the learned physicians' prescription policy model.

1.5 Baseline Classification Algorithms

In this section, we outline several baseline classification methods we use in the following chapters. In medical applications, accuracy is important, but also interpretability of the

predictions is indispensable [Vellido Alcacena et al., 2012], strengthening the confidence of medical professionals in the results. Sparse classifiers are interpretable, since they provide succinct information on few dominant features leading to the prediction [Lee et al., 2006]. Moreover, medical data sets are often imbalanced since there are much fewer patients with a condition (e.g., hospitalized) versus "healthy" individuals (nonhospitalized). This makes it harder for supervised learning methods to learn since a training set may be dominated by negative (nonhospitalized) class samples. Sparsity, therefore, is useful in this context because there are fewer parameters in the classifier one needs to learn. In this light, we experiment with sparse versions of various classification methods and show their advantages. While harder to interpret than linear and sparse algorithms, ensemble methods that build collections of classifiers, such as random forests, can model nonlinear relationships and have been proven to provide very accurate models for common healthcare problems [Casanova et al., 2014].

1.5.1 RBF, Linear & Sparse Linear Support Vector Machines

A Support Vector Machine (SVM) is an efficient binary classifier [Cortes and Vapnik, 1995, Scholkopf et al., 1997]. The SVM training algorithm seeks a separating hyperplane in the feature space, so that data points from the two different classes reside on different sides of that hyperplane. We can calculate the distance of each input data point from the hyperplane. The minimum over all these distances is called *margin*. The goal of SVM is to find the hyperplane that has the maximum margin. In many cases, however, data points are neither linearly nor perfectly separable. So called *soft-margin* SVM, tolerates misclassification errors and can leverage kernel functions to "elevate" the features into a higher dimensional space where linear separability is possible (*kernelized SVMs*) [Cortes and Vapnik, 1995].

Given our interest in interpretable, hence sparse, classifiers we formulate a sparse version of linear SVM (SLSVM) as follows. We are given training data $\mathbf{x}_i \in \mathbb{R}^D$ and labels $y_i \in \{-1, 1\}$, $i = 1, \ldots, n$, where $\mathbf{x}_i$ is the vector of features for the ith patient and $y_i = 1$

(resp., $y_i = -1$) indicates that the patient will (resp., not) be hospitalized. We seek to find the classifier (β, β_0), $\beta \in \mathbb{R}^D, \beta_0 \in \mathbb{R}$ by solving:

$$\begin{array}{ll} \min\limits_{\beta,\beta_0,\xi_i} & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \xi_i + \rho\|\beta\|_1 \\ \text{s.t.} & \xi_i \geq 0, \quad \forall i, \\ & y_i(\mathbf{x}_i'\beta + \beta_0) \geq 1 - \xi_i, \quad \forall i, \end{array} \tag{1.1}$$

where ξ_i is a misclassification penalty. The first term in the objective has the effect of maximizing the margin. The second objective term minimizes the total misclassification penalty. The last term $\|\beta\|_1$ term in the objective, imposes sparsity in the feature vector β, thus allowing only a sparse subset of features to contribute to the classification decision. The parameters C and ρ are tunable parameters that control the relative importance of the misclassification and the sparsity terms, respectively, compared to each other and, also, the margin term. When $\rho = 0$, the above formulation yields a standard linear SVM classifier.

A linear SVM finds a linear hyperplane in the feature space and cannot handle well cases where a nonlinear separating surface between classes is more appropriate. To that end, kernel functions are being used that map the features to a higher dimensional space where a linear hyperplane would be applicable. In the absence of the sparse-inducing ℓ_1-norm term, kernelized SVMs use $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)'\phi(\mathbf{x}_i)$ as a kernel for some feature mapping function ϕ and solve an optimization problem that is based on the dual of (1.1) to find an optimal (β, β_0). In our applications, we will employ the widely used Radial Basis Function (RBF) $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$ [Scholkopf et al., 1997] as the kernel function in our experiments.

1.5.2 Random Forests

A random forest (RF) is an ensemble of de-correlated trees [Friedman et al., 2001, Hastie et al., 2015, Bishop, 2006a]. Each decision tree is formed using a training set obtained by sampling (with replacement) a random subset of the original data. While growing each decision tree, random forests use a random subset of the set of features (variables) at each

node split. Essentially, the algorithm uses bagging for both trees and features, where bagging (or bootstrap aggregating) is a technique for reducing the variance of an estimated predictor by averaging many noisy but approximately unbiased models. Each decision tree is fully grown until a minimum size is reached, i.e., there is no pruning. While the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Bagging achieves de-correlating the trees by constructing them using different training sets. To make a prediction at a new sample, random forests take the majority vote among the outputs of the grown trees in the ensemble. Random forests run very efficiently on large datasets, do not have the risk of overfitting (as, e.g., Adaboost [Freund and Schapire, 1995], a boosting method) and can handle datasets with unbalanced classes. The number of trees in the ensemble is selected through cross-validation.

1.5.3 Sparse Logistic Regression

Logistic Regression (LR) [Bishop, 2006b] is a linear classifier widely used in many classification problems. It models the posterior probability of an event of interest as a logistic function of a linear combination of the input features, with parameters θ that weigh the input features and an offset θ_0. The parameters of the model are selected by maximizing the log-likelihood using a gradient method. For the test samples, decisions are made by thresholding the log-likelihood ratio of the positive (hospitalized) class over the negative class. Logistic regression is popular in the medical literature because it predicts a probability of a sample belonging to the positive class. Here, we use an ℓ_q-regularized (sparse) logistic regression [Lee et al., 2006, Pudil et al., 1994], which adds an extra penalty term proportional to $\|\theta\|_q$ in the log-likelihood, where $\|\theta\|_q = (\sum_{i=1}^{n} |\theta_i|^q)^{1/q}$ denotes the q-norm of vector θ. The motivation is to induce sparsity, effectively "selecting" a sparse subset of features. More specifically, we solve the following convex problem using a gradient-type

method:

$$\min_{\theta} \quad \sum_{i=1}^{n} -\log p(y_i|\mathbf{x}_i;\theta,\theta_0) + \lambda\|\theta\|_q \tag{1.2}$$

where the likelihood function is given by

$$p(y_i = 1|\mathbf{x}_i;\theta,\theta_0) = \frac{1}{1+e^{-\theta_0-\theta'\mathbf{x}_i}}$$

the parameter θ represents the weight of the input variables and $q \geq 1$. λ is a tunable parameter controlling the sparsity term. When $\lambda = 0$, we obtain a standard logistic regression model. Settings $q = 1$ and $q = 2$ represent L1-regularized and L2-regularized logistic regression, respectively.

1.5.4 Performance Evaluation

For binary classification models, we report two performance metrics AUC and AUCPR, which are the areas under the receiver operating characteristic (ROC) curve and the area under the precision-recall curve (PRC), respectively. A random binary classifier can achieve an AUC of 0.5 and AUCPR as the proportion of positive samples [Saito and Rehmsmeier, 2015, Davis et al., 2005, Lever et al., 2016]. Compared to AUC, AUCPR puts more weights towards prediction accuracy of positive samples because it also considers the proportion of true positive samples in all positive predictions [Saito and Rehmsmeier, 2015, Davis et al., 2005]. In some studies, we also provide ROC (AUC) and PRC (AUCPR) curves to enable a comprehensive understanding of the classification performance.

Notation: By convention, we use lower case bold letters for all vectors and upper case bold letters for all matrices. Unless otherwise specified, all vectors are column vectors. For economy of space, we write $\mathbf{x} = \left(x_1, \ldots, x_{\dim(\mathbf{x})}\right)$ to denote the column vector $\mathbf{x}$, where $\dim(\mathbf{x})$ is the dimension of $\mathbf{x}$. We use "prime" to denote the transpose of a matrix or vector and $|\mathcal{D}|$ the cardinality of a set $\mathcal{D}$. Unless otherwise specified, $\|\cdot\|$ denotes the ℓ_2 norm and $\|\cdot\|_1$ the ℓ_1 norm. $\|\mathbf{x}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}$ denotes the p-norm of vector $\mathbf{x}$. For a matrix $\mathbf{P}$,

$\|\mathbf{P}\|_\infty$ denotes the maximum absolute row sum. Sets are denoted by script letters.

Chapter 2

Predictive Analytics of Patients' Hospitalizations

In the healthcare domain, it is intuitive that people may get sick for various reasons (viewed as different clusters) while healthy people are healthy in every aspect (forming only one cluster). Thus, patients naturally arise from different groups with various age, sex, race or diseases. From a learning perspective, if the hidden groups are not predefined and we would like to learn an optimal group partition in the process of training classifiers, the problem could be viewed as a combination of clustering and classification.

As we already outlined in Section 1.1, the hospitalization prediction problem is considered as a binary classification problem, where the negative class represents the healthy people and the positive class the opposite. We assume that the positive (hospitalized) class consists of multiple clusters, whereas the samples of the negative (non-hospitalized) class are drawn from a single distribution. For each positive cluster, we assume there is a distinct and sparse set of discriminative dimensions, based on which it is separated from the negative class (cf. Fig. 2·1) and, therefore, the "local boundary" (classifier) could naturally be assumed to be different and lying in a lower-dimensional subspace of the feature vector. The common supervised learning methods can certainly make classifications without considering the hidden clusters, but the hidden clusters could be useful in assisting classification and leading to better classification results. Furthermore, with the identified hidden groups, the classification model becomes more interpretable in addition to accurately generating classification labels.

In the literature, there are generally two types of assumptions about hidden clusters in

a classification problem: implicit or explicit. The implicit assumption is more frequent and examples can be found e.g., in piecewise linear techniques [Pele et al., 2013], in feature space partitioning methods [Breiman et al., 1984], or when learning a mixture model of different SVMs applied to the data [Fu et al., 2010]. None of these methods have clustering as the goal, clustering is just a by-product of their classification models. An explicit consideration of clusters within a classification model is proposed in [Gu and Han, 2013], where training samples are first put into clusters and then separate classifiers are trained. Due to sequential steps, clustering does not take advantage of label information. Therefore, the advantage of these methods is mainly to speed up the training phase.

The target of our problem requires simultaneous cluster identification and classification. Our related work has been described in [Dai et al., 2015, Brisimi et al., 2018, Xu et al., 2016]. This chapter is organized as follows. Section 2.1 presents formally the joint clustering and classification problem we would like to solve and proposes two alternative solution methodologies, while also commenting on how well they scale with the size of the problem. Section 2.2 analyzes the Alternating Clustering and Classification framework we propose, establishing convergence, sample complexity and generalization guarantees. The proposed methods are tested on simulated data in Section 2.3, as well as, real-world healthcare data in Section 2.4. Specifically, we aim at predicting future hospitalizations based on patients' EHRs. Conclusions are in Section 2.5.

2.1 Problem Formulation

The classification problem under consideration satisfies the following assumptions:

- The negative class samples are assumed to be i.i.d. and drawn from a single cluster with distribution P_0.
- The positive class samples belong to L clusters, with distributions $P_1^1, \cdots, P_1^L$.

- Different positive clusters have different features that separate them from the negative samples.

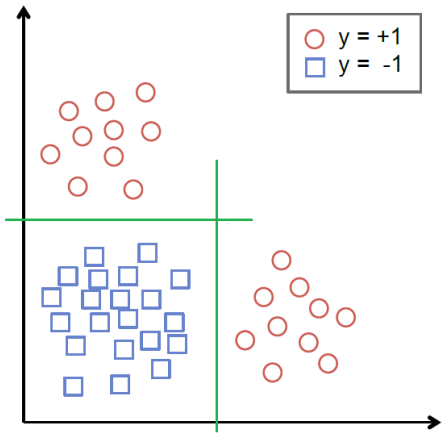

Figure 2·1: The positive class contains two clusters and each cluster is linearly separable from the negative class.

We propose a joint cluster detection and classification problem under a Sparse Linear SVM (SLSVM) framework. Let $\mathbf{x}_i^+$ and $\mathbf{x}_j^-$ be the D-dimensional positive and negative feature vectors (samples), and y_i^+, y_j^- the corresponding labels, where $i \in \{1,2,\ldots,N^+\}$, $j \in \{1,2,\ldots,N^-\}$, and $y_i^+ = 1,\ \forall i, y_j^- = -1,\ \forall j$. Assuming L hidden clusters in the positive class, we seek to discover: (a) the L hidden clusters, denoted by a mapping function $i \to l(i)$, $l(i) \in \{1,2,\ldots,L\}$, and (b) L classifiers, one for each cluster. Let T^l be a parameter controlling the sparsity of the classifier for each cluster l. We formulate the *Joint Clustering and Classification (JCC)* problem as follows:

$$
\begin{aligned}
\min_{\substack{\beta^l,\beta_0^l,l(i)\\ \zeta_j^l,\xi_i^l}} \quad & \sum_{l=1}^{L}\left(\tfrac{1}{2}||\beta^l||_2^2 + \lambda^+ \sum_{i:l(i)=l} \xi_i^{l(i)} + \lambda^- \sum_{j=1}^{N^-} \zeta_j^l\right) \\
\text{s.t.} \quad & \sum_{d=1}^{D} |\beta_d^l| \le T^l, \qquad \forall l, \\
& \xi_i^{l(i)} \ge 1 - y_i^+\beta_0^{l(i)} - \sum_{d=1}^{D} y_i^+\beta_d^{l(i)} x_{i,d}^+, \quad \forall i, \\
& \zeta_j^l \ge 1 - y_j^-\beta_0^l - \sum_{d=1}^{D} y_j^-\beta_d^l x_{j,d}^-, \quad \forall j,l, \\
& \xi_i^{l(i)},\ \zeta_j^l \ge 0, \qquad \forall i,j,l,
\end{aligned}
\tag{2.1}
$$

where $y_i^+ = 1,\ \forall i$ and $y_j^- = -1,\ \forall j$.

In formulation (2.1), the empirical costs of the negative samples are counted L times because they are drawn from a single distribution and, as a result, they are not clustered but simply copied into each cluster. Parameters λ^- and λ^+ control the weights of costs from the negative and the positive samples. The constraint $\sum_{d=1}^{D} |\beta_d^l| \leq T^l$ is an ℓ_1-relaxation of the sparsity requirement to the local classifiers, which aligns the formulation with the problem assumptions and estimates more robust local classifiers.

We propose two approaches to solve (2.1): (a) to transform the joint problem into a *Mixed Integer Programming* problem (hereafter referred to as MIP), which suffers however from scaling limitations, and (b) to alternately train a classification model and then re-cluster the positive samples, which scales well and also provides theoretical performance guarantees.

2.1.1 Mixed Integer Programming Formulation

We introduce binary indicator variables to represent the cluster assignment into the JCC formulation (2.1) since each positive sample can only be assigned to one cluster, and transform the original problem into an equivalent MIP:

$$
\begin{aligned}
\min_{\substack{\beta^l,\beta_0^l,z_{il}\\ \zeta_j^l,\xi_i^l}} \quad & \sum_{l=1}^{L}\left(\tfrac{1}{2}\|\beta^l\|_2^2+\lambda^+\sum_{i=1}^{N^+}\xi_i^l+\lambda^-\sum_{j=1}^{N^-}\zeta_j^l\right)\\
\text{s.t.} \quad & \sum_{d=1}^{D}|\beta_d^l|\leq T^l, \qquad \forall l,\\
& \xi_i^l \geq 1-y_i^+\beta_0^l-\sum_{d=1}^{D}y_i^+\beta_d^l x_{i,d}^+ - M\sum_{k\neq l}z_{ik}, \quad \forall i,l,\\
& \zeta_j^l \geq 1-y_j^-\beta_0^l-\sum_{d=1}^{D}y_j^-\beta_d^l x_{j,d}^-, \quad \forall j,l,\\
& \sum_{l=1}^{L}z_{il}=1,\ \forall i;\ z_{il}\in\{0,1\},\ \forall i,l;\ \xi_i^l,\ \zeta_j^l\geq 0,\forall i,j,l;
\end{aligned}
\tag{2.2}
$$

where $z_{il}=1$ when $l(i)=l$ and 0 otherwise (binary variables describing the cluster assignments), $y_i^+=1,\ \forall i$, $y_j^-=-1,\ \forall j$ and M is a large positive real number.

Theorem 2.1.1. *The MIP formulation (2.2) is equivalent to the original JCC formulation (2.1).*

Proof. Let C^*_{JCC} and C^*_{MIP} be the optimal objective values of problems (2.1) and (2.2) .

Given any feasible solution to the JCC problem $l(i), \beta^l, \beta^l_0, \zeta^l_i, \forall l, i$, and $\xi^{l(i)}_{i,JCC}$, a feasible solution to the MIP problem is:

$$z_{il} = \begin{cases} 1, \ l(i) = l, \\ 0, \ otherwise, \end{cases} \qquad \xi^l_{i,MIP} = \begin{cases} \xi^l_{i,JCC}, \ l(i) = l, \\ 0, \ otherwise; \end{cases}$$

and $\beta^l, \beta^l_0, \zeta^l_i$ remain the same as in the JCC solution.

The feasibility of the constructed MIP solution is verified as follows. Notice that with the exception of the second constraint in the MIP formulation (2.2) (the big-M constraint), all other constraints can be easily verified to be satisfied by the constructed MIP solution. For the big-M constraint, if $z_{il} = 1$, then $M \sum_{k \neq l} z_{ik} = 0$, and the big-$M$ constraint holds since $\xi^l_{i,MIP} = \xi^l_{i,JCC}$. If, however, $z_{il} = 0$, then $M \sum_{k \neq l} z_{ik} = M$, and the big-$M$ constraint also holds (trivially).

The above two feasible solutions have the same objective value, and this equality holds for any feasible solution to the JCC problem, hence we can conclude that $C^*_{JCC} \geq C^*_{MIP}$.

Next, we prove that each optimal solution to MIP problem satisfies $\xi^l_{i,MIP} = 0$ when $z_{il} = 0$. Note that when $z_{il} = 0$, $M \sum_{k \neq l} z_{ik} = M$, and the big-$M$ constraint becomes $\xi^l_{i,MIP} \geq 1 - y^+_i \beta^l_0 - \sum_{d=1}^{D} y^+_i \beta^l_d x^+_{i,d} - M$, which will always hold since M is a large enough number. Therefore, to minimize the objective, the optimal solution should select the smallest feasible $\xi^l_{i,MIP}$, i.e., $\xi^l_{i,MIP} = 0$.

Given an optimal solution to MIP problem, a corresponding feasible solution to JCC problem is: if $z_{il} = 1$, then $\xi^l_{i,JCC} = \xi^l_{i,MIP}$, and $l(i) = l$; and all other variables retain their values in the MIP solution. Since the two solutions have the same objective cost, it follows $C^*_{JCC} \leq C^*_{MIP}$. □

In order to obtain better clustering performance, we introduce a penalty term in the objective function seeking to minimize the within-cluster distances (making samples in the same cluster more similar to each other): $\rho \sum_{i_1=1}^{N^+} \sum_{i_2=1}^{N^+} \sigma_{i_1 i_2} \|x^+_{i_1} - x^+_{i_2}\|^2$, where

$$\sigma_{i_1 i_2} = \begin{cases} 1, \ \text{if } x^+_{i_1} \text{ and } x^+_{i_2} \text{ belong to the same cluster,} \\ 0, \ \text{otherwise.} \end{cases}$$

For σ to follow this definition, we need to impose the constraint $z_{i_1 l} + z_{i_2 l} - \sigma_{i_1 i_2} \leq 1, \ \forall \ i_1 \neq i_2, l$ and $\sigma_{i,j} \in \{0, 1\}$.

The MIP approach presented above comes in a compact form, solves jointly the clustering and the classification problem, and exhibits good performance on small-scale problems (see Sec. 2.3). However, there are no general polynomial-time algorithms for solving MIPs, thus, making it problematic for large datasets that are most common in real problem instances. This motivates us to develop the following *Alternating Clustering and Classification (ACC)* approach, which does not suffer from these limitations.

2.2 Alternating Clustering and Classification

Given cluster assignments $l(i)$ for all positive samples, the JCC problem (2.1) can be decoupled into L separate quadratic optimization problems. This motivates our *Alternating Clustering and Classification (ACC)* approach (Alg.1-Alg.3) with two major modules: (i) training a classifier for each cluster and (ii) re-clustering positive samples given all the estimated classifiers. The process starts with a random cluster assignment of the positive clusters and then alternates between the two modules. Note that the constraint (2.3) in Alg. 2 is necessary for proving the convergence of ACC. Lastly, the re-clustering of the positive samples is based on $\mathcal{C}$, a subset of $\{1,\ldots,D\}$, which is a set of selected features that allows us to add prior knowledge about the clusters so that the identified clusters provide intuitive explanations. In a notational remark, we denote $\mathbf{x}_{i,\mathcal{C}}^{+}$ (resp., $\mathbf{x}_{\mathcal{C}}$) as the projection of the D-dimensional feature vector $\mathbf{x}_i^{+}$ (resp., $\mathbf{x}$) on the subset $\mathcal{C}$.

In the simulation and real data experiments we conduct, we compare our new algorithm ACC to SVMs (linear and RBF), and two other hierarchical approaches that combine clustering with classification, named Cluster-Then-Linear-SVM (CT-LSVM) and Cluster-Then-Sparse-Linear-SVM (CT-SLSVM). Specifically, the CT-LSVM algorithm first clusters the positive samples based on the feature set $\mathcal{C}$ using the widely used k-means method [Friedman et al., 2001], then copies the negative samples into each cluster, and finally trains a Linear SVM classifier for each cluster. The only difference between CT-

Algorithm 1 ACC Training

Initialization:
Randomly assign positive class sample i to cluster $l(i)$, for $i \in \{1,\ldots,N^+\}$ and $l(i) \in \{1,\ldots,L\}$.
repeat
Classification Step:
Train an SLSVM classifier for each cluster of positive samples combined with all negative samples. Each classifier is the outcome of a quadratic optimization problem (cf. (2.7)) and provides a hyperplane perpendicular to β^l and a corresponding optimal objective value O^l .
Re-clustering Step:
Re-cluster the positive samples based on the classifiers β^l and update the $l(i)$'s.
until no $l(i)$ is changed or $\sum_l O^l$ is not decreasing.

SLSVM and CT-LSVM is that CT-SLSVM adopts Sparse Linear SVM classifiers in the last step.

Notice that ACC has an alternating procedure while CT-LSVM, CT-SLSVM do not. With only one-time clustering, CT-LSVM and CT-SLSVM create unsupervised clusters without making use of the negative samples, whereas ACC is taking class information and classifiers into consideration so that the clusters also help the classification.

2.2.1 ACC Theoretical Performance Guarantees

We begin by presenting a result that suggests a favorable sample complexity for SLSVM compared to the standard linear SVM. Suppose that SLSVM for the l-th cluster yields Q^l $(< D)$ non-zero elements of β^l, thus, identifying a Q^l-dimensional feature subspace used for classification. The value of Q^l is controlled by T^l. Assume we draw a training set with N^- negative samples from P_0 and N_l^+ positive samples from P_1^l, where $N^l = N_l^+ + N^-$. Let R_N^l denote the expected training error rate and R^l the expected test error under these distributions.

Theorem 2.2.1. *For a sparse linear SVM lying in a Q-dimensional subspace of the original*

Algorithm 2 Re-clustering procedure given classifiers

Input: positive samples $\mathbf{x}_i^+$, classifiers β^l, current cluster assignment which assigns sample i to cluster $l(i)$.
for all $i \in \{1,\ldots,N^+\}$ **do**

for all $l \in \{1,\ldots,L\}$ **do**
calculate the projection a_i^l of positive sample i onto the classifier for cluster l using only elements in $\mathcal{C}$: $a_i^l = \mathbf{x}_{i,\mathcal{C}}^{+\prime}\beta_{\mathcal{C}}^l$;
end for
update cluster assignment of sample i from $l(i)$ to
$l^*(i) = \arg\max_l a_i^l$, subject to

$$\mathbf{x}_i^{+\prime}\beta^{l^*(i)} + \beta_0^{l^*(i)} \;\geq\; \mathbf{x}_i^{+\prime}\beta^{l(i)} + \beta_0^{l(i)}. \tag{2.3}$$

end for

Algorithm 3 ACC Testing

for each test sample **x** **do**
Assign it to cluster $l^* = \arg\max_l \mathbf{x}_{\mathcal{C}}{}'\beta_{\mathcal{C}}^l$.
Classify **x** with β^{l^*}.
end for

D-dimensional space, for any $\varepsilon > 0$ *and* $\delta \in (0,1)$*, if the sample size* N^l *satisfies*

$$N^l \geq \frac{8}{\varepsilon^2}\left((Q^l+1)log\frac{2eN^l}{Q^l+1} + Q^l log\frac{eD}{Q^l} + log\frac{2}{\delta}\right), \tag{2.4}$$

then with probability no smaller than $1-\delta$*,* $R^l - R_N^l \leq \varepsilon$*.*

Proof. We will use a result from [Bousquet et al., 2004]. We note that the family of linear classifiers in a D-dimensional space has VC-dimension $D+1$ ([Vapnik, 1998]). Let $\mathcal{G}$ be a function family with VC-dimension $D+1$. For ease of notation we will drop the reference to the l-th cluster as the result applies to all clusters. Let $R_N(g)$ denote the training error rate of classifier g on N training samples randomly drawn from an underlying distribution $\mathcal{P}$. Let $R(g)$ denote the expected test error of g with respect to $\mathcal{P}$. The following theorem from [Bousquet et al., 2004] is useful in establishing our result.

Theorem 2.2.2 ([Bousquet et al., 2004])**.** *If the function family* $\mathcal{G}$ *has VC-dimension* $D+1$*,*

then the probability

$$P\left[R(g)-R_N(g)\leq 2\sqrt{2\frac{(D+1)\log\frac{2eN}{D+1}+\log\frac{2}{\rho}}{N}}\right]\geq 1-\rho \tag{2.5}$$

for any function $g\in\mathcal{G}$ *and* $\rho\in(0,1)$.

For the given ε select large enough N such that

$$\varepsilon\geq 2\sqrt{2((D+1)\log(2eN/D+1)+\log(2/\rho))/N}.$$

Thm. 2.2.2 implies that

$$\rho\leq 2\exp\left((D+1)\log\frac{2eN}{D+1}-\frac{N\varepsilon^2}{8}\right)$$

and

$$P\left(R(g)-R_N(g)\geq\varepsilon\right)\leq 2\exp\left((D+1)\log\frac{2eN}{D+1}-\frac{N\varepsilon^2}{8}\right),$$

where we may have to further increase N so that $\rho\in(0,1)$. If we let g lie in a Q-dimensional subspace, we have

$$P\left(R(g)-R_N(g)\geq\varepsilon\right)\leq 2\exp\left((Q+1)\log\frac{2eN}{Q+1}-\frac{N\varepsilon^2}{8}\right). \tag{2.6}$$

In our setting, the classifier g is restricted to a Q-dimensional feature subspace. The bound in (2.6) holds for any such Q-dimensional subspace selected by the optimization. Since there are $\binom{D}{Q}$ possible choices for the subspace, using the union bound and the inequality $\binom{D}{Q}\leq(\frac{eD}{Q})^Q=\exp(Q\log\frac{eD}{Q})$, we obtain:

$$\begin{aligned}P\left(R(g)-R_N(g)\geq\varepsilon\right)&\leq\binom{D}{Q}2\exp((Q+1)\log\frac{2eN}{Q+1}-\frac{N\varepsilon^2}{8})\\&\leq 2\exp\left(Q\log\frac{eD}{Q}+(Q+1)\log\frac{2eN}{Q+1}-\frac{N\varepsilon^2}{8}\right).\end{aligned}$$

In order to ensure $P\left(R(g)-R_N(g)\geq\varepsilon\right)\leq\delta$ for the given $\delta\in(0,1)$, we select large enough N with

$$N\geq\frac{8}{\varepsilon^2}\left((Q+1)\log\frac{2eN}{Q+1}+Q\log\frac{eD}{Q}+\log\frac{2}{\delta}\right),$$

which concludes the proof to Theorem 2.2.1. □

Theorem 2.2.3. *The ACC algorithm converges for any set $\mathcal{C}$.*

Proof. At each alternating cycle, for each cluster l we train a SLSVM with positive samples of that cluster combined with all negative samples. This produces an optimal value O^l and the corresponding classifier (β^l, β_0^l). Specifically, the formulation is:

$$\begin{aligned} O^l = \min_{\substack{\beta^l,\beta_0^l,\\ \zeta_j^l,\xi_i^l}} & \frac{1}{2}||\beta^l||^2 + \lambda^+ \sum_{i=1}^{N_l^+} \xi_i^l + \lambda^- \sum_{j=1}^{N^-} \zeta_j^l \\ \text{s.t. } & \xi_i^l \geq 1 - y_i^+ \beta_0^l - \textstyle\sum_{d=1}^D y_i^+ \beta_d^l x_{i,d}^+, \ \forall i; \\ & \zeta_j^l \geq 1 - y_j^- \beta_0^l - \textstyle\sum_{d=1}^D y_j^- \beta_d^l x_{j,d}^-, \ \forall j; \\ & \textstyle\sum_{d=1}^D |\beta_d^l| \leq T^l; \ \xi_i^l, \zeta_j^l \geq 0, \ \forall i, j. \end{aligned} \tag{2.7}$$

We use the sum of the optimal objective function values in (2.7) across different clusters to prove the convergence. We have

$$Z = \sum_{l=1}^{L} O^l = \sum_{l=1}^{L} \left(\frac{1}{2}||\beta^l||^2 + \lambda^- \sum_{j=1}^{N^-} \zeta_j^l \right) + \lambda^+ \sum_{i=1}^{N^+} \xi_i^{l(i)},$$

where $\sum_{l=1}^{L} N_l^+ = N^+$, $l(i)$ maps sample i to cluster $l(i)$, and β^l, β_0^l, ζ_j^l and $\xi_i^{l(i)}$ are optimal solutions of (2.7) for each l. Now, let us consider the change of Z at each iteration of the ACC procedure.

First, we consider the re-clustering step given SLSVMs. During the re-clustering step, the classifier and slack variables for negative samples are not modified. Only the $\xi_i^{l(i)}$ get modified since the assignment functions $l(i)$ change. When we switch positive sample i from cluster $l(i)$ to $l^*(i)$, we can simply assign value $\xi_i^{l(i)}$ to $\xi_i^{l^*(i)}$. Therefore, the value of Z does not change during the re-clustering phase and takes the form

$$Z = \sum_{l=1}^{L} \left(\frac{1}{2}||\beta^l||^2 + \lambda^+ \sum_{\{i:l^*(i)=l\}} \xi_i^l + \lambda^- \sum_{j=1}^{N^-} \zeta_j^l \right).$$

Next, given new cluster assignments we re-train the local classifiers by resolving problem (2.7) for each cluster l. Notice that re-clustering was done subject to the constraint in Eq. (2.3) (see Alg. 2). Since $y_i^+ = 1$, we have

$$\xi_i^{l(i)} \geq 1 - \beta_0^{l(i)} - \sum_{d=1}^{D} \beta_d^{l(i)} x_{i,d}^+ \geq 1 - \beta_0^{l^*(i)} - \sum_{d=1}^{D} \beta_d^{l^*(i)} x_{i,d}^+.$$

The first inequality holds due to $\xi_i^{l(i)}$ being feasible for (2.7). The second inequality holds due to $y_i^+ = 1$ and Eq. (2.3) in Alg. 2. Thus, by assigning $\xi_i^{l(i)}$ to $\xi_i^{l^*(i)}$, it follows that the $\xi_i^{l^*(i)}$ remain feasible for problem (2.7). Given that the remaining decision variables do not change, $(\beta^l, \beta_0^l, \zeta_j^l, \xi_i^{l^*(i)}, \forall i = 1,\ldots,N_l^+, \forall j = 1,\ldots,N^-)$ forms a feasible solution of problem (2.7). This solution has a cost equal to O^l. Re-optimizing can produce an optimal value that is no worse. It follows that in every iteration of ACC, Z is monotonically non-increasing. Given that Z is bounded below by zero, we establish the convergence of ACC. □

As a remark on convergence, it is worth mentioning that the values λ^+ and λ^- should be fixed across all clusters to guarantee convergence.

Let $\mathcal{H}$ denote the family of clustering/classification functions produced by ACC.

Theorem 2.2.4. *The VC-dimension of $\mathcal{H}$ is bounded by*

$$V_{ACC} \stackrel{\triangle}{=} (L+1)L(D+1)\log\left(e\frac{(L+1)L}{2}\right). \tag{2.8}$$

Proof. The proof is based on Lemma 2 of [Sontag, 1998]. Given the L functions for clustering, named $g_1, g_2, \ldots, g_L$, the final cluster of a sample is determined by the maximum of g_1 to g_L. This clustering process could be viewed as the output of $(L-1)L/2$ comparisons between pairs of g_i and g_j, where $1 \le i < j \le L$. The pairwise comparison could be further transformed into a boolean function (i.e., $\text{sign}(g_i - g_j)$). Then together with the L classifiers for each cluster, we have a total of $(L+1)L/2$ boolean functions to make the final classification. Among all these boolean functions, the maximum VC-dimension is $D+1$. □

Theorem 2.2.4 implies that the VC-dimension of ACC-based classification grows linearly with the dimension of data samples and polynomially (between quadratic and cubic) with the number of clusters. Since the local (per cluster) classifiers are trained under an ℓ_1 constraint, they are typically defined in a lower dimensional subspace. At the same time, the clustering function also lies in a lower dimensional space $\mathcal{C}$. Thus, the "effective" VC-dimension could be smaller than the bound in Theorem 2.2.4.

An immediate consequence of Theorem 2.2.4 is the following corollary which establishes out-of-sample generalization guarantees for ACC-based classification and is based on a result in [Bousquet et al., 2004]. To state the result, let $N = N^+ + N^-$ the size of the training set. Let R_N denote the expected training error rate and R the expected test error (out-of-sample) of the ACC-based classifier.

Corollary 2.2.5. *For any* $\rho \in (0,1)$*, with probability at least* $1-\rho$ *it holds:*

$$R \leq R_N + 2\sqrt{2\frac{V_{ACC}\log\frac{2eN}{V_{ACC}} + \log\frac{2}{\rho}}{N}}.$$

2.3 Simulations

In this section, we validate the efficiency of MIP and ACC by experimenting on a small-scale and a large-scale synthetic dataset, respectively. The synthetic datasets follow the model assumptions described in Section 2.1.

2.3.1 Settings of Simulation Data

Let $D = 5$ and the negative class be simply a D-dimensional standard normally distributed random vector $\mathcal{N}(0, I_D)$. For the positive class, there are 4 clusters ($L = 4$) and let $\mathcal{C} = \{1,2,3,4\}$ in ACC, meaning the first 4 dimensions are used for clustering. The remaining dimension is elevated by 0.3 in the mean from the standard normal distribution. For each positive cluster, there is one dimension of $\mathcal{C}$ elevated to be normally distributed, $\mathcal{N}(3,4)$, and the remaining three cluster dimensions are still standard normally distributed.

In these synthetic datasets, imbalanced positive clusters are created to make the problem even harder and resemble situations appearing in practice. In the training phase of the small-scale (respectively, large-scale) dataset, 56 (resp., 560) samples are generated, including 28 (resp., 280) negative samples, 12 (resp., 120) samples equally split into the first 3 positive clusters and 16 (resp., 160) samples for the last positive cluster. 56 (resp.,

4200) samples are generated for testing in a similar way.

Settings of Tuning Parameters

Since the MIP approach cannot scale well, we compare the MIP with ACC only on the small-scale dataset. For larger datasets, we compare the ACC algorithm to SVMs (with a linear kernel and an RBF kernel) and the two hierarchical methods, CT-LSVM and CT-SLSVM. All comparisons are based on 50 repetitions of the simulations. The model parameters for all these methods are tuned through 3-fold cross validation with only training data.

For MIP in the small-scale experiments, we tune parameters λ^- in $(100, 10, 1)$ with $\lambda^+ = L\lambda^-$. T^l is fixed to 2 (based on preliminary experiments). L, M and ρ are fixed to $4, 100$ and 1 respectively, to save on computational cost.

For ACC parameters, in both small-scale and large-scale dataset experiments, we performed preliminary experiments to tune T^l and fix it to be 3. The tuning parameters are selected from λ^- in $(100, 10, 1, 0.1)$ with $\lambda^+ = L\lambda^-$. In the large-scale dataset, L is explicitly varied in $(2, 3, 4, 5, 6)$ to demonstrate the effect of the number of clusters in the ACC.

The penalty costs of the slack variables in linear SVM, RBF SVM, the linear SVM in CT-LSVM and sparse linear SVM in CT-SLSVM are also tuned among $(100, 10, 1, 0.1)$. Furthermore, the kernel width of RBF SVM is tuned among $(10, 3, 1, 0.3, 0.1)$. For CT-LSVM and CT-SLSVM, the number of clusters in k-means is set as the true number 4.

2.3.2 Performance Evaluation

The average prediction accuracy across 50 repetitions on small-scale and large-scale datasets are shown in Table 2.1 and Table 2.2 respectively. We use *Area Under the Receiver Operating Characteristic Curve (AUC)* as the criterion for accuracy, because it captures the tradeoff between false positives and false negatives.

In Table 2.1, the average accuracy (avg. AUC) and 95% Confidence Interval (CI) of AUC are presented. In Table 2.2, the number of clusters is varied and the average AUC is reported together with its standard deviations (std.) over 50 runs. The last column of Table 2.2 presents the percentage of repetitions that each method outperforms RBF SVM. From

Table 2.1: AUC on small-scale synthetic data.

Settings	avg. AUC	95% CI of AUC
MIP ($L = 4$)	75.98%	[73.75%, 78.21%]
ACC ($L = 4$)	76.00%	[73.37%, 78.65%]

Table 2.2: AUC on large-scale synthetic data.

Settings	avg. AUC	std. AUC	Percentage
ACC ($L = 2$)	79.62%	1.80%	80
ACC ($L = 3$)	80.80%	2.02%	84
ACC ($L = 4$)	81.25%	1.68%	86
ACC ($L = 5$)	81.59%	1.91%	86
ACC ($L = 6$)	81.95%	1.78%	90
Lin. SVM	74.50%	1.34%	22
RBF SVM	77.24%	3.40%	-
CT-LSVM (L=4)	77.41%	2.51%	48
CT-SLSVM (L=4)	77.07%	2.81%	46

Table 2.1 we observe that the MIP and ACC perform similarly. The results in Table 2.2 indicate the following:

- ACC outperforms SVMs (with linear and RBF kernel) and the two hierarchical methods for various L values.
- Larger L values (even larger than the true number of clusters) lead to better prediction accuracy.

- Setting L to a value smaller than the true number of clusters has a bigger impact on the average AUC performance compared to setting it to a value larger than the number of true clusters. This is intuitive and provides a rule of thumb for setting the L value in real applications.

The classification accuracy is only one aspect of our JCC method. Another important aspect is identifying the underlying clusters. Since ACC performs the best at $L = 6$, we examine the details of the clusters identified by ACC for each repetition of the experiment. Specifically, we examine the mean vectors of each cluster. If the clusters are correctly identified, each mean vector has only one element substantially larger than 0, and the elevated features across all clusters should cover the four features of the true underlying clusters. By using this criterion, we mathematically test whether clusters are correctly identified in each repetition of the experiment. It turns out that in 86% (43 out of 50) of the repetitions, ACC correctly identified the clusters.

2.4 Predicting Diabetes-related Hospitalizations

We apply our methodology to the problem of predicting whether a patient will be hospitalized (admitted to the hospital) within one year from examining the patients' medical history as captured in their EHRs. We cast the problem as a classification problem, classifying patients between hospitalized and non-hospitalized.

2.4.1 The Diabetes Dataset

The data used for the experiments come from Boston Medical Center (BMC). BMC is the largest safety-net hospital in New England and with 13 affiliated Community Health Centers (CHCs) provides care for about 30% of Boston residents. The population of the study consists of patients with at least one diagnosis record of diabetes mellitus (ICD9 code 250) between $01/01/2007$ and $12/31/2012$. For each patient in the above set, we extract

the medical history (demographics, visit history, problems, procedures and department information) during the period $01/01/2001 - 12/31/2012$. The data we process for these patients comes from the hospital EHR and billing systems, which record admissions or visits and the primary diagnosis/reason. The diabetes-related medical history of the patients is described by various categories of medical factors (that we identified using feedback from doctors), which, along with some examples corresponding to each, are shown in Table 2.3. Overall, this dataset consists of $40,921$ patients.

In more detail, with every patient visit to the hospital, at least one record with a medical factor and a time-stamp containing the admittance date (and the discharge date) is created. In order to organize all the information available in some uniform way for all patients, some pre-processing of the data is required. Details will be discussed in the next subsection. We will refer to the summarized information of the medical factors over a specific time interval as features. Each feature related to Diagnoses, Procedures CPT (Current Procedural Terminology), Procedures ICD9 (International Classification of Diseases, 9th edition) and visits to each Department is an integer count of such records for a specific patient during the specific time interval. Zero indicates the absence of any record.

2.4.2 Data Pre-processing

The features are formed as combinations of different medical factors (instead of considering the factors as separate features) that better describe what happened to the patients during their visits to the hospital. Specifically, we formulate triplets that consist of a diagnosis, a procedure (or the information that no procedure was done) and the service department. An example of a complex feature (a triplet) is the diagnosis of ischemic heart disease that led to an adjunct vascular system procedure (procedure on a single vessel) while the patient was admitted to the inpatient care. Clearly, since each category can take one of several discrete values, a huge number of combinations should be considered. Naturally, not all possible combinations occur, which reduces significantly the total number of potential features that

Table 2.3: Medical factors.

Ontology	Examples
Demographics	Gender, Age, Race
Diagnoses	e.g., Diabetes mellitus with complications, Thyroid disorders, Hypertensive disease, Pulmonary heart disease, Heart failure, Aneurysm, Skin infections, Abnormal glucose tolerance test, Family history of diabetes mellitus
Procedures (CPT or ICD9)	e.g., Procedure on single vessel, Insertion of intraocular lens prosthesis at time of cataract extraction, Venous catheterization, Hemodialysis, Transfusion of packed cells
Admissions	e.g., Diabetes (with and without) complications, Heart failure and shock, Deep Vein Thrombophlebitis, Renal failure, Chest pain, Chronic obstructive pulmonary disease, Nutritional & misc metabolic disorders, Bone Diseases & Arthropathies, Kidney & urinary tract infections, Acute myocardial infarction, O.R. procedures for obesity, Hypertension
Laboratory Test Values	Hematology, Chemistry, Urinalysis, Coagulation tests
Vital Signs	e.g., Blood pressure, Pulse, Respiratory rate, Temperature, Body Mass Index (BMI)
Blood Glucose Regulation Agents	Insulin, Anti-hypoglycemic, Oral hypoglycemic agents, etc.
Service By Department	Inpatient (admit), Inpatient (observe), Outpatient, Emergency Room

describe each patient. Also for each patient, we extract information about the diabetes type over their history (keeping only patients with type 2) and demographics including age, gender and race.

Next, we present several data organization and pre-processing steps we take. For each patient, a target year is fixed and all past patient records are organized as follows.

(a) *Setting the target time interval to a calendar year.* Based on some preliminary experiments we conducted, we observed that there is greater variability in the results when trying to predict hospitalizations in periods of time shorter than a year (e.g., predicting hospitalization in the next 1, 3 or 6 months). Thus, we have designed our experiment to predict hospitalizations in the target time interval of a year starting on the 1st of January and ending on the 31st of December.

(b) *Selection of the target year.* As a result of the nature of the data, the two classes are highly imbalanced. To increase the number of hospitalized patient examples, if a patient had only one hospitalization throughout 2007-2012, the year of hospitalization will be set as the target year. If a patient had multiple hospitalizations, a target year between the first and the last hospitalizations will be randomly selected. 2012 is set as the target year for patients with no hospitalization, so that there is as much available history for them as possible.

(c) *Removing patients with no record.* Patients who have no records before the target year are removed, since there is nothing on which a prediction can be based. The total number of patients left is $33,122$, including the $26,478$ patients with type 2 diabetes under consideration. After this process, the proportion of hospitalized patients with type 2 diabetes in the dataset is 13.48% (3570 out of $26,478$).

(d) *Forming the complex features.* We create a diagnoses-procedures-service department indicator triplet (complex feature) to keep track of which diagnosis occurs with which procedure and service department. The procedures that are not associated with any diabetes-related diagnosis are removed. Diagnoses in the dataset are listed in the most detailed level of the ICD9 coding system. We group together procedures that belong to the same ICD/CPT family, resulting in 31 categories (out of 2004 in total).

(e) *Summarization of the complex features in the history of a patient.* We form four time blocks for each medical factor. Time blocks 1, 2, and 3 summarize the medical factors over one, two, and three years before the target year, whereas the 4th time block averages all earlier records. Naturally not all combinations of diagnoses-procedures-service department occur, and we only keep the triplets that occur; then adding the demographic features produces a 9402-dimensional vector of features characterizing each patient.

(f) *Reducing the number of complex features.* We remove all the features that do not contain enough information for a significant amount of the population (less than 1% of the patients), as they could not help us generalize. This leaves 320 complex medical and 3 demographic features.

(g) *Other detailed information.* We also consider 245 more detailed medical features, including lab test values, vital signs and blood glucose regulation agents (see Table II). By calculating the average lab test values, average vital signs or existence of regulation agents in the 4 time blocks, we obtain $245 \times 4 = 980$ additional features. Removing features with standard deviation less than $1E-4$, reduces the number of features to 945. Together with the features we described earlier, this results in $945+3+320 = 1268$ features.

(h) *Identifying the diabetes type.* The ICD9 code for diabetes is assigned to category 250 (diabetes mellitus). The fifth digit of the diagnosis code determines the type of diabetes and whether it is uncontrolled or not stated as uncontrolled. Keeping only the $26,478$ patients with type 2 diabetes, we have two types of diabetes diagnoses: type 2, not stated as uncontrolled (fifth digit 0), and type 2 or unspecified type, uncontrolled (fifth digit 2). Based on these types, we count how many records of each type each patient had in the four time blocks before the target year, thus adding 8 new features for each patient.

(i) *Splitting the data into a training set and a test set randomly.* As is common in supervised machine learning, the population is randomly split into a training and a test set. Since from a statistical point of view, all the data points (patients' features) are drawn from the same distribution, we do not differentiate between patients whose records appear earlier in time than others with later time stamps. A retrospective/prospective approach appears more often in the medical literature and is more relevant in a clini-

cal trial setting, rather than in our algorithmic approach. What matters in our setting is that for each patient prediction we make (hospitalization/non-hospitalization in a target year), we only use that patient's information before the target year.

2.4.3 Performance Evaluation

In the diabetes-related hospitalization prediction problem, hospitalized patients form the positive class while non-hospitalized the negative class. We randomly select 40% of all samples as training set and the remaining 60% are used for testing. This random splitting is repeated 10 times. The dataset is highly imbalanced as there are many more non-hospitalized patients than hospitalized. While the Receiver Operating Characteristic (ROC) curves could provide misleading interpretation of specificity when utilized in imbalanced classification cases [Saito and Rehmsmeier, 2015, Davis et al., 2005], the Precision-Recall Curve (PRC) presents a more accurate measurement of imbalanced classification performance because it also considers the fraction of true positive samples among all positive predictions [Saito and Rehmsmeier, 2015, Davis et al., 2005]. It is worth noting that unlike the ROC curve, the PRC is not guaranteed to be monotonic [Lever et al., 2016]. The Area Under the ROC Curve (AUC) and the Area Under the Precision-Recall Curve (AUCPR) are summary statistics, taking values between 0 and 1, that allow us to compare different ROC and PRC curves, respectively. The higher the AUC (AUCPR) value of the ROC (PRC) curve, the better. Consequently, for the imbalanced diabetic dataset, we present both ROC (AUC) and PRC (AUCPR) curves to enable a comprehensive understanding of the classification performance.

We evaluate classification performance out-of-sample, i.e., in a test set not seen during training. Figure 2 and Figure 3 plot ROC and PRC curves for a variety of classification methods, respectively; Table III and Table IV list the corresponding AUCs and AUCPRs (average and standard deviation of AUC and AUCPR over 10 runs with different training and test sets). Parameter tuning was done for all methods using k-fold cross validation.

For ACC, the initial assignments of the positive samples to the clusters are obtained from k-means clustering, and multi-start is implemented to find the best local optimum. The parameters as used in (1) are set as follows. The number of clusters L explicitly takes its values from 2, 3, 4 for all methods involving clustering, the soft-margin parameter for the negative class λ^- takes its values from 100, 10, 1, 0.1 and the soft-margin parameter for the positive class λ^+ is set equal to $L\lambda^-$. Some preliminary experiments led us to set the sparsity-controlling parameter $T^l = 12$ to save computational cost. For ACC, we employ one more innovation to improve the prediction results. Specifically, for each cluster, we solve several instances of the per-cluster sparse SVM as follows. First, we solve the problem with all features and a fixed T^l. This has the effect of selecting a subset of the features. Then, we solve a 2nd instance of the problem using only the subset of the features selected. We keep iterating in this fashion until a relatively small subset of features is being used.

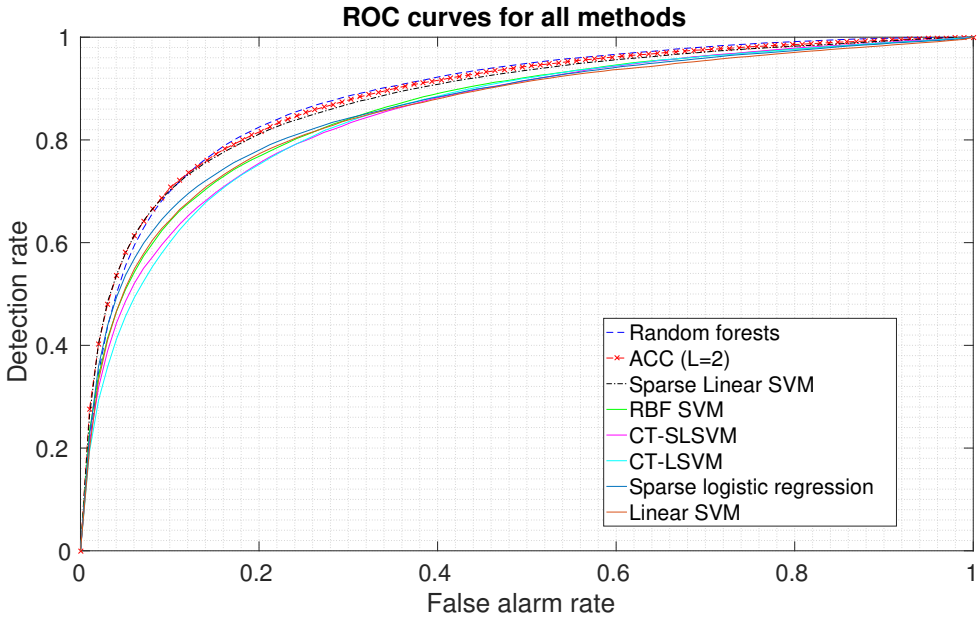

Figure 2·2: ROC curves for various classification methods.

For all methods 40% of the data are used for training and the rest for testing. The train-

ing data are normalized to have zero mean and unit standard deviation and are balanced by down-sampling the negative population. We also compare ACC with two other hierarchical approaches that first cluster the data using the k-means clustering[36] and then perform the classification task using linear SVM (we denote the method as CT-LSVM) and sparse (l_1-regularized) linear SVM (we denote the method as CT-SLSVM). Only the best results for CT-LSVM (obtained under L=2) and CT-SLSVM (obtained under L=2) are presented.

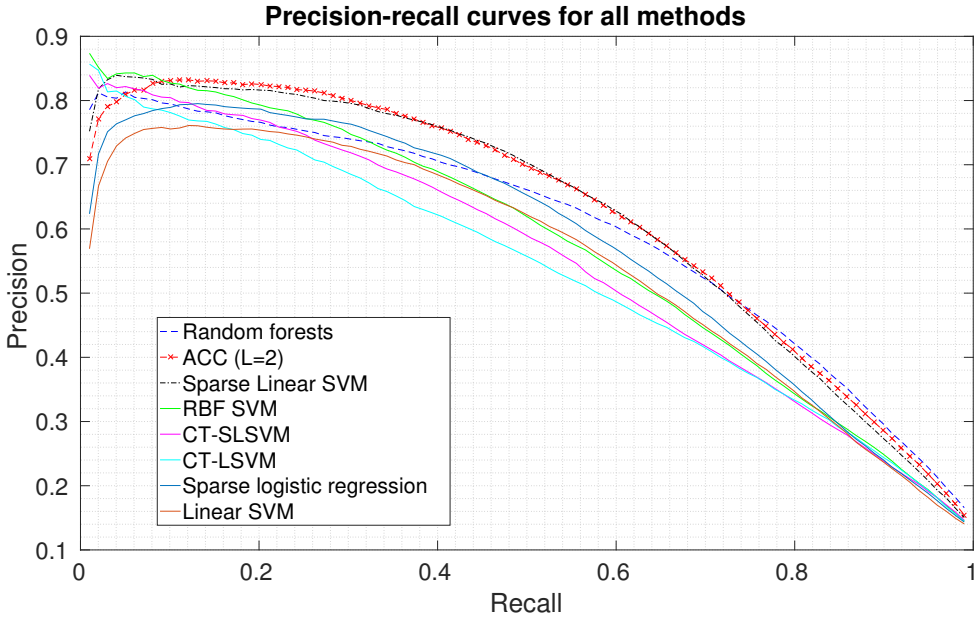

Figure 2·3: PRC curves for various classification methods.

Clustering with ACC can use a subset of "diagnostic" features (subset C in Alg. 2), since these are the features that better delineate across different types of diabetes complications. We base, however, the clustering in our experiments on all features due to the fact that almost all triplet features are related to "diagnostic" features. Although a random forest has a slight advantage over other methods in terms of AUC in Table III, it produces a very complex classifier, lacks interpretability, and has lower AUCPR than the ACC classifiers. Furthermore, as seen from Figure 3, ACC has significantly larger precision than the random forest when recall is relatively small. In Table III and Table IV, ACC has the best AUCPR

and the second best AUC among all methods, which demonstrates its superior classification accuracy in this imbalanced dataset, and it is able to detect the hidden positive clusters and identify why a specific patient is labeled as hospitalized. Among ACC variants, the best performance is obtained for L=2 clusters, with the performance of the other variants using more clusters being close behind. The fact that ACC (L=2) is better than ACC (L=1) illustrates that appropriate clustering can not only produce meaningful cluster interpretations, but also improve classification performance. It is interesting that ACC performs quite well even though the resulting classifiers are relatively sparse and do not use many features. This also makes them easy to implement. Notice that ACC utilizes sparse linear SVM as the base classifier. ACC also proved to be efficient from a computational point of view, since in our implementation, it is faster than random forests by a factor of 3.

Table 2.4: Average (avg) and standard deviation (std) of the area under roc curve (AUC) of various methods we have experimented with over 10 runs.

Method	Avg AUC	Std AUC	Method	Avg AUC	Std AUC
ACC (L=1)	0.8814	0.0025	Linear SVM	0.8531	0.0029
ACC (L=2)	0.8861	0.0032	RBF SVM	0.8594	0.0037
ACC (L=3)	0.8829	0.0039	sparse logistic regression	0.8613	0.0027
ACC (L=4)	0.8812	0.0027	Random Forests	0.8882	0.0054
CT-SLSVM (L=2)	0.8522	0.0034	CT-LSVM (L=2)	0.8502	0.0072

In an attempt to interpret the ACC clusters, we list in Table V the mean value over each cluster of the features used by the per-cluster classifiers. This is done for a single repetition of the experiment and L=2, yielding interesting clusters and highlighting the interpretative power of ACC. We concentrate on the most distinguishable features in the clusters. Specifically, for each feature we used Welch's t-test to compute a two-tailed p-value, where the null hypothesis was that the two cohorts (patients in cluster 1 and cluster 2) have equal means. All the features listed in Table V have a p-value less than 0.001,

Table 2.5: Average and standard deviation of the area under precision-recall curve (AUCPR) of various methods we have experimented with over 10 runs.

Method	Avg AUCPR	Std AUCPR	Method	Avg AUCPR	Std AUCPR
ACC (L=1)	0.6200	0.0104	Linear SVM	0.5512	0.0129
ACC (L=2)	0.6214	0.0106	RBF SVM	0.5758	0.0110
ACC (L=3)	0.6085	0.0115	sparse logistic regression	0.5752	0.0091
ACC (L=4)	0.6035	0.0109	Random Forests	0.6003	0.0159
CT-SLSVM (L=2)	0.5518	0.0124	CT-LSVM (L=2)	0.5355	0.0175

and the small p-value indicates strong evidence against the null hypothesis. ACC assigns 51.87% of hospitalized patients in the training set to cluster 2 and the remaining to cluster 1. We observe that hospitalized patients in Cluster 1 are older, have more hypertension and heart failure (measured in avg. number of diagnoses), take more drugs for heart diseases (measured in avg, number of drugs taken), and have indicators of renal disease (higher serum creatinine values and higher blood urea nitrogen). Hospitalized patients in Cluster 2 have diabetes with not as significant heart disease complications, indicated as diabetes with no associated procedures, hospitalizations, and, in general, not stated as uncontrolled. This is quite interesting and consistent with earlier work that identified the relationship between diabetes and specific complications (heart disease in our case). [18, 37] It appears, that the method is identifying a cluster of patients with diabetes and heart disease and is using a different classifier for these patients compared to the remaining patients.

2.5 Conclusions

In this work, we focused on the challenge of predicting future hospitalizations for patients with diabetes problems, based on their EHRs. We explored a diverse set of methods, namely kernelized, linear and ℓ_1-regularized linear SVMs, ℓ_1-regularized logistic regres-

Table 2.6: Average feature values in the clusters produced by ACC (L=2).

Variables	Mean in Cluster 1	Mean in Cluster 2	p-value	Variables	Mean in Cluster 1	Mean in Cluster 2	p-value
Age	72.98	66.11	1.04E-18	Creatinine, serum	1.27	0.96	7.85E-11
Hypertensive Diseases (diagnoses)	1.52	1.16	5.44E-04	Glucose, point of care	148.65	162.23	7.33E-38
Heart Failure (diagnoses)	0.31	0.09	5.69E-09	Platelet count	226.4	263.63	1.67E-44
Diabetes Mellitus No Procedure Emergency Room	0.13	0.23	6.57E-07	Review/Order Lab tests	0.43	0.25	2.70E-19
Diabetes Mellitus No Procedure In Patient (Observe)	0.94	1.48	2.92E-10	Review/Order Radiology	0.23	0.14	3.87E-08
Diabetes Type 2, not stated as uncontrolled	1.86	3.7	1.73E-25	Review/Order other tests	0.22	0.1	2.34E-17
Cardiology-related Medicine	0.75	0.18	3.12E-16	Urea nitrogen, blood	21.77	16.8	3.42E-26

sion, and random forests. Our main contribution is the introduction of a novel joint clustering and classification method that discovers hidden clusters in the positive samples (hospitalized) and identifies sparse classifiers for each cluster separating the positive samples from the negative ones (nonhospitalized). The joint problem is nonconvex (formulated as an integer optimization problem); still we developed an alternating optimization approach (termed ACC) that can solve very large instances. We established the convergence of ACC, characterized its sample complexity, and derived a bound on VC dimension that leads to out-of-sample performance guarantees. For all the methods we proposed, we evaluated their performance in terms of classification accuracy and interpretability, an equally crucial criterion in the medical domain. Our ACC approach yielded the best performance among methods that are amenable to an interpretation (or explanation) of the prediction. Our findings highlight a number of important insights and opportunities by offering a more targeted strategy for "at-risk" individuals. Our algorithms could easily be applied to care management reports or EHR-based prompts and alerts with the goal of identifying individuals who might benefit from additional care management and outreach. With a 20% false alarm rate, we can correctly predict about 81% of the hospitalized patients while providing insight as to why each prediction is made. Because costs associated with preventive actions (such as tests, medications, office visits) are orders of magnitude lower than hospitalization costs, one can tolerate significant false alarm rates and still save a large amount of money in preventable hospitalization costs. The proposed algorithm has wider applicability and the potential to be applied to other medical case studies, helping, for example, discover cohorts of patients with similar underlying issues and devising cohort-specific predictive models.

Chapter 3

In-Cycle Success Rate Prediction During the First IVF Cycle

Many efforts to establish predictive models to estimate the likelihood of success before and during the IVF treatment cycle have been undertaken, however they have yet to find a complete model that predicts IVF success rate based on patients' demographics, vital signs, lifestyle, diagnosis and other fertility-related variables. The existing works do not appear to be accurate enough for external validation studies. Some works can be characterized as cross studies based on many existing studies, hence the heterogeneity of different studies makes the comprehensive results less reliable [Broekmans et al., 2006]. Other works only investigate how the IVF success rate will be impacted by one or few variables, such as age, anti-müllerian hormone (AMH) and follicle stimulation hormone (FSH), rather than considering all available variables [Templeton et al., 1996, Sunkara et al., 2011, La Marca et al., 2010]. Some studies provide limited prediction accuracy (AUC less than 0.6) [Broer et al., 2013, Lukaszuk et al., 2013, Bjercke et al., 2005, Choi et al., 2013]. Many studies utilize a very small dataset with only hundreds of samples in the model training phase [Blank et al., 2019,Hafiz et al., 2017,Gianaroli et al., 2013]. Most of the available predictive models in the literature are based on very complex algorithms like ranking algorithms, Random Forest, Bayesian networks, etc., which are difficult to be interpreted or applied in real medical applications [Blank et al., 2019, Hafiz et al., 2017, Gianaroli et al., 2013, Güvenir et al., 2015].

3.1 IVF Data

The IVF dataset used for the experiments has been collected from multiple IVF centers in Massachusetts, New York and California, including $41,771$ subjects with $71,440$ IVF cycles. By examining this vast IVF dataset, we have designed several predictive models to determine the subjects' success rate during their first IVF treatment cycle. Although there is no doubt that the pregnancy success rate increases with more IVF treatment cycles [Smith et al., 2015], the extra time and expense for multiple IVF cycles is a huge burden for many women. Therefore, the success rate of the first IVF cycle is an important representative measure that can inform the decision of undergoing IVF treatment. To assess the probability of success when using the subjects' own eggs, we eliminate the records involving donated eggs and ensure that all predictors are from the subjects' own data.

We remove records of subjects who have no embryos transferred (implanted) because they decided to freeze all embryos, or planned a cycle for future implantation, or had no eggs/sperm, because such decisions result in no pregnancy irrespective of the subjects' fertility potential. We keep records of subjects who have no embryos transferred due to embryo abnormalities, which, clearly, also result in no pregnancy.

Overall, there are $26,040$ subjects and IVF cycles satisfying the selection criteria, among which $11,586$ subjects become pregnant and the rest $14,454$ subjects do not reach pregnancy. The non-pregnant subjects include $3,358$ cases where no embryos were implanted (due to embryo abnormalities) and $11,096$ cases where pregnancy did not occur despite implantation. As shown in Table 3.1, the subjects are characterized by some demographic information and a number of medical factors including vital signs, egg characteristics, sperm characteristics, fertility-related hormones, lifestyle, etc.

The variables in Table 3.1 are recorded sequentially in a typical IVF cycle [Massachusetts General Hospital Fertility Center, 2019,ESHRE Guideline Group on Good Practice in IVF Labs et al., 2016]. At the beginning of the first IVF cycle, some information is

collected from IVF subjects, including their demographics (age, race), vital signs (BMI), lifestyle (smoke, drink, exercise days per week), and diagnoses (infertility, tubal problems, genetic issues including chromosome anomaly or genetic defect, etc.) Then, some ovarian reserve tests are conducted on the third day of menstruation in female subjects to obtain laboratory-tested hormone levels: estradiol (E2), progesterone (P4), luteinizing hormone (LH), follicle stimulating hormone (FSH), endo thickness, antral follicle count (AFC). E2 levels are frequently evaluated to monitor follicular development, and variable "max E2" represents the peak E2 value during a spontaneous cycle. [Prasad et al., 2014,Delaney et al., 2012]

Next, transvaginal oocyte retrieval (TVOR) technology is used to remove oocytes from the female ovary for in vitro fertilization. The variable "oocytes retrieved" represents the number of retrieved oocytes. At this stage, a sperm sample is also collected from the male partner and the semen parameters are measured, including sperm volume, sperm concentration, sperm motility, and sperm progression. The next step is insemination (fertilization) by mixing eggs and sperm to produce embryos.

Embryo quality assessment is conducted including morphology characteristics, developmental stage and optional pre-implantation genetic diagnostics (PGD), based on which embryos are selected for transfer. The variable "total cryoed embryos" means the number of embryos frozen after fertilization, and the variable "day 5 available for freezing" represents embryos available to freeze five days after fertilization happens. Finally, at the three or five days after fertilization occurs, the selected embryos will be implanted into the female subjects' uterus. Some blood pregnancy tests will be scheduled around two weeks later to determine the final pregnancy result [Massachusetts General Hospital Fertility Center, 2019, ESHRE Guideline Group on Good Practice in IVF Labs et al., 2016].

Assume that all individuals are independent, we will build several reliable predictive models using a large amount of subjects, and analyze how different variables impact the

outcome.

Table 3.1: Various types of medical factors in the IVF data.

Ontology	Examples
Demographics	Age, Race
Vital Signs	Body Mass Index (BMI)
Egg & Embryo	Oocytes retrieved, Total embryos cryoed, Day 5 available for freezing
Sperm	Sperm volume, Sperm concentration, Sperm motility, Sperm progression
Lifestyle	Smoke, Drink, Exercise days per week
Laboratory Test Hormone Values	Day 3 ovarian test values: Estradiol (E2), Progesterone (P4), Luteinizing Hormone (LH), Follicle Stimulating Hormone (FSH), Endo Thickness, Antral follicle count (AFC); Max E2;
Diagnoses	Female infertility (42.38%); Male infertility/Oligospermia/Vasectomy (9.16%); Genetic issues (5.84%); Uterus problems (6.71%); Tubal problems (4.88%); Ovulation problems/PCOS, Ovarian failure/diminished reserve, Menstruation problems (17.43%); Obstetrics/Gynecology (Ob/gyn) infections, Sexually Transmitted Diseases (STD), Abortion, Pelvic Pain/adhesion, Hypothyroidism, Abdominal Pain/fatigue, Obesity and Others(18.82%);

3.1.1 Data Preprocessing

Data was collapsed in to categories. We cluster different diagnoses notes into 7 diagnostic groups; the details of which are shown in Table 3.1. The distribution of the subjects' demographics, vital signs and diagnoses are listed in Table 3.2.

First, we encode all categorical variables to generate numerical input values to feed into machine learning models. To that end, each possible value of a categorical variable is represented by a new variable that represents the occurrence of this value (binary variable). For instance, the "age" variable takes five possible values {<35, 35-37, 38-40, 41-42, 42+}, so there will be five distinct new indicator variables {age <35, age 35-37, age 38-40, age 41-42, age 42+}. All these new variables are binary variables that can only take values from {1, 0}.

Next, to improve accuracy of generalization to the population, we remove all variables that do not have enough population information (less than 0.1% of all records). For each

variable, we identify outliers that are more than 3 standard deviations away from the mean of non-missing variable values, and treat these outliers as missing values.

We also conduct data imputation on the remaining missing values. For continuous variables, the missing values are replaced with the median of non-missing values. For categorical variables, another variable is used to represent the occurrence of the missing values. To prevent multicollinearity, for highly correlated variables (with correlation ≥ 0.8), we only keep one of them because highly correlated variables usually provide redundant information. After such procedures there are 67 variables left as predictors.

Normalization is an important pre-processing stage before training predictive models. In this study, we adopt the Min-Max normalization [Jain and Bhandare, 2011, Liu et al., 2011] which linearly transform the variables to lie in the range $[0,1]$. The rescaling is achieved using the linear transformation given as: $\tilde{x}_i = (x_i \check{} \min(\mathbf{x}))/(\max(\mathbf{x}) \check{} \min(\mathbf{x}))$. The advantage of this normalization method is derived from the fact that it retains the original distribution of data except for a scaling factor and transforms all the values into a common numerical range $[0,1]$. This enables comparability of different variables and interpretability of variable importance.

3.2 Experimental Settings

3.2.1 Training and Test Sets

We randomly divide data into a training set (80%) and a test set (20%). The predictive models are trained only using the subjects' variables and labels in the training set. The prediction performance of trained models is evaluated on the test set. For a test sample, based on the subject' predictors, the trained predictive models provide a label, which can be compared with the ground truth.

Table 3.2: IVF subjects' demographics, vital signs and diagnoses of "Pregnant" and "Non-pregnant" subjects after the 1st IVF cycle. "Non-pregnant" means no embryos were implanted (due to embryo abnormalities) or pregnancy did not occur despite implantation.

Age in years	**Pregnant**	**Non-Pregnant**
<35	6333 (24.32%)	5439 (20.89%)
35-37	2638 (10.13%)	3095 (11.89%)
38-40	1837 (7.05%)	3035 (11.66%)
41-42	593 (2.28%)	1627 (6.25%)
42+	185 (0.71%)	1258 (4.83%)
Race	**Pregnant**	**Non-Pregnant**
White/Caucasian	2576 (9.89%)	2360 (9.06%)
Asian	461 (1.77%)	1926 (7.4%)
Hispanic/Latino	223 (0.86%)	311 (1.19%)
African	105 (0.4%)	173 (0.66%)
Unknown	8221 (31.57%)	9684 (37.19%)
BMI	**Pregnant**	**Non-Pregnant**
18.5 -24.9	4826 (18.53%)	6097 (23.41%)
25.5 -29.9	2341 (8.99%)	2748 (10.55%)
30.0 -34.9	1002 (3.85%)	1098 (4.22%)
35.0 -39.9	514 (1.97%)	606 (2.33%)
40+	380 (1.46%)	434 (1.67%)
<18.5	258 (0.99%)	497 (1.91%)
Diagnosis	**Pregnant**	**Non-Pregnant**
Female Infertility	1855 (7.12%)	2483 (9.54%)
Male Infertility/Oligospermia/Vasectomy	987 (3.79%)	970 (3.73%)
Uterus Problems	636 (2.44%)	702 (2.7%)
Genetic Issues	777 (2.98%)	734 (2.82%)
Tubal Problems	378 (1.45%)	462 (1.77%)
Ovulation problems/PCOS, Ovarian failure/diminished reserve, Menstruation problems	1301 (5%)	2326 (8.93%)
Ob/gyn infections, STD, Abortion, Pelvic Pain/adhesion, Hypothyroidism, Abdominal Pain/fatigue, Obesity and Others	2037 (7.82%)	2238 (8.59%)

3.2.2 Predictive Models

The classification models used to predict IVF success include linear models such as L1-regularized Logistic Regression (L1LR) and L2-regularized Logistic Regression (L2LR), linear SVM and nonlinear models such as Random Forest; all of these models have been described in Section 1.5. In addition, we use a Gradient-Boosted Decision Tree-based model to enhance the prediction power.

A Gradient-Boosted Decision Tree (GBDT) [Hastie et al., 2015, Friedman, 2001, Friedman, 2002] is an ensemble of multiple decision tree models; it creates a more powerful predictive model for classification and regression than a single base decision tree model. Unlike Random Forests which build distinct base decision tree models in parallel, GBDT builds a series of base decision trees, each of which is trained to correct the mistakes in the previous model in the series. In general, GBDT uses a large number of shallow trees (weak learners) built in a non-random manner to create models that make fewer errors as more trees are added. The learning rate controls the preferences of each new tree trying to correct the remaining mistakes from the previous round, hence a higher learning rate will result in more complex trees. Once the model is trained, only modest memory and runtime are needed to predict a new sample.

Extreme Gradient Boosting (XGBoost) [Chen and Guestrin, 2016] is a scalable system for learning boosted tree ensembles; it achieves performance improvements by introducing weighted approximate quantile sketch for approximate tree learning and a sparsity-aware algorithm for sparse datasets. With systems optimization techniques, such as parallelization, cache optimization, distributed computing, XGBoost is able to scale beyond billions of samples with much fewer resources compared to existing systems, hence, it has been widely adopted to achieve state-of-the-art performance on many data challenges.

Although ensemble classification methods like RF and GBDT are among the best off-the-shelf supervised learning methods that achieve excellent accuracy, they are too complex

for humans to interpret and have slow training phase, which requires exhaustive tuning of the hyper parameters such as the number of trees, the learning rate, etc.

3.3 Experimental Results

3.3.1 Pregnant Versus Non-pregnant

For an IVF subject, we first train models to predict whether the subject will become pregnant or not during their first IVF cycles with their own oocytes. All the classification models are trained and evaluated 10 times with different randomly generated training sets and test sets. Table 3.3 shows the mean and standard deviation of AUC and AUCPR from all classification methods over 10 runs.

Table 3.3: Average (avg) and standard deviation (std) of the Area Under ROC Curve (AUC) of predicting IVF pregnant versus non-pregnant.

Method	avg AUC	std AUC	avg AUCPR	std AUCPR
L1LR	0.6835	0.0047	0.5992	0.0064
L2LR	0.6837	0.0043	0.5998	0.0062
Linear SVM	0.6764	0.0039	0.5965	0.0059
RF	0.6997	0.0052	0.6191	0.0069
XGBoost	0.7095	0.0050	0.6290	0.0061

The interpretability of the results is critical to ensuring practical usage. We take the best L2LR model and examine the corresponding coefficients. The most important variables in the predictive model are ranked by the absolute values of the corresponding model coefficients, and listed in Table 3.4, where we only show variables with a p-value<0.05 and with an absolute coefficient no less than $\frac{1}{10} \times 1.67 = 0.167$, which is one-tenth of the absolute value of the largest absolute coefficient.

From Table 3.4, we can see that “age”-related, “egg”-related (i.e., Day 5 Available for freezing, Total Cryoed Embryos) variables and “Max E2” are among the most important variables. Some studies have shown that the pregnancy rate of embryos transferred on day 5

Table 3.4: Most significant variables for IVF pregnancy prediction.

Rank	Variables	Coefficient	95% CI	p-value
1	Age 42+	-1.67	[-1.87, -1.47]	0.000
2	Day 5 Available for freezing	1.46	[1.01, 1.91]	0.000
3	Race Asian	-1.13	[-1.26, -1.01]	0.000
4	Max E2	0.87	[0.28, 1.47]	0.004
5	Total Cryoed Embryos	0.84	[0.47, 1.22]	0.000
6	Oocytes Retrieved	0.82	[0.29, 1.36]	0.003
7	Age 41-42	-0.7	[-0.82, -0.57]	0.000
8	Diagnosis of Male Infertility/Oligospermia/Vasectomy	0.54	[0.3, 0.79]	0.001
9	Sperm Motility	0.36	[0.24, 0.48]	0.000
10	Age 38-40	-0.29	[-0.38, -0.2]	0.000
11	Race Caucasian	0.18	[0.1, 0.26]	0.000
12	Age <35	0.17	[0.1, 0.25]	0.000

or 6 after fertilization is higher than early or late implantation [Glujovsky et al., 2010]. Note that the variable "Race Asian" has a negative coefficient; this is consistent with existing literature indicating that Asian women are more likely to have infertility or lower clinical pregnancy rate [Wellons et al., 2008, Purcell et al., 2007, Jayaprakasan et al., 2014]. One explanation is that genetic and environmental exposure may vary by race [Shapiro et al., 2017,Fujimoto et al., 2010]. Some studies have hypothesized that Asians are more sensitive to gonadotropin stimulation than Caucasian [Palep-Singh et al., 2007] and that current IVF-related protocols are almost derived from North American or Western European studies and may not be equally applicable to all races [Niederberger et al., 2018]. It has been shown that E2 levels are significantly higher in pregnant women than in non-pregnant women [Prasad et al., 2014], and that IVF is very effective in treating male infertility [Schlegel and Girardi, 1997].

It is typically expensive to collect data through laboratory tests, hence classification models with parsimonious variables are preferred to support clinical decision-making. We

develop a simple classification model with only the most important five variables "Age 42+", "Day 5 Available for freezing", "Race Asian", "Max E2", and "Total Cryoed Embryos". Each variable is scaled individually to be in the range $[0,1]$, and we denote these normalized variables as x_1, x_2, x_3, x_4, x_5, respectively. The L2LR model provides high prediction accuracy with mean_AUC=0.6702, std_AUC=0.0055 and mean_AUCPR=0.5830, std_AUCPR=0.0061 over 10 random runs. A simple formula from the best L2LR model to calculate the probability of a patient to be classified as pregnant is $p = \frac{e^f}{1+e^f}$, where $f = -1.44x_1 + 2.33x_2 - 1.36x_3 + 1.54x_4 + 1.46x_5 - 0.53$; the corresponding model coefficients and 95% confidence intervals are listed in Table 3.5.

Table 3.5: Predictive IVF success rate model with only the 5 most important variables.

Variables	Coefficient	95% CI	p-value
Age 42+	-1.44	[-1.62, -1.26]	0.000
Day 5 Available for freezing	2.33	[1.97, 2.69]	0.000
Race Asian	-1.36	[-1.49, -1.24]	0.000
Max E2	1.54	[1.01, 2.06]	0.000
Total Cryoed Embryos	1.46	[1.11, 1.81]	0.000
Model Intercept	-0.53	[-0.60, -0.47]	0.000

Finally, in order to understand how various types of variables in Table 3.1 contribute to predicting IVF pregnancy outcomes, we apply all aforementioned classification models to each type of variables and report their performance in Table 3.6. We can see from the table that "Egg & Embryo"-related variables, "Laboratory Test Hormone Values", "Age" are among the most informative categories of variables.

In Table 3.6, the "Egg & Embryo"-type variables include Oocytes Retrieved, Total embryos cryoed, and Day 5 available for freezing. "Laboratory Test Hormone Values" represents variables "Max E2" and all Day 3 ovarian test values, such as Estradiol (E2), Progesterone (P4), Luteinizing Hormone (LH), Follicle Stimulating Hormone (FSH), Endo Thickness, Antral follicle count (AFC). "Diagnoses" represents diagnoses of female infertility,

Male infertility/Oligospermia/Vasectomy, genetic issues, uterus problems, tubal problems; ovulation problems/PCOS, ovarian failure/diminished reserve, menstruation problems; and Obstetrics/Gynecology (Ob/gyn) infections, Sexually Transmitted Diseases (STD), abortion, pelvic pain, pelvic adhesion, Hypothyroidism, Abdominal Pain/fatigue, Obesity and Others. "Sperm"-type variables include sperm volume, sperm concentration, sperm motility, sperm progression. "Lifestyle" represents smoking or not, drinking or not, exercise days per week.

Table 3.6: Performance of IVF outcome prediction with only one type of variables from Table 3.1.

Variables	avg AUC	std AUC	avg AUCPR	std AUCPR
Egg & Embryo	0.6443	0.0062	0.5652	0.0090
Laboratory Test Hormone Values	0.6192	0.0040	0.5405	0.0076
Age	0.6185	0.0059	0.2315	0.0029
Race	0.5670	0.0052	0.3862	0.0051
Diagnoses	0.5627	0.0043	0.4988	0.0082
Sperm	0.5541	0.0081	0.4818	0.0096
Lifestyle	0.5512	0.0068	0.4916	0.0075
BMI	0.5176	0.0063	0.4198	0.0069

3.3.2 Non-Pregnant Subjects: No Embryo Transferred Due to Abnormal Embryos Versus Not-Pregnant with Embryo Implantation

For subjects who are predicted to be non-pregnant during their first IVF cycles with their own oocytes according to the trained classification models, we further investigate whether no embryos were implanted (due to embryo abnormalities) or pregnancy did not occur despite implantation. For simplicity, we denote the former group as NoET-AE (no embryos transferred due to abnormal embryos) and the latter group as NP-ET (not pregnant with embryos transferred), respectively. All the classification models are trained and evaluated

10 times with different randomly generated training sets and test sets. Table 3.7 shows the mean and standard deviation of AUC and AUCPR from all classification methods over 10 runs.

Table 3.7: Predicting whether no embryos were implanted (due to embryo abnormalities) or pregnancy did not occur despite implantation.

Method	avg AUC	std AUC	avg AUCPR	std AUCPR
L1LR	0.8666	0.0055	0.7172	0.0127
L2LR	0.8667	0.0055	0.7173	0.0128
L2SVM	0.8647	0.0049	0.7012	0.0166
RF	0.8922	0.0052	0.7912	0.0138
XGBoost	0.9103	0.0043	0.8175	0.0111

Although NoET-AE and NP-ET are both non-pregnant groups, they may have different reasons leading to non-pregnancy. To understand this, we train two L2LR models to separate each non-pregnant group from the pregnant subjects, and assess the sensitivity of the models to specific predictive variables. Using the two trained L2LR models, we can extract two sets of top 10 most important variables that differentiate the two non-pregnant groups from pregnant subjects. The two sets of most important features share common features including “Age 42+”, “Max E2”, “Age 41-42”. The unique set of variables that separate NP-ET from pregnant are “Diagnoses of Tubal problems”, “Race African”; while the unique set of variables that contribute to NoET-AE are “Race Asian”, “Diagnoses of Ovulation problems/PCOS, Ovarian failure/diminished reserve, Menstruation problems ”, “Diagnoses of Ob/gyn infections, STD, Abortion, Pelvic pain/adhesion, Hypothyroidism, Abdominal Pain/fatigue, Obesity and Others”, “Race Hispanic/Latino”, “BMI $<$ 18.5”. Women who are underweight or obese are more likely to have a lower rate of implantation [Zhang et al., 2019]. Ob/gyn infection and STD have adverse effects on pregnant women. [Cotch et al., 1997]

3.4 Discussion and Conclusions

By examining a large IVF dataset with 26,040 subjects, we have designed several predictive models to predict IVF success rate during the first IVF cycle. The most important variables in the predictive model are listed to understand the prediction effect of predictors on the IVF outcome. We also propose a simple linear classification model with high accuracy that depends on only several significant variables, from which the formula to calculate the probability of a patient to be pregnant in the first IVF treatment cycle is derived. We conducted further analysis on the prediction impact of different types of variables on the first cycle IVF outcome. Finally, for predicted non-pregnant subjects, we were also able to predict whether no embryos were implanted (due to embryo abnormalities) or pregnancy did not occur despite implantation.

According to the World Health Organization (WHO), infertility is a reproductive system disease defined as the failure to reach a clinical pregnancy after unprotected intercourse for twelve months or longer [Zegers-Hochschild et al., 2009]. In the case that the woman is 35 or older then the time frame for assessing an infertility issue is 6 months [The Society for Assisted Reproductive Technology (SART), 2016]. Statistics shows that one out of seven couples struggle to have a baby on their own [The Society for Assisted Reproductive Technology (SART), 2016]. However with the advancement of assisted reproductive medicine, chances of conceiving with the appropriate help are greater than ever. It is worth mentioning that it has been more than 40 years since the first time ART was used to impregnate a woman [Wang and Sauer, 2006]. The CDC reports that 31% of ART cycles resulted in live births in women less than 35 years old [Centers for Disease Control and Prevention et al., 2018]. Yet, as commented before infertility can be caused by several factors such as: tubal factors (35%), ovulation factors (25%), male-related factors (40% sole contributor or contributing cause), and others falling under these categories [The Society for Assisted Reproductive Technology (SART), 2016]. Determining which ART procedure is best suited

for each couple is important so that their resources are maximized to help them conceive. This is why it is important to have access to an accurate and comprehensive protocol to predict how successful the first IVF cycle can be for a couple that uses their own eggs [Seli, 2011].

We provide predictive models to estimate the first cycle IVF success rates based on fertility-related variables such as age, eggs, sperm, hormones and lifestyle variables. Our models can accurately predict the IVF success rate and quantify the impact of the exposures. Our work achieves 70.95% AUC in predicting success following the first IVF cycle. Even with the most important five predictors, the predictive model can achieve an accuracy (AUC) of 67.02%. Based on the analysis of variable importance in the model, we found a number of most informative exposures consistent with the existing literature; for instance, "egg & embryo", "age" and "laboratory test hormone values" variables are among the most influential variables of the IVF success rate. [Broer et al., 2013, Xu et al., 2019] Furthermore, for predicted non-pregnant subjects, we predict whether no embryos were implanted (due to embryo abnormalities) or pregnancy did not occur despite implantation with an AUC of 91.03%.

Although our prediction model already contains many informative variables and achieves high accuracy, the model will provide more accuracy if there is more relevant information such as genetics or patient education. The advantage of the model is that it not only utilizes a large dataset to provide a high-accuracy IVF success rate prediction model, but also quantitatively measures the impact of exposures on the results, thus providing useful information for doctors and subjects to make decisions. The proposed simple classification model includes a pregnant probability calculation formula, which can provide doctors more reliable information to make better decisions. The predictive models can be easily adapted to other disease predictions (e.g. ovarian poor response) with more variables.

Chapter 4

Learning Parametric Policies and Transition Probability Models of MDPs From Data

We consider the problem of estimating the policy and transition probability model of a Markov Decision Process from data (state, action, next state tuples). The transition probability and policy are assumed to be parametric functions of a sparse set of features associated with the tuples. We propose two regularized maximum likelihood estimation algorithms for learning the transition probability model and policy, respectively. An upper bound is established on the regret, which is the difference between the average reward of the estimated policy under the estimated transition probabilities and that of the original unknown policy under the true (unknown) transition probabilities. We provide a sample complexity result showing that we can achieve a low regret with a relatively small amount of training samples. We illustrate the theoretical results with a healthcare example.

4.1 Related Work and Contributions

In healthcare applications, using data mining to predict future states of patients has been studied with a Markov chain model [Liu et al., 2013], deep learning [Choi et al., 2016], and Bayesian networks [Weiss et al., 2012]. However, these works are not efficient in learning policies. Although some recent works (including [Bertsimas et al., 2017, Xu and Paschalidis, 2019, Chen and Paschalidis, 2019]) develop methods to learn actions from data, they formulate the problem as a static classification problem and do not account for the sequential nature of these actions and their impact on future states. There is substantial work on

learning MDP policies by observing experts' demonstrations. A survey on robot learning from demonstrations is studied in [Argall et al., 2009]. The work in [Paternain et al., 2018] learns policies for MDPs in continuous spaces, and [Yu et al., 2016] conducts strategy learning for non-task-oriented conversational systems. [Wen et al., 2017] learns the MDP policy from demonstrations where extra side information on task requirements is available. Other works consider the problem of learning transition probabilities. In [Geller et al., 2019], an algorithm for learning the health state transition matrix via wireless body area networks modeled as POMDPs is proposed. Another work [Kent et al., 2018] learns the ransition functions and obtains the optimal policy through value iteration with respect to the learned transition functions. However, these works only validate the effeciency of proposed algorithms through experiments without providing theoretical guarantee. [Yakowitz et al., 1979] provides a non-parametric estimation of Markov transition functions with a convergence guarantee. [Györfi and Kohler, 2007] utilizes Poisson regression to estimate general conditional distributions. The articles [Iyengar, 2005, Nilim and Ghaoui, 2005, Wiesemann et al., 2013] aim to obtain an optimal policy to alleviate the sensitivity to uncertainty in underlying transition probabilities. These methods are not tractable in many DP applications, since it can take overwhelming computational effort to compute an optimal solution according to Bellman's curse of dimensionality.

More closely related to this work, [Hanawal et al., 2018] investigates the question of learning an MDP policy from data, where the explicit MDP model is known. Here, we consider a more general situation where the MDP model is unknown, therefore, both the policy and transition probabilities need to be estimated from data.

To address the problem of estimating both the agent's policy and conditional transition probabilities, we propose two learning models and an estimation algorithm based on regularized logistic regression. The use of regularization is motivated by a recent body of work (see [Abadeh et al., 2015, Chen and Paschalidis, 2018, Chen and Paschalidis, 2019] and ref-

erences therein) which establish that to render the estimates robust to outliers in the training data one needs to solve properly regularized empirical loss minimization problems.

We characterize the sample complexity of estimating model parameters. We also derive a bound on the regret, which indicates the difference between the average reward of the estimated policy under the estimated transition probabilities and the average reward of the original policy under the true transition dynamics. Compared to the earlier work [Hanawal et al., 2018], this work proposes a learning algorithm that has a solid average reward guarantee even in the case where the MDP dynamics are unknown. A conference version of this work has appeared in [Zhu et al., 2019].

We organize the remaining parts of this work as follows. In Section 4.2, we introduce the MDP model and the learning problem formulation. In Section 4.3, we present the proposed learning algorithm and the corresponding performance metrics. In Section 4.4, we establish the algorithm's performance on log loss or the distance of the estimated parameters from the original ones. In Section 4.5, we bound the regret of the estimated policy under the estimated MDP dynamics. In Section 4.6, we illustrate the theoretical results using two simulation experiments. Conclusions are in Section 4.7.

4.2 Problem Formulation

Consider a finite-state Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathbf{P}, R, \mu)$, where $\mathcal{S}$, $\mathcal{A}$ are the sets of possible states and actions, respectively. For any (state, action, next state) tuple (s, a, q), denote by $P(q|s,a)$ the transition probability from state s to state q conditional on taking action a. We will use $\mathbf{P}(\cdot|s,a)$ to denote the transition probability vector at state s under action a. The function R represents the one-step reward of the MDP. The function μ is a policy that maps a state to a probability distribution of actions; specifically, $\mu(a|s)$ represents the probability of adopting action a given state s.

For general MDPs with a large state-action space, we use a class of parametric

(Boltzmann-type) functions to approximate the policy and conditional transition probabilities:

$$P_\xi(q|s,a) = \frac{\exp\{\xi'\psi(s,a,q)\}}{\sum_{y\in\mathcal{N}_s}\exp\{\xi'\psi(s,a,y)\}}, \tag{4.1}$$

$$\mu_\theta(a|s) = \frac{\exp\{\theta'\phi(s,a)\}}{\sum_{b\in\mathcal{A}}\exp\{\theta'\phi(s,b)\}}, \tag{4.2}$$

where $\xi \in \mathbb{R}^N$ and $\theta \in \mathbb{R}^n$ are parameters to be learned and $\mathcal{N}_s$ is the set of all feasible next states from the current state s. The kernel functions $\psi : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]^N$ and $\phi : \mathcal{S}\times\mathcal{A} \to [0,1]^n$ transform (state, action, next state) tuples (s,a,q) and (state, action) pairs (s,a) into corresponding desired features, respectively. For the sake of brevity, we will use the term transition probabilities ξ to refer to the conditional transition probabilities induced by the model (4.1) with parameter ξ, and, similarly, we will use the term policy θ to refer to the policy defined in (4.2) induced by parameter θ.

We note that the models in (4.1) and (4.2) use a given set of feature vector functions ψ and ϕ that encode important aspects of the raw states and actions. Such features may be designed using intuition about a particular application or obtained more systematically using kernel function ideas (see [Hanawal et al., 2018, Sec. IV] and the discussion in Section 4.6.2).

Given an MDP driven by policy θ with conditional transition probabilities $\mathbf{P}_\xi$, the MDP states form a Markov chain. We use $\mathbf{M}_{\xi,\theta}$ to represent its transition matrix, where $M_{\xi,\theta}(q|s) = \sum_{a\in\mathcal{A}}\mu_\theta(a|s)P_\xi(q|s,a)$ for all (state, next state) pairs (s,q). Both the policy and conditional transition probabilities are Boltzmann-type, hence, there is a unique stationary distribution $\pi_{\xi,\theta}(s)$ of the induced Markov chain. This implies a unique stationary distribution of state-action pairs (s,a), denoted by $\eta_{\xi,\theta}(s,a) = \pi_{\xi,\theta}(s)\mu_\theta(a|s)$, based on which we can define the infinite horizon average reward induced by ξ,θ as

$$\overline{R}(\xi,\theta) = \sum_{(a,s)}\eta_{\xi,\theta}(s,a)R(s,a).$$

4.3 Estimating the Policy and Transition Probabilities

Given an observed collection of states and corresponding actions by an agent, we aim to learn the agent's policy and the corresponding transition probabilities, denoted by θ^* and ξ^*, respectively.

The set of (state, action, next state) tuples driven by policy θ^* with conditional transition probabilities ξ^* is denoted by $\mathcal{X} := \mathcal{X}(\xi^*, \theta^*) = \{(s_i, a_i, q_i);\ i = 1, \ldots, m\}$. Assume that the state-action pairs $\{(s_i, a_i);\ i = 1, \ldots, m\}$ are i.i.d. and sampled from the stationary distribution $\eta_{\xi,\theta}$. Given state s_i and action a_i, the next state q_i is selected based on the conditional transition probability ξ^*. Therefore, the tuples $\{(s_i, a_i, q_i)\}$ in the data set $\mathcal{X}$ are i.i.d and follow the distribution $\mathcal{D} \sim P_{\xi^*}(q|s,a)\eta_{\xi^*,\theta^*}(s,a)$. Since usually only a small subset of features are significant in learning appropriate models, we assume θ^* and ξ^* are sparse, having $r < n$ and $q < N$ non-zero elements, respectively. Also, assume that θ^* and ξ^* are bounded by K, element-wise.

Due to the probabilistic nature of (4.1) and (4.2), we can adopt *Maximum Likelihood (ML)* estimation with logistic regression to estimate the parameters from data. Given our sparsity assumption on the models θ^* and ξ^*, the distance between two sample points in the feature-label space (ϕ, a) and (ψ, q), respectively, is best measured using the sparse ℓ_∞ norm which accounts for the most dominant feature instead of the entire (and potentially irrelevant) feature vector. This suggests [Chen and Paschalidis, 2018], that one needs to use the dual norm, i.e., an ℓ_1 norm, to regularize the ML estimation problem. We will introduce this regularization as a constraint in the formulation. Specifically, we formulate the learning problem of conditional transition probabilities as follows:

$$\begin{aligned} &\max_{\xi \in \Re^N} \sum_{i=1}^{m} \log P_\xi(q_i|s_i, a_i) \\ &\text{s.t.} \quad \|\xi\|_1 \leq B_\xi, \end{aligned} \tag{4.3}$$

in which B_ξ adjusts the sparsity of the estimated transition probabilities. Similarly, we formulate the policy learning as follows:

$$\max_{\theta \in \mathcal{R}^n} \sum_{i=1}^{m} \log \mu_\theta(a_i|s_i) \tag{4.4}$$
$$\text{s.t} \quad \|\theta\|_1 \leq B_\theta,$$

in which B_θ adjusts the sparsity of the estimated policy.

The performance of the ML estimate can be evaluated through a log-loss metric, which is defined as the expected value of the negative log-likelihood over the sample distribution. Specifically, for any parameters ξ and θ, the log loss of the transition probability model and the policy are defined as follows:

$$\varepsilon(\xi) = \mathbb{E}_{(s,a,q)\sim\mathcal{D}}[-\log P_\xi(q|s,a)],$$

$$\zeta(\theta) = \mathbb{E}_{(s,a,q)\sim\mathcal{D}}[-\log \mu_\theta(a|s)].$$

The expectation is taken over the distribution $\mathcal{D}$. Since the explicit distribution is unknown, we can only observe (state, action, next state) tuples. Given any data set $\mathcal{X}$, an empirical version of log-losses can be defined as follows:

$$\hat{\varepsilon}_\mathcal{X}(\xi) = \frac{1}{m}\sum_{i=1}^{m}(-\log P_\xi(q_i|s_i,a_i)),$$

$$\hat{\zeta}_\mathcal{X}(\theta) = \frac{1}{m}\sum_{i=1}^{m}(-\log \mu_\theta(a_i|s_i)).$$

The empirical log-losses are calculated by the log losses over the training data set $\mathcal{X}(\xi^*, \theta^*)$ and denoted by

$$\hat{\varepsilon}(\xi) \triangleq \hat{\varepsilon}_{\mathcal{X}(\xi^*,\theta^*)}(\xi), \quad \hat{\zeta}(\theta) \triangleq \hat{\zeta}_{\mathcal{X}(\xi^*,\theta^*)}(\theta).$$

We summarize our algorithms to learn the policy θ^* and transition probability model ξ^* in Algorithm 4.

Algorithm 4 Training algorithm to estimate the policy θ^* (or transition probabilities ξ^*) from samples $\mathcal{X}$.

Initialization: Fix hyper-parameters $C > rK$ ($C > lK$) and $0 < \gamma < 1$.
Randomly split the whole data set $\mathcal{X}$ into two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ with size γm and $(1-\gamma)m$ respectively. Use $\mathcal{X}_1$ and $\mathcal{X}_2$ as training set and cross-validation set, respectively.
Training phase:
for $B = 0, 1, 2, \ldots, C$ **do**
solve optimization problem (4.4) (or (4.3)) with the training set $\mathcal{X}_1$ and right hand side of the constraint equal to B. Let θ_B (or ξ_B) denote the obtained optimal solution.
end for
Validation phase: Among the obtained solutions θ_B's (or ξ_B's) from the training phase, select the best solution with the lowest "hold-out" error on the validation set $\mathcal{X}_2$, i.e, $\hat{B} = \arg\min_{B \in \{0,1,\ldots,C\}} \hat{\zeta}_{\mathcal{X}_2}(\theta_B)$ and set $\hat{\theta} = \theta_{\hat{B}}$ (or $\hat{B} = \arg\min_{B \in \{0,1,\ldots,C\}} \hat{\varepsilon}_{\mathcal{X}_2}(\xi_B)$ and set $\hat{\xi} = \xi_{\hat{B}}$) , where $\hat{\zeta}_{\mathcal{X}_2}(\cdot)$ (or $\hat{\varepsilon}_{\mathcal{X}_2}(\cdot)$) denotes the empirical log loss of the policy function (transition probabilities) on the validation set $\mathcal{X}_2$.

4.4 Log-Loss Generalization Guarantees

In this section, we will establish theoretical results on sample complexity, showing that our estimation algorithm (Algorithm 4) is guaranteed to achieve high accuracy with relatively few training samples.

We first relate the difference between log losses of two conditional transition probabilities vectors (policies) to their relative entropy, or Kullback-Leibler (KL) divergence, which characterizes the difference between two distributions. The KL divergence of two conditional transition probability vectors ξ_1 and ξ_2 at (s,a) is defined as:

$$\begin{aligned} &D(\mathbf{P}_{\xi_1}(\cdot|s,a) \| \mathbf{P}_{\xi_2}(\cdot|s,a)) \\ &= \sum_q P_{\xi_1}(q|s,a) \log \frac{P_{\xi_1}(q|x,a)}{P_{\xi_2}(q|s,a)}. \end{aligned}$$

Similarly, the KL divergence between policies θ_1 and θ_2 at state s is

$$D(\mu_{\theta_1}(\cdot|s) \| \mu_{\theta_2}(\cdot|s)) = \sum_a \mu_{\theta_1}(a|s) \log \frac{\mu_{\theta_1}(a|s)}{\mu_{\theta_2}(a|s)}.$$

Next, we will define the average KL divergence of the policy and transition probability

model under some sample distribution. Recall that the stationary distributions of the state and state-action Markov chains are denoted by $\pi_{\xi,\theta}$ and $\eta_{\xi,\theta}$, respectively. The average KL divergence between policies θ_1 and θ_2 is defined as follows:

$$D_{\xi,\theta}(\mu_{\theta_1}\|\mu_{\theta_2}) = \sum_s \pi_{\xi,\theta}(s) D(\mu_{\theta_1}(\cdot|s)\|\mu_{\theta_2}(\cdot|s)),$$

and the average KL divergence between conditional transition probabilities ξ_1 and ξ_2 is defined as

$$D_{\xi,\theta}(\mathbf{P}_{\xi_1}\|\mathbf{P}_{\xi_2}) = \sum_{s,a} \eta_{\xi,\theta}(s,a) D\left(\mathbf{P}_{\xi_1}(\cdot|s,a)\|\mathbf{P}_{\xi_2}(\cdot|s,a)\right).$$

According to [Hanawal et al., 2018], the difference between log losses of two policies is equal to their Kullback-Leibler (KL) divergence.

Lemma 4.4.1 ([Hanawal et al., 2018])**.** *Let $\hat{\theta}$ be an estimate of the policy θ, then*

$$\zeta(\hat{\theta}) - \zeta(\theta) = D_{\xi,\theta}(\mu_\theta\|\mu_{\hat{\theta}}).$$

Similarly, we can prove that the difference between log losses of two transition probability models are equal to their KL divergence. The proof is similar to that for Lemma 4.4.1 and hence omitted.

Corollary 4.4.2. *Let $\hat{\xi}$ be an estimate of the policy ξ, then*

$$\varepsilon(\hat{\xi}) - \varepsilon(\xi) = D_{\xi,\theta}(\mathbf{P}_\xi\|\mathbf{P}_{\hat{\xi}}).$$

In [Hanawal et al., 2018], we have obtained a bound of the difference between log losses of the policy μ_{θ^*} and the estimated policy $\mu_{\hat{\theta}}$ as follows.

Theorem 4.4.3 ([Hanawal et al., 2018])**.** *Given any positive values $\varepsilon > 0$ and $\delta > 0$, if the sample size satisfies the following condition*

$$m = \Omega\Big((\log n) \cdot poly(r, K, C, H, \log(1/\delta), 1/\varepsilon)\Big),$$

where H is the maximum number of feasible actions at a state of the MDP, then with prob-

ability at least $1-\delta$*,*

$$|\zeta(\hat{\theta}) - \zeta(\theta)| = D_{\xi^*,\theta^*}(\mu_{\theta^*} \| \mu_{\hat{\theta}}) \leq \varepsilon.$$

This implies that $\hat{\theta}$ obtained by Algorithm 4 can be arbitrarily close to the unknown policy θ^*. The function $\text{poly}(s)$ denotes a polynomial function with respect to elements of s. Specifically, $m = \Omega(H^3)$ in terms of only H.

Similarly, in the following corollary, we bound the difference between the log losses of the original and estimated conditional transition probability, $\mathbf{P}_{\xi^*}$ and $\mathbf{P}_{\hat{\xi}}$.

Corollary 4.4.4. *Given any positive values* $\varepsilon > 0$ *and* $\delta > 0$*, if the sample size satisfies the following condition*

$$m = \Omega\Big((\log N) \cdot poly(l, K, C, M, \log(1/\delta), 1/\varepsilon)\Big),$$

where $M = \max_s |\mathcal{N}_s|$*, then with probability at least* $1-\delta$*,*

$$|\varepsilon(\hat{\xi}) - \varepsilon(\xi)| = D_{\xi^*,\theta^*}(\mathbf{P}_{\xi^*} \| \mathbf{P}_{\hat{\xi}}) \leq \varepsilon.$$

This implies that $\hat{\xi}$ obtained by Algorithm 4 can be arbitrarily close to the original transition probability ξ^*, with $m = \Omega(M^3)$ in terms of only M. The proof is similar to that for Theorem 4.4.3 and hence omitted.

4.5 Bounds on Regret

With the estimated conditional transition probabilities from Section 4.3, we are able to compute the average reward of the estimated policy by simulating the MDP. A natural question is how good this simulated average reward is, compared to the average reward of the original policy under the true MDP dynamics.

In this section, we develop a bound on the regret using the estimated policy as opposed to the original policy. Specifically, we define the regret $\text{Reg}(\mathcal{X})$ as

$$\text{Reg}(\mathcal{X}) = \overline{R}(\xi^*, \theta^*) - \overline{R}(\hat{\xi}, \hat{\theta}),$$

in which $\hat{\theta}$ and $\hat{\xi}$ are the estimated policy and estimated conditional transition probabilities from the samples $\mathcal{X}$, respectively. We need the following preliminary definitions and lemmata.

Definition 1 ([Hanawal et al., 2018]). *Given a Markov chain with transition probability matrix $\mathbf{M}_{\xi,\theta}$ induced by policy θ and conditional transition probability parameter ξ, the fundamental matrix of the Markov chain is defined as*

$$\mathbf{Z}_{\xi,\theta} = (\mathbf{A}_{\xi,\theta} + \mathbf{e}\pi'_{\xi,\theta})^{-1},$$

where $\mathbf{A}_{\xi,\theta} = \mathbf{I} - \mathbf{M}_{\xi,\theta}$, $\pi_{\xi,\theta}$ represents the stationary distribution associated with $\mathbf{M}_{\xi,\theta}$, and $\mathbf{e}$ is the vector with all elements equal to 1.

Definition 2 ([Hanawal et al., 2018]). *The group inverse of a square matrix $\mathbf{A}$, denoted as $\mathbf{A}^{\#}$, is the unique square matrix satisfying the following conditions*

$$\mathbf{A}\mathbf{A}^{\#}\mathbf{A} = \mathbf{A},\ \mathbf{A}^{\#}\mathbf{A}\mathbf{A}^{\#} = \mathbf{A},\ \mathbf{A}\mathbf{A}^{\#} = \mathbf{A}^{\#}\mathbf{A}.$$

Definition 3 ([Hanawal et al., 2018]). *Given a matrix $\mathbf{B}$ with equal row sums, its ergodic coefficient is defined as*

$$\tau(\mathbf{B}) = \sup_{\mathbf{v}'\mathbf{e}=0;\|\mathbf{v}\|_1=1} \|\mathbf{v}'\mathbf{B}\|_1 = \frac{1}{2}\max_{i,j}\sum_s |b_{is} - b_{js}|. \tag{4.5}$$

Lemma 4.5.1. *([Cover and Thomas, 2006, Lemma 11.6.1]) Consider any two probability vectors $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{R}^n$. It holds that*

$$D(\mathbf{p}_1\|\mathbf{p}_2) \geq \frac{1}{2\ln 2}\|\mathbf{p}_1 - \mathbf{p}_2\|_1^2. \tag{4.6}$$

Lemma 4.5.2 ([Hanawal et al., 2018]). *Given two stochastic matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ (and associated fundamental matrices $\mathbf{Z}_i$ and matrices $\mathbf{A}_i = \mathbf{I} - \mathbf{P}_i$, $i = 1,2$), assume π_1 and π_2 are the corresponding unique stationary distributions, respectively. Let $\mathbf{E} = \mathbf{P}_1 - \mathbf{P}_2$, then*

$$\|\pi_1 - \pi_2\|_1 \leq \kappa\|\pi'_1\mathbf{E}\|_1, \tag{4.7}$$

where κ is a constant that can take one of the following values: $\kappa = \|\mathbf{Z}_2\|_\infty$, or $\kappa = \|\mathbf{A}_2^{\#}\|_\infty$, or $\kappa = 1/(1-\tau(\mathbf{P}_2))$, or $\kappa = \tau(\mathbf{Z}_2) = \tau(\mathbf{A}_2^{\#})$.

Now we are ready to bound the regret of the estimated policy using the above preliminary definitions and results.

Theorem 4.5.3. *Given any positive values* $\varepsilon > 0$, $\delta > 0$, *assume Algorithm 4 adopts*

$$\Omega((\log(\max(N,n))) \cdot poly(r,l,K,C,M,\log(1/\delta),\, 1/\varepsilon, H)$$

i.i.d. training samples to learn parameters $\hat{\xi}$ *and* $\hat{\theta}$. *Then, with probability of at least* $1-\delta$, *the following result holds*

$$|\overline{R}(\xi^*,\theta^*) - \overline{R}(\hat{\xi},\hat{\theta})| \leq 2\sqrt{\ln 2\varepsilon} R_{\max}(1+2\kappa),$$

in which $R_{\max} = \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} |R(s,a)|$ *and* κ *is a constant value that depends on the transition probability matrix* $\mathbf{M}_{\hat{\xi},\hat{\theta}}$ *(and its corresponding fundamental matrix* $\mathbf{Z}_{\hat{\xi},\hat{\theta}}$ *and matrix* $\mathbf{A}_{\hat{\xi},\hat{\theta}} = \mathbf{I} - \mathbf{M}_{\hat{\xi},\hat{\theta}}$*). This constant* κ *can be any of the following:* $\kappa = \|\mathbf{Z}_{\hat{\xi},\hat{\theta}}\|_\infty$, *or* $\kappa = \|\mathbf{A}^{\#}_{\hat{\xi},\hat{\theta}}\|_\infty$, *or* $\kappa = 1/(1-\tau(\mathbf{M}_{\hat{\xi},\hat{\theta}}))$, *or* $\kappa = \tau(\mathbf{Z}_{\hat{\xi},\hat{\theta}}) = \tau(\mathbf{A}^{\#}_{\hat{\xi},\hat{\theta}})$.

Proof. We will first express the regret as the sum of two parts, and then bound each part separately.

$$\begin{aligned}
&\mathrm{Reg}(\mathcal{X}) = \overline{R}(\xi^*,\theta^*) - \overline{R}(\hat{\xi},\hat{\theta}) \\
=&\sum_s \sum_a [\eta_{\xi^*,\theta^*}(s,a) - \eta_{\hat{\xi},\hat{\theta}}(s,a)] R(s,a) \\
=&\sum_s \pi_{\xi^*,\theta^*}(s) \sum_a \mu_{\theta^*}(a|s) R(s,a) \\
&- \sum_s \pi_{\hat{\xi},\hat{\theta}}(s) \sum_a \mu_{\hat{\theta}}(a|s) R(s,a) \\
=&\sum_s \pi_{\xi^*,\theta^*}(s) \sum_a [\mu_{\theta^*}(a|s) - \mu_{\hat{\theta}}(a|s)] R(s,a) \\
&- \sum_s [\pi_{\hat{\xi},\hat{\theta}}(s) - \pi_{\xi^*,\theta^*}(s)] \sum_a \mu_{\hat{\theta}}(a|s) R(s,a) \\
\leq&\left| \sum_s \pi_{\xi^*,\theta^*}(s) \sum_a [\mu_{\theta^*}(a|s) - \mu_{\hat{\theta}}(a|s)] R(s,a) \right| \\
+&\left| \sum_s [\pi_{\hat{\xi},\hat{\theta}}(s) - \pi_{\xi^*,\theta^*}(s)] \sum_a \mu_{\hat{\theta}}(a|s) R(s,a) \right|. \qquad (4.8)
\end{aligned}$$

Note that the first term in equation (4.8) depends on the value $\sum_a [\mu_{\theta^*}(a|s) - \mu_{\hat{\theta}}(a|s)]$, which is the learning error in estimating the policy. The second term depends on $\sum_s |\pi_{\hat{\xi},\hat{\theta}}(s) - \pi_{\xi^*,\theta^*}(s)|$, which is the perturbation error of the stationary distribution of the Markov chain

by applying the estimated policy $\hat{\theta}$. In the following, we will bound the two terms separately, and begin with the first term.

$$\begin{aligned}
&\left|\sum_s \pi_{\xi^*,\theta^*}(s)\sum_a [\mu_{\theta^*}(a|s)-\mu_{\hat{\theta}}(a|s)]R(s,a)\right| \\
\leq &\sum_s \pi_{\xi^*,\theta^*}(s)\sum_a |(\mu_{\theta^*}(a|s)-\mu_{\hat{\theta}}(a|s))|\cdot|R(s,a)| \\
\leq &R_{\max}\sum_s \pi_{\xi^*,\theta^*}(s)\sum_a |(\mu_{\theta^*}(a|s)-\mu_{\hat{\theta}}(a|s))| \\
= &R_{\max}\sum_s \pi_{\xi^*,\theta^*}(s)\|(\mu_{\theta^*}(\cdot|s)-\mu_{\hat{\theta}}(\cdot|s))\|_1. \qquad (4.9)
\end{aligned}$$

The bound in (4.9) depends on the difference between the log-loss of the estimated policy $\hat{\theta}$ and that of target policy θ^*. Based on Lemma 4.5.1, we have

$$\begin{aligned}
&\left|\sum_s \pi_{\xi^*,\theta^*}(s)\sum_a [\mu_{\theta^*}(a|s)-\mu_{\hat{\theta}}(a|s)]R(s,a)\right| \\
\leq &\ R_{\max}\sum_s \pi_{\xi^*,\theta^*}(s)\sqrt{2(\ln 2)D\left(\mu_{\theta^*}(\cdot|s)\|\mu_{\hat{\theta}}(\cdot|s)\right)} \\
\leq &\ \sqrt{2\ln 2}R_{\max}\cdot \\
&\ \sqrt{\sum_s \pi_{\xi^*,\theta^*}(s)D\left(\mu_{\theta^*}(\cdot|s)\|\mu_{\hat{\theta}}(\cdot|s)\right)} \\
= &\ \sqrt{2\ln 2}R_{\max}\sqrt{D_{\xi^*,\theta^*}(\mu_{\theta^*}\|\mu_{\hat{\theta}})} \\
\leq &\ \sqrt{2(\ln 2)\varepsilon}R_{\max}. \qquad (4.10)
\end{aligned}$$

The first inequality holds by applying Lemma 4.5.1 with $\mathbf{p}_1 = \mu_{\theta^*}(\cdot|s)$ and $\mathbf{p}_2 = \mu_{\hat{\theta}}(\cdot|s)$ for each state s. The second inequality is obtained by applying Jensen's inequality. The final inequality holds using Theorem 4.4.3.

Next, we will bound the second term in (4.8).

$$\begin{aligned}
&\left|\sum_s (\pi_{\hat{\xi},\hat{\theta}}(s)-\pi_{\xi^*,\theta^*}(s))\sum_a \mu_{\hat{\theta}}(a|s)R(s,a)\right| \\
\leq &\ \sum_s |\pi_{\hat{\xi},\hat{\theta}}(s)-\pi_{\xi^*,\theta^*}(s)|\sum_a |\mu_{\hat{\theta}}(a|s)R(s,a)| \\
\leq &\ R_{\max}\sum_s |\pi_{\hat{\xi},\hat{\theta}}(s)-\pi_{\xi^*,\theta^*}(s)|\sum_a \mu_{\hat{\theta}}(a|s) \\
= &\ R_{\max}\sum_s |\pi_{\hat{\xi},\hat{\theta}}(s)-\pi_{\xi^*,\theta^*}(s)| \qquad (4.11) \\
\leq &\ R_{\max}\kappa\|\pi'_{\xi^*,\theta^*}(\mathbf{M}_{\xi^*,\theta^*}-\mathbf{M}_{\hat{\xi},\hat{\theta}})\|_1. \qquad (4.12)
\end{aligned}$$

The equality (4.11) can be obtained since $\sum_a \mu_{\hat{\theta}}(a|s) = 1$ for all s, and (4.12) holds by using Lemma 4.5.2.

For the last term in equation (4.12), by applying the definition of the transition probability $\mathbf{M}_{\xi,\theta}$ associated with parameters ξ and θ, we obtain

$$\begin{aligned}
&\|\pi'_{\xi^*,\theta^*}(\mathbf{M}_{\xi^*,\theta^*} - \mathbf{M}_{\hat{\xi},\hat{\theta}})\|_1 \\
&= \sum_q \Big| \sum_s \pi_{\xi^*,\theta^*}(s) \cdot \\
&\qquad \sum_a [P_{\xi^*}(q|s,a)\mu_{\theta^*}(a|s) - P_{\hat{\xi}}(q|s,a)\mu_{\hat{\theta}}(a|s)] \Big| \\
&\leq \sum_s \pi_{\xi^*,\theta^*}(s) \sum_q \sum_a \Big| P_{\hat{\xi}}(q|s,a) \left[\mu_{\theta^*}(a|s) - \mu_{\hat{\theta}}(a|s)\right] \Big| \\
&+ \sum_s \sum_a \pi_{\xi^*,\theta^*}(s)\mu_{\theta^*}(a|s) \sum_q \Big| P_{\hat{\xi}}(q|s,a) - P_{\xi^*}(q|s,a) \Big| \\
&\leq \sum_s \pi_{\xi^*,\theta^*}(s) \sum_a \left|\mu_{\theta^*}(a|s) - \mu_{\hat{\theta}}(a|s)\right| \\
&+ \sum_{s,a} \pi_{\xi^*,\theta^*}(s,a) \sum_q \Big| P_{\hat{\xi}}(q|s,a) - P_{\xi^*}(q|s,a) \Big|,
\end{aligned}$$

in which the last inequality holds by using $\sum_q P(q|s,a) = 1$ for all (s,a). Now, using inequality (4.10), Theorem 4.4.3 and Corollary 4.4.4, we are able to bound (4.9) as

$$\begin{aligned}
&\left| \sum_s (\pi_{\hat{\xi},\hat{\theta}}(s) - \pi_{\xi^*,\theta^*}(s)) \sum_a \mu_{\hat{\theta}}(a|s) R(s,a) \right| \\
\leq & 2\sqrt{2\varepsilon \ln 2}\kappa R_{\max}. \qquad (4.13)
\end{aligned}$$

Finally, by combining (4.10), (4.13) and (4.8), we obtain the result of Theorem 4.5.3. □

Note that in Theorem 4.5.3, the regret relies on the constant value κ, which is referred to as the condition number of the estimated policy under the estimated transition dynamics. The regret monotonically increases with the condition number.

4.6 A Disease Progression Example

In this section, we illustrate the efficiency of our algorithms using a disease progression experiment where we seek to learn the prescription policy reflected in the data.

Table 4.1: Disease state transition probabilities conditioned on actions.

State Diff $\mathbf{z}_t =$	$(0,0)$	$(0,1)$	$(0,-1)$	$(-1,0)$	$(1,0)$
$\mathbf{a}_t = (0,0)$	0.7	0.1	0.05	0.05	0.1
$\mathbf{a}_t = (1,0)$	0.4	0.1	0.05	0.35	0.1
$\mathbf{a}_t = (0,1)$	0.4	0.1	0.35	0.05	0.1
$\mathbf{a}_t = (1,1)$	0.2	0.1	0.3	0.3	0.1

4.6.1 Experimental Settings

We consider patients with a chronic disease and design an MDP to model the effect of drugs. We denote the state of the MDP by $\mathbf{s} = (s_1, s_2)$, where s_1, $s_2 \in \{0, 1, \ldots, 20\}$. Here, s_1 denotes the severity of the disease itself, and s_2 the severity of the comorbidities that the patient may face. The actions in the MDP represent the various treatments for the patient. Assume there are two drug types with different treatment effects, one for the main symptoms of the disease, and another for the comorbidities. The actions can be denoted by $\mathbf{a} = (a_1, a_2)$, where $a_i \in \{0, 1\}$ is a indicator of whether the patient takes the i-th type of drug or not.

We assume that the disease state can only transition from the current state to neighboring states; specifically, the state difference $\mathbf{z}_t = \mathbf{s}_{t+1} - \mathbf{s}_t$ at time t should satisfy $\|\mathbf{z}_t\|_1 \leq 1$ for all $\mathbf{s}_t$, $\mathbf{s}_{t+1}$ and $\mathbf{a}_t$. Furthermore, we assume that the transition probability $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ depends only on the state difference $\mathbf{z}_t = \mathbf{s}_{t+1} - \mathbf{s}_t$ and action $\mathbf{a}_t$. The transition probability matrix (conditioned on all actions) is shown in Table 4.1. The MDP is also assumed to have a bouncing boundary, i.e., when the patient takes an action leading the state outside of the boundary, the transition is bounced back without changing the current state.

Consistent with Table 4.1, if the patient takes none of the two treatments, i.e., $\mathbf{a}_t = (0,0)$, both disease and comorbidity indicators s_1 and s_2 will tend to stay the same or become more severe. The disease severity decreases if the patient uses the type-1 drug, i.e., $\mathbf{a}_t = (1,0)$. The comorbidity symptoms are relieved when the patient takes type-2

drug, i.e., $\mathbf{a}_t = (0,1)$. Because of the drug interactions, if both drugs are used together, i.e., $\mathbf{a}_t = (1,1)$, they have diminished effects compared to being individually used.

When a patient enters a disease state, he/she can collect the immediate reward associated at the state. Assume doctors (experts) use a policy to optimize the long-term expected average reward. We wish to learn the prescription policy and disease transition probabilities from observed tuples of (state, action, next state).

Assume that the best state $(0,0)$ is associated with reward R_0, and the associated reward can "spread" according to a Gaussian distribution, i.e., immediate rewards are defined as follows:

$$R(\mathbf{s},\mathbf{a}) = R(\mathbf{s}) = R_0 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-\|\mathbf{s}\|^2}{2\sigma^2}},$$

for all actions $\mathbf{a}$. The parameter σ controls the discounting rate of reward being decreased as the disease state becomes worse. The parameters are tunable to achieve desirable behaviors [Matignon et al., 2006]. In this experiment, we set $R_0 = 30$ and $\sigma = 10$.

4.6.2 Policy and Transition Probability Learning

Designing appropriate features is essential for our learning algorithm. Following the approach of [Hanawal et al., 2018, Sec. IV], the features are based on a set of representative states. Regarding transition probabilities, we assume

$$P_\xi(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t) = \frac{\exp\{\xi'\psi(\mathbf{s}_{t+1}-\mathbf{s}_t,\mathbf{a}_t)\}}{\sum_{\mathbf{s}\in\mathcal{S}}\exp\{\xi'\psi(\mathbf{s}-\mathbf{s}_t,\mathbf{a}_t)\}}.$$

The feature vector is constructed as follows. We select a set of p representative state differences $\mathcal{R} = \{(i_k,j_k) : i_k, j_k \in \{-1,0,1\},\ k = 1,\ldots,p\}$ and define feature functions:

$$\psi_{i_k,j_k}(\mathbf{z},\mathbf{a},\mathbf{b}) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{\frac{-\|\mathbf{z}-(i_k,j_k)\|^2}{2}}, & \text{if } \mathbf{b} = \mathbf{a}, \\ 0, & \text{otherwise,} \end{cases}$$

for $k = 1,\ldots,p$ and $\mathbf{b} \in \mathcal{A}$. Then, $\psi(\mathbf{z},\mathbf{a}) = (\psi_{i,j}(\mathbf{z},\mathbf{a},\mathbf{b});\ \forall(i,j) \in \mathcal{R},\ \forall\mathbf{b} \in \mathcal{A})$.

Similarly, assume that the original policy has the following form

$$\mu_\theta(\mathbf{a}|\mathbf{s}) = \frac{\exp\{\theta'\phi(\mathbf{s},\mathbf{a})\}}{\sum_{\mathbf{b}\in\mathcal{A}}\exp\{\theta'\phi(\mathbf{s},\mathbf{b})\}}.$$

In this case, the feature vector is constructed by selecting a set $\mathcal{B}$ of representative states from the state space and defining features

$$\phi_{\mathbf{u}}(\mathbf{s},\mathbf{a}) = \sum_{\mathbf{q}} P_\xi(\mathbf{q}|\mathbf{s},\mathbf{a}) f_{\mathbf{u}}(\mathbf{q}), \qquad \mathbf{u}\in\mathcal{B},$$

where

$$f_{\mathbf{u}}(\mathbf{q}) = \frac{1}{\sqrt{2\pi}} e^{\frac{-\|\mathbf{u}-\mathbf{q}\|^2}{2}}.$$

Then, the feature vector is $\phi(\mathbf{s},\mathbf{a}) = (\phi_{\mathbf{u}}(\mathbf{s},\mathbf{a});\ \forall \mathbf{u}\in\mathcal{B})$.

4.6.3 Performance Evaluation

For the medical application discussed in Sections 4.6.1,we solve the corresponding MDP using value iteration and obtain an optimal policy. We use this policy to generate (state, action, next state) tuples. Given the observed tuples, we will estimate a policy $\mu_\theta(\mathbf{a}_t|\mathbf{s}_t)$ parameterized by θ and an induced transition probability model $P_\xi(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$, parameterized by ξ using the feature functions described in Section 4.6.2.

We compare the average rewards of the following four policies:

1. Optimal policy: obtained from the value iteration algorithm and used to generate (state, action, next state) tuples.

2. ℓ_1-regularized policy: trained using Algorithm 4 using ℓ_1-regularized logistic regression.

3. Unregularized policy: trained using Algorithm 4 to solve the logistic regression problems without regularization.

4. Greedy policy: at each state taking an action which maximizes the expected immediate reward.

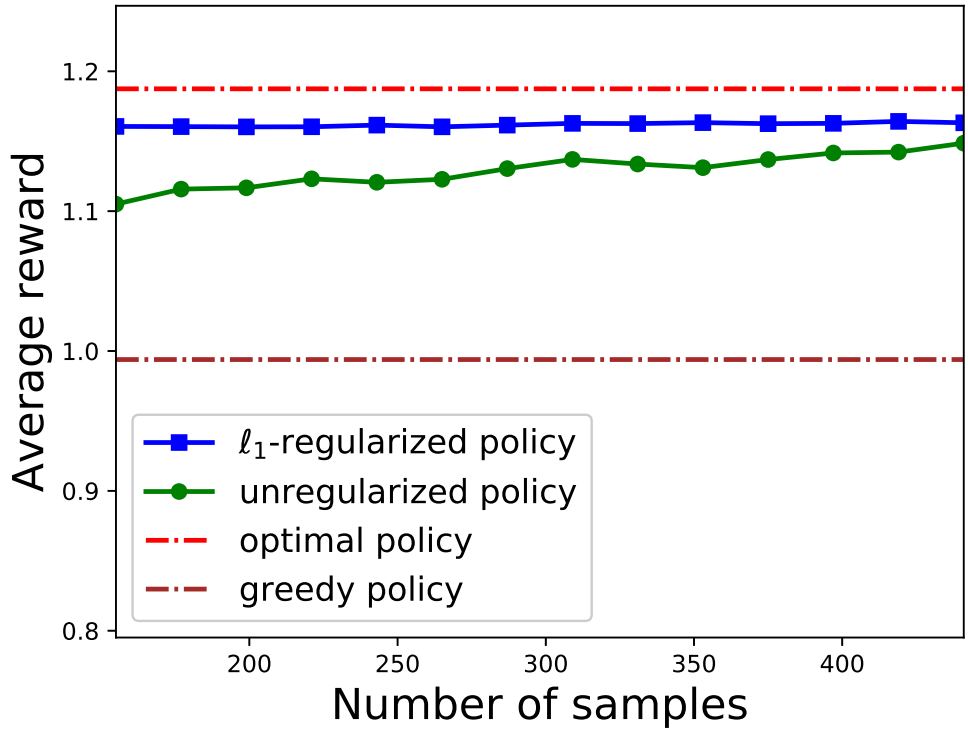

Figure 4·1: Disease progression example: average rewards vs. sample size.

To investigate how training sample size influences the performance of our proposed algorithm, we implemented Algorithm 4 using different sample sizes m. For each sample size, we conduct 10 independent runs, obtain a policy from each run, and then simulate this policy under the true MDP dynamics to assess its performance. We average over the 10 runs the per-run average reward obtained. These averages are plotted Figures 4·1 for the healthcare application.

We can see that the ℓ_1-regularized policy achieves reward that is closer to the reward achieved by the unknown original policy, which is consistent with Theorem 4.5.3. In addi-

tion, the ℓ_1-regularized policy consistently outperforms the greedy and unregularized policies, especially in the case of small sample sizes. This illustrates the benefits of regularization and is consistent with our related comments in the Introduction.

In addition, we compare two different estimates of the conditional transition probabilities as follows:

1. an ℓ_1-regularized estimate, where we use Algorithm 4 with the ℓ_1-regularized logistic regression to learn the conditional transition probability model; and

2. an unregularized estimate, where we use Algorithm 4 with unregularized logistic regression to learn the conditional transition probability model.

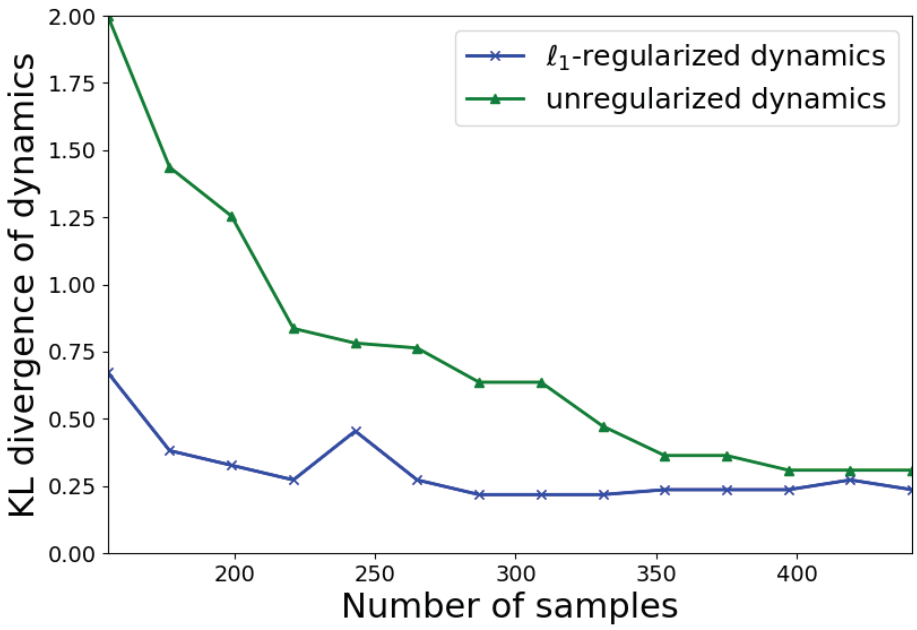

Figure 4·2: Disease progression example: average KL divergence between the true and estimated conditional transition probabilities.

Specifically, we computed the KL-divergence between the true and estimated conditional transition probabilities $D_{\xi^*,\theta^*}(\mathbf{P}_{\xi^*}\|\mathbf{P}_{\hat{\xi}}) = \varepsilon(\hat{\xi}) - \varepsilon(\xi^*)$ as a function of sample size.

The KL-divergence result for the healthcare example is shown in Figures 4·2, respectively. The lower KL-divergence is, the better the estimate is. Through all different sample sizes in the two examples, the ℓ_1-regularized estimate consistently outperforms its unregularized alternative, and more so in the case of small sample sizes. In addition, the regularization algorithm achieves accurate estimates with only a very small number of samples, which validates Corollary 4.4.4.

Note that even though the original policy and conditional transition probabilities do not follow the parametric forms in equations (4.1) and (4.2), our algorithm is still able to achieve high accuracy.

4.7 Conclusions

This work investigates the problem of learning a policy and an associated transition probability model in an MDP based on observed (state, action, next state) tuples. We propose two regularized logistic regression models to estimate the true transition probabilities and original policy. We establish out-of-sample generalization bounds on log-loss for the policy and transition probability estimation. Further, we derive an upper bound on the regret of the estimated policy under the learned transition probability model. The theoretical results are validated in a disease progression application. The numerical results show satisfactory learning of both the policy and transition dynamics and illustrate the benefits of using robust learning techniques.

Chapter 5

Personalized Pharmacological Therapy Recommendations

Diabetes is recognized as the worldwide fastest growing chronic disease. One in eleven adults worldwide suffers from diabetes (415 million), and 12% of global healthcare expenditures ($673 billion) were spent on diabetes in 2015 [International Diabetes Federation, 2015]. In the U.S. alone, diabetic patients constituted 9.4% of the population in 2015 (30.3 million), and another 84.1 million had prediabetes, a condition that usually results in type 2 diabetes within five years if no treatment is applied [Centers for Disease Control and Prevention and others, 2017]. The corresponding total costs of diagnosed diabetes amount to $327 billion in 2017, up from $245 billion in 2012, and representing a 26% increase over a five-year period. Furthermore, the treatment in the form of anti-hyperglycemic medications to directly treat diabetes costs about $30.8 billion annually in the U.S. [American Diabetes Association, 2018b]. In this work, we focus deliberately on the prescriptive analytics with applications to diabetes.

Among all patients with diabetes, type 2 diabetes patients account for $90\% - 95\%$. For these patients, the first and foremost task is to properly control glycemia. A widely used measure of long-term glycemia is glycated hemoglobin (HbA1c [1]), which reflects average glycemia over a 3-month period. There is a close relationship between the risks of complications and glycemia [American Diabetes Association, 2003, Stratton et al., 2000].

[1] The HbA1c values in our work are reported in "Diabetes Control and Complications Trial" (DCCT) units and indicated by percentage, representing the percentage of glycated hemoglobin to total hemoglobin [Sacks, 2012].

Specifically, [Stratton et al., 2000] shows that people with type 2 diabetes who reduce their HbA1c level by one percentage point (e.g., from 9.0% to 8.0%) are 19%, 16%, and 43% less likely to suffer cataracts, heart failure, and amputation or death due to peripheral vascular disease, respectively. Glycemic control can be achieved by lifestyle changes and pharmacotherapy that targets glucose levels [American Diabetes Association, 2018a]. Boosting the treatment efficiency and quality is, unquestionably, an important advance that can contribute to the reduction of medical costs and improve patients' long-term quality of life. It is also worth mentioning that the quality of care, access to specialists, and physicians' recommendations vary drastically depending on location. To address these issues, a potential solution is *automated personalized treatment recommendations* based on patient Electronic Health Records (EHR) [Mathur and Sutton, 2017]. These can serve as a suggestion to prescribing physicians and help increase uniformity of prescriptions across different geographical locations.

There is limited past research in the area of prescription recommendations; in general the problem has not received a lot of attention, whereas predictive models (e.g., of glucose concentrations [Georga et al., 2012], or complications from diabetes [Zarkogianni et al., 2017]) have been better studied. One line of work treats the patient as a dynamic system and applies control theory (see e.g., [Zhao et al., 2014]). [Calderón and Jaimes, 2018] proposes a mechanism for stress interventions based on fuzzy control theory. Other works use Markov Decision Process (MDP) techniques to provide breast and ovarian cancer intervention strategies [Abdollahian and Das, 2015]. Related work has also been carried out using Reinforcement Learning (RL), which is able to learn the dynamics and cost functions through interaction with the real environment [Asoh et al., 2013]. Similar in scope to our learning paradigm, but different in terms of the approach used, is the work in [Yoon et al., 2016], which learns rewards/costs of prescriptions from data and recommends actions that optimize these empirical rewards; the approach is applied to selecting chemotherapy drugs.

Most of the aforementioned works suffer from a number of pitfalls: (a) Impractical assumptions; both the control theory and the RL-based methods require (or need to learn/construct) a highly accurate model or simulation system to characterize the interaction between treatments and patients' disease progression, which is a difficult research problem in itself and highly dependent on domain expertise. Besides, the Markov assumption needs to be validated for an MDP framework. (b) Lack of generalization; a simulation or dynamic system designed for one disease is not easily migratable to other diseases. (c) Computationally demanding; due to the exponentially computational complexity, MDP models can only involve very few disease states, which are often too simple to characterize the complex decision-making process of physicians. Hence, properly designing the states and the action spaces to balance the trade-off between accurate disease characterization and computational complexity is also a challenging issue. (d) Non-personalized recommendations; MDP-based models group patients into various states, therefore they can only provide group recommendations.

The present work seeks to address the personalized treatment recommendation problem with solutions that are computationally effective and can be easily generalized without the need for a complex simulation environment. The primary challenge for the personalized therapy recommendation is to evaluate the patient-specific therapeutic effect of a treatment, because the effect of the same pharmacotherapy treatment may vary from individual to individual. This could be achieved through clinical trials, however, since a clinical trial can only involve a single treatment, it is impossible to quantify the effects of various prescription treatments on the same person under the same conditions, not to mention that conducting clinical trials for a large amount of patients is both economically ineffective and time-consuming. Fortunately, we are able to access a large number of patients' medical data with the increasingly available EHRs, from which we can recognize useful patterns about patient-specific treatment effects and the doctors' prescription algorithm.

There are naturally two *data-driven* strategies for personalized treatment recommendations. The first strategy is to learn the treatment algorithm directly from the doctors' prescriptions; this method takes advantage of the doctors' domain expertise and is more likely to be accepted; however, it is hard to improve the doctors' (implicit) algorithm simply by learning from them. Another strategy is to directly select the treatment that optimizes the *predicted* prescription effect [Bertsimas et al., 2017]; yet, this method may produce problematic advice, especially when the accuracy of the predictive model is low.

Our work can be seen as a combination of the above two strategies. Specifically, we first use regression models to predict the patient-specific treatment effect, then learn the current physicians' prescription algorithm from data, and finally improve the learned algorithm by optimizing over its parameters. A related approach is based on a weighted k-nearest neighbor regression model [Bertsimas et al., 2017]. However, there are several important differences between the two approaches. First, our work designs a parametric randomized model which effectively learns the current prescription algorithm and enables better understanding of the physicians' prescriptions from a statistical perspective. Second, the treatment recommendations in [Bertsimas et al., 2017] depend only on the prescription effect regression models and can be problematic especially when the regression model has low accuracy. In contrast, our recommendations incorporate information from both the estimated current prescription model and the treatment effect regression models, leading, as we will see, to improved outcomes. Finally, our model provides randomized recommendations, assigning a probability as confidence score for each recommended treatment, which provides additional nuanced prescription information for physicians who could make a decision by taking into account factors that may not have been captured by the model.

The remainder of this chapter is organized as follows. Sec. 5.1 outlines our approach to the personalized prescription problem and decomposes the problem into three subproblems: prescription effect prediction, learning the physicians' prescription algorithm, and

improving this prescription algorithm. The details of solving these subproblems are presented in Sec. 5.2 – Sec. 5.4, respectively. Sec. 5.5 presents experimental results, using our approach to make diabetic treatment recommendations. Conclusions are drawn in Sec. 5.6.

5.1 Problem Definition

Our objective is to learn and improve the physicians' prescription algorithm inferred from patients' EHR data in order to optimize treatment outcomes, for instance, to better control the blood glucose level measured by HbA1c for diabetic patients. Our personalized prescription framework includes the following three subproblems: (a) developing predictive models to estimate the effect (on HbA1c) of a specific prescription; (b) learning a representation of the physicians' prescription algorithm from the EHR data so that we can predict the prescription for a new patient; and (c) developing an approach to modify the estimated prescription algorithm to achieve better (HbA1c) results.

We briefly outline our approach to each of the three subproblems. For subproblem (a), we develop multiple regression models to evaluate the effects of various pharmacological treatments on diabetic patients. We will evaluate the accuracy of different models and assess their consistency. For subproblem (b), based on the predicted prescription outcome and the patients' medical history, we will formulate the prescription learning problem as a multi-class classification problem and obtain a parametric model that can predict the physicians' prescriptions. For subproblem (c), by altering the parametric model, we seek to improve the current algorithm and achieve better glycemia control.

We use the following notation throughout the work. We denote by $\mathbf{X} \in \mathbb{R}^{n \times d}$ the patients' health records, where rows and columns correspond to the n patients and their d features, respectively; each feature is a medical fact extracted from the patient's EHR. We use $\mathbf{y} \in \mathbb{R}^n$ to represent the prescription effect (future HbA1c; one entry per patient) under the current (physicians') prescriptions. We let L be the number of different drug regimens

that can be prescribed. We group all available records based on the type of prescription given and denote by $\mathbf{X}^\ell \in \mathbb{R}^{n_\ell \times d}$ the set of n_ℓ records where the patients' current treatment is of type $\ell = 1, \ldots, L$, with $\mathbf{y}^\ell \in \mathbb{R}^{n_\ell}$ denoting the corresponding outcome (true future HbA1c).

5.2 Predicting the Prescription Effects

We adopt both linear and nonlinear regression models to predict the effect of the ℓ-th treatment type, $\ell = 1, \ldots, L$. Among linear regression models, the following methods are considered:

(1) Least Absolute Shrinkage and Selection Operator (LASSO) regression [Tibshirani, 1996]:

$$(\hat{\beta}^\ell_{LA}, \hat{\beta}^\ell_{LA,0}) = \arg\min_{\beta^\ell, \beta^\ell_0} \|\mathbf{X}^\ell_t \beta^\ell + \beta^\ell_0 - \mathbf{y}^\ell_t\|_2^2 + \lambda \|\beta^\ell\|_1;$$

(2) Elastic Net regression (EN) [Zou and Hastie, 2005]:

$$(\hat{\beta}^\ell_{EN}, \hat{\beta}^\ell_{EN,0}) = \arg\min_{\beta^\ell, \beta^\ell_0} \|\mathbf{X}^\ell_t \beta^\ell + \beta^\ell_0 - \mathbf{y}^\ell_t\|_2^2 + \lambda_1 \|\beta^\ell\|_1 + \lambda_2 \|\beta^\ell\|_2.$$

The input to the above model is the training dataset, denoted by $\{\mathbf{X}^\ell_t, \mathbf{y}^\ell_t\}$, which is some random subset of $\{\mathbf{X}^\ell, \mathbf{y}^\ell\}$ defined earlier. The model (and output of the training process) is defined in terms of the model parameters $(\beta^\ell, \beta^\ell_0)$. Based on the coefficients β^ℓ in the model, we can assess the importance of various features in predicting the treatment effects. The regularization parameters λ, λ_1, λ_2 control the tradeoff between minimizing the empirical risk (error) and the level of robustness of the model. Recent work, e.g., [Chen and Paschalidis, 2018], has established a connection between regularization and robustness, essentially showing that the regularized problem solves a robust version of empirical risk minimization (i.e., minimizing the error over the training set). When $\lambda = 0$ or $\lambda_1 = \lambda_2 = 0$, the models are equivalent to the Ordinary Least Squares (OLS) regression model.

LASSO regression employs ℓ_1-norm regularization, where a large enough λ will drive some of the elements in the coefficient vector β^ℓ to be zero, thus, leading to sparse models that are easily interpretable and offering effective feature selection [Friedman et al., 2001]. The Elastic Net regression, adds an ℓ_2-norm regularizer and has a monotonicity property: as λ_1 increases, coefficient values driven to zero with smaller λ_1 remain at zero; thus, coefficient values progressively are driven to zero [Zou and Hastie, 2005]. In general, the form of the regularizer is connected with the proper distance metric in the original $(\mathbf{x}, y)$ data space [Chen and Paschalidis, 2018].

For both problems, the regularization terms can be turned into constraints, e.g., $\|\beta^\ell\|_1 \leq m$ for LASSO, and the corresponding problems solved as convex quadratic programming problems with convex constraints (linear in the case of LASSO, quadratic for Elastic Net). Such problems can be easily solved by sequential quadratic programming or interior point methods [Tibshirani, 1996]. Besides, two efficient methods based on Least Angle Regression (LARS) are applicable to LASSO [Efron et al., 2004] and Elastic Net regression [Zou and Hastie, 2005], respectively.

More complex nonlinear and non-parametric models, like Random Forests and *k*-nearest-neighbor based methods, can also be employed. Random Forests (RF) regression is an ensemble method training many de-correlated decision trees to construct a complex model where the prediction for a new data point is made by averaging the predictions from the base tree models [Breiman, 2001]. The key intuition is that the average of many uncorrelated base tree models is more robust to noise, hence, the ensemble regression model will have lower variance and will be less likely to overfit than a single tree model.

The *k*-nearest neighbor method (*k*NN) is a non-parametric approach that can be used for both regression and classification. The prediction for a new test sample is the average over its *k* nearest neighbors in the training set. The intrinsic local information property makes *k*NN-based models sensitive to the local structure of data. Besides, the performance of *k*NN

models is also known to suffer from the computational complexity in high dimensional cases and lacks robustness to noisy data [Hinneburg et al., 2000].

To provide a comprehensive comparison with the work in [Bertsimas et al., 2017], we also consider a weighted-kNN (WkNN) model. Instead of the Euclidean distance used in kNN, the WkNN adopts the coefficients from a linear regression model to serve as the weights of features when calculating the distance between samples. In our implementation, we use the coefficients from LASSO since it provides better regression accuracy than ordinary least squares regression.

We split all data into a training and a test set, and derive a model based on the training set, where the trained model can be any type of the aforementioned regression models. Given a test sample $\mathbf{x}_i$, a treatment type $\ell = 1, \ldots, L$, and the true effect $y(\mathbf{x}_i, \ell)$, the predicted effect (future HbA1c) under treatment type ℓ will be denoted by $\hat{y}_i = \hat{y}(\mathbf{x}_i, \ell)$. To evaluate the accuracy of the regression methods, we consider the coefficient of determination R^2 out-of-sample (i.e., in the test set):

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \frac{1}{n}\sum_i y_i)^2},$$

which is a classic accuracy criterion for regression models and represents the proportion of the variance in the response explained by the predictive model.

5.3 Learning the Physicians' Prescription Algorithm

Naturally, when the physicians write a prescription, they not only consider the patients' current condition, but also take into account the potential patient-specific effects of different prescriptions. Therefore, for the problem of learning the physicians' prescription algorithm, besides patients' demographic and medical information, we also incorporate the predicted effects obtained from Sec. 5.2 as additional input features.

We let $\mathbf{u} \in \{1, \ldots, L\}^n$ denote the doctors' prescription treatment types for each patient.

Let also $\hat{\mathbf{Y}} \in \mathbb{R}^{n \times LM}$ denote the prescription effects predicted by all M regression methods $\{\mathcal{A}_1, \ldots, \mathcal{A}_M\}$, where the $((m-1)L+1)$-st to mL-th columns of $\hat{\mathbf{Y}}$ contain the predicted effects under prescription type $\ell = 1, \ldots, L$ from the m-th regression method $\mathcal{A}_m$ in Sec. 5.2, where $m = 1, \ldots, M$. We denote by $\mathbf{Z} = [\mathbf{X}\ \hat{\mathbf{Y}}]$ the combined feature set, consisting of the patient's health records $\mathbf{X}$ and the predicted effects for each patient.

Our objective is to learn the current physicians' prescription algorithm; specifically, find a mapping from the combined features $\mathbf{Z}$ to actions $\mathbf{u}$. This can be formulated as a multi-class classification problem. We adopt the multinomial logistic regression (also called softmax regression) method to solve this problem [Hosmer Jr et al., 2013]. The probability of the sample $\mathbf{z}_i = (\mathbf{x}_i, \hat{\mathbf{y}}_i)$ being prescribed with treatment type ℓ is:

$$P(u_i = \ell | \mathbf{z}_i) = \frac{\exp\{(\theta^\ell)' \phi(\mathbf{z}_i)\}}{\sum_{\ell=1}^{L} \exp\{(\theta^\ell)' \phi(\mathbf{z}_i)\}}, \tag{5.1}$$

where θ^ℓ is a weight vector corresponding to treatment type ℓ and $\phi(\mathbf{z}_i)$ is a feature vector which, in general, is some (potentially nonlinear) transformation of the i-th record $\mathbf{z}_i$. The mapping $\phi(\cdot)$ can be obtained by feature selection methods such as REcursive Feature selection (REF) [Guyon et al., 2002] and Principle Component Analysis (PCA) [Hotelling, 1933].

The model parameter θ^ℓ in (5.1) can be determined through maximum likelihood estimation, and the detailed problem formulation is an ℓ_1-regularized softmax regression [Krishnapuram et al., 2005]:

$$\begin{array}{ll} \min_{\theta^\ell} & -\sum_{i=1}^{n_t} \log \frac{\exp\{(\theta^{u_i})' \phi(\mathbf{z}_i)\}}{\sum_{\ell=1}^{L} \exp\{(\theta^\ell)' \phi(\mathbf{z}_i)\}} \\ \text{s.t.} & \|\theta^\ell\|_1 \leq \gamma, \quad \forall \ell. \end{array} \tag{5.2}$$

The inputs and outputs to the model are the training dataset $\{(\phi(\mathbf{z}_i), u_i),\ i = 1, \ldots, n_t\}$, and model parameters $\{\theta^\ell,\ \ell = 1, \ldots, L\}$, respectively. The ℓ_1 regularization is well known to prevent over-fitting, induce sparsity, and is thus useful for feature selection [Lee et al., 2006]. Problem (5.2) is a convex optimization problem due to the convexity of both the

objective function and the feasible set. [Krishnapuram et al., 2005] designs a fast iterative method following the bounded optimization approach [Lange et al., 2000], which iteratively maximizes sequential quadratic surrogate functions. Another method, "SAGA", recently proposed in [Defazio et al., 2014], is an incremental gradient method with a linear convergence rate.

For the testing phase, given a test sample $\mathbf{x}_{test} \in \mathbb{R}^d$, we first obtain its predicted effects $\hat{\mathbf{y}}_{test} \in \mathbb{R}^{LM}$ by the regression methods in Sec. 5.2, and then create the new feature vector $\mathbf{z}_{test} = (\mathbf{x}_{test}, \hat{\mathbf{y}}_{test})$. The learned prescription type is determined by picking the action with the maximum likelihood:

$$\begin{aligned} u^*(\mathbf{z}_{test}) &= \arg\max_{\ell} P(u_i = \ell | \mathbf{z}_{test}) \\ &= \arg\max_{\ell} \frac{\exp\{(\tilde{\theta}^{\ell})' \phi(\mathbf{z}_{test})\}}{\sum_{\ell=1}^{L} \exp\{(\tilde{\theta}^{\ell})' \phi(\mathbf{z}_{test})\}}, \end{aligned} \tag{5.3}$$

where $\tilde{\theta}^{\ell}$ is the optimal parameter vector obtained from (5.2). To evaluate the efficiency of the multi-class classification method, we use the out-of-sample prediction accuracy, which is the percentage of the correctly predicted samples among all test samples.

5.4 Improving the Prescription Algorithm

Next, we want to obtain a prescription recommendation algorithm which leads to better prescription effects (lower future HbA1c) by optimizing over the parameters of the learned algorithm. Given a sample $\mathbf{x}$, the idea is to select the action which minimizes the future predicted HbA1c, i.e.,

$$u_A^*(\mathbf{x}) = \arg\min_{u} \hat{y}_A(\mathbf{x}, u),$$

where $A \in \{\mathcal{A}_1, \ldots, \mathcal{A}_M\}$ denotes the regression method to be used and $\hat{y}_A(\mathbf{x}, u)$ the predicted future HbA1c under A.

Of course, relying too much on the regressed future HbA1c runs the risk of being misled

by an inaccurate prediction. To that end, we introduce a form of "hedging," which will find an algorithm consistent with actions $u_A^*(\mathbf{x})$ that does not deviate too much from the algorithm learned from physician actions. To that end, we formulate the following problem:

$$\begin{array}{ll} \min_{\theta^\ell} & -\sum_{i=1}^n \log \frac{\exp\{(\theta^{u_A^*(\mathbf{x}_i)})'\phi(\mathbf{z}_i)\}}{\sum_{\ell=1}^L \exp\{(\theta^\ell)'\phi(\mathbf{z}_i)\}} \\ \text{s.t.} & \|\theta^\ell\|_1 \le \gamma, \quad \forall \ell, \\ & \|\tilde{\theta}^\ell - \theta^\ell\|_2 < \eta, \quad \forall \ell, \end{array} \tag{5.4}$$

where γ and η are fixed positive numbers, and $\tilde{\theta}^\ell$, $\ell = 1, \ldots, L$, are estimated model parameters obtained by solving the prescription learning problem (5.2) in Sec. 5.3. This model incorporates information from both the prescription effect regression and the current physicians' prescription algorithm model. Notice that the objective function aims to train a parametric recommendation algorithm which learns the actions optimizing the prescription effects, and the constraints restrict the difference between the recommended and the original algorithm; in effect introducing some form of regularization. The problem is a constrained ℓ_1-regularized softmax regression problem with $\phi(\mathbf{z}_i)$ as input features and $u_A^*(\mathbf{x}_i)$ as the target labels.

Unlike a deterministic algorithm, the randomized prescription algorithm obtained from (5.4) provides a probability associated with each possible prescription; specifically,

$$P(\hat{u}(\mathbf{z}) = \ell) = \frac{\exp\{(\hat{\theta}^\ell)'\phi(\mathbf{z})\}}{\sum_{\ell=1}^L \exp\{(\hat{\theta}^\ell)'\phi(\mathbf{z})\}}$$

is the probability of recommended action $\hat{u}(\mathbf{z})$ for a patient with feature vector $\mathbf{z}$, where $\hat{\theta}^\ell$ are the optimal model parameters obtained by (5.4). This additional probabilistic information can be quite useful in assisting physicians' decisions compared to a deterministic algorithm, especially in the cases where there are two or more treatments with similar probability scores.

When evaluating the recommended prescription algorithm, we select the most likely

action obtained from the recommended prescription algorithm for each sample $\mathbf{z}$, that is,

$$u^*(\mathbf{z}) = \arg\max_{\ell} \frac{\exp\{(\hat{\theta}^{\ell})'\phi(\mathbf{z})\}}{\sum_{\ell=1}^{L} \exp\{(\hat{\theta}^{\ell})'\phi(\mathbf{z})\}}. \tag{5.5}$$

We will use the regression methods from Sec. 5.2 to predict the treatment effect (future HbA1c). We will compare the predicted future HbA1c with its counterpart corresponding to the original physicians' prescription algorithm and calculate the relative improvement ratio in Sec. 5.5.

5.5 Experimental Results on an Actual Diabetic Dataset

In this section, we will use our approach to offer personalized prescription recommendations for patients with type 2 diabetes. The EHR dataset we use to train the various models and evaluate their performance comes from the Boston Medical Center (BMC). BMC is an academic medical center and the largest safety-net hospital in New England; with its 14 affiliated community health centers provides care for about 30% of Boston residents. BMC provides a full range of pediatric and adult care services, from primary care to advanced specialty care. The EHRs from the hospital and information from community health centers constitute a rich and diverse dataset.

5.5.1 Data Descriptions and Preprocessing

We select all medical and demographic records from 01/01/2001 to 12/31/2014 for patients who satisfy the following conditions: (a) received a prescription of at least one anti-hyperglycemic agent, including insulin, metformin, sulfonylureas, etc; (b) had at least two long-term blood glucose laboratory measurement records (HbA1c); and (c) did not have a Type 1 diabetes diagnosis record. There are $8,893$ such patients in total.

The diabetes-related medical history of the patients is described by various categories of medical factors as described below.

1. Demographics: Gender, Age, Race.

2. Diagnoses: Diabetes mellitus with complications, Hypertensive disease, Heart failure, Skin infections, etc.

3. Procedures (CPT or ICD9): Procedure on single vessel, Insertion of intraocular lens prosthesis at time of cataract extraction, Venous catheterization, Transfusion of packed cells, etc.

4. Admissions: e.g., Diabetes (with and without) complications, Heart failure and shock, Chest pain, Chronic obstructive pulmonary disease, Acute myocardial infarction, O.R. procedures for obesity.

5. Service By Department: Inpatient (admit), Inpatient (observe), Outpatient, Emergency Room.

6. Laboratory Test Values: Hematology, Chemistry, Urinalysis, Coagulation tests.

7. Vital Signs: e.g., Blood pressure, Pulse, Respiratory rate, Temperature, Body Mass Index (BMI).

8. Blood Glucose Regulation Agents: Insulin, Oral Anti-hyperglycemic agents, etc.

In order to organize the available information in a uniform way for all patients, the following preprocessing procedure has been applied to the dataset.

(a) For each patient, we split the patients' medical history into 90-day windows starting from 30 days before their first HbA1c measurement. The selection of a 90-day period is based on the commonly used 3-month observation period in the "Standards of Medical Care in Diabetes", which is the most authoritative guideline for diabetes care [American Diabetes Association, 2018a].

(b) To summarize the features of a patient in each time window, the average values of continuous features and the number of occurrences of the categorical values in the window are calculated, respectively.

(c) The glucose regulation agents for Type 2 diabetes can be grouped into two main types, oral-type agents, including Metformin, sulfonylureas, thiazolidinediones, etc; and injectable-type agents like insulin, GLP-1 Receptor Agonists, Amylin Analogue, etc. [American Diabetes Association, 2018a]. According to the medications used in a 90-day period, we group the treatments into three types with gradually increasing power: (1) no treatment, (2) treatment with only oral agents, and (3) treatment with injectable agents involved.

(d) To make full use of the medical history for the patient under consideration, we use the patient's medical records in the two periods preceding the target period, including HbA1c, treatment types and all other medical factors. A typical patient sample for our prescription framework is $\mathbf{x}_t = (\mathbf{v}_{t-2}, \mathbf{v}_{t-1}, \mathbf{v}_t)$, where $\mathbf{v}_{t-i}$ is a vector of features reflecting the patient's medical records in the period which is i periods before the target period. Note that a patient with a long medical history can form multiple such samples corresponding to various choices of the target period. We denote by u_t the target-period treatment type based on the physicians' prescription in the EHR, and by y_{t+1} the observed in the EHR average HbA1c during the next period.

(e) We remove the records where either the target-period or the next-period average HbA1c value is missing, and then replace other continuous missing features with the corresponding median. For categorical missing features, we introduce another "missing" indicator variable. These preprocessing steps lead to a total of $37,756$ patient samples.

5.5.2 Experimental Results

We randomly split the dataset into a training set and a test set with $2/3$ and $1/3$ of all samples, respectively. All model hyper-parameters are determined through 5-fold cross-validation using the training set. The training and test sets are fixed as we solve the three subproblems in the personalized prescription recommendation framework (as outlined in Sec. 5.1). For the samples in the training set, we assume that the true next-period HbA1c and the current-period (target-period) treatment type are available. Unless otherwise specified, all the experiments are repeated for 10 runs with randomly selected training sets.

Modeling of the prescription effects

We group all the samples based on the type of current-period prescribed treatment. For all patients with treatment type ℓ, a regression model is trained on the training set by the methods in Sec. 5.2. The inputs of the regression models are $\{\mathbf{X}^\ell, \mathbf{y}^\ell\}$, where $\mathbf{X}^\ell \in \mathbb{R}^{n_\ell \times d}$ denotes the n_ℓ records of patients whose current prescription treatments are of type ℓ, and $\mathbf{y}^\ell$ is the corresponding true effect (future HbA1c), $\ell = 1, \ldots, L$. We repeat the experiments for 10 runs with randomly selected training sets, and show in Table 5.1 the mean and standard deviation (std) of the out-of-sample prediction accuracy (R^2) for various regression methods under the three treatment types, where $u = 0, 1, 2$ represents no treatment, treatment with only oral agents, and treatment with injectable agents involved, respectively. Random Forests yields the best prediction accuracy and kNN the worst compared to other alternatives. In the sequel, we will consider all regression methods discussed in Sec. 5.2 except kNN as effective regression models.

Due to the feature selection capability of Elastic Net regression, we present in Table 5.2 the most important features obtained from the Elastic Net model for "oral treatments" which has the maximal accuracy over the 10 runs. We then rank them by the absolute values of the corresponding model coefficient. (Variables have been standardized, hence,

Table 5.1: Model accuracy for treatment effect prediction.

		u=0	u=1	u=2
LASSO (LA)	mean(R^2)	0.55	0.51	0.45
	std(R^2)	0.01	0.01	0.01
Elastic Net (EN)	mean(R^2)	0.55	0.51	0.46
	std(R^2)	0.01	0.01	0.01
Random Forest (RF)	mean(R^2)	0.56	0.53	0.53
	std(R^2)	0.01	0.01	0.01
WkNN	mean(R^2)	0.53	0.51	0.50
	std(R^2)	0.01	0.01	0.01
kNN	mean(R^2)	0.25	0.20	0.26
	std(R^2)	0.01	0.01	0.002

the coefficients are comparable.) Positive coefficients imply a positive correlation between the feature and outcome. The most important features include: historical HbA1c, historical oral treatment information, blood pressure, age, various types of glucose, and mean corpuscular hemoglobin which is the average mass of hemoglobin per red blood cell in a blood sample. Endocrinologists would agree that there is a high correlation between future HbA1c and these factors.

Learning a prescription algorithm from data

Next, we apply the method of Sec. 5.3. From the previous step, we can obtain the predicted prescription effects $\hat{y}(\mathbf{x})$ for each sample $\mathbf{x}$, and we can create a new combined feature vector $\mathbf{z}(\mathbf{x}) = (\mathbf{x}, \hat{\mathbf{y}}(\mathbf{x}))$. The transformed feature $\phi(\mathbf{z})$ in Equation (5.1) can be selected as a subset of the following features: prescription effects predicted by distinct methods, HbA1c in the current and preceding two periods, treatment history and all other available features. Once we learn the model parameters θ^ℓ for all ℓ from (5.2), we can then use the maximum likelihood rule (5.3) to select a prescription-type for a new test sample.

For different selections of the feature vector $\phi(\mathbf{z})$, Tables 5.3 and 5.4 list the mean and standard deviation (std) of the prescription learning accuracy, defined as the percentage of

Table 5.2: The most important features from Elastic Net regression for predicting future HbA1c under oral treatments.

Rank	Feature	Coef
1	Target-period HbA1c	0.73
2	Target-period blood glucose tested by finger stick	0.32
3	HbA1c, 2 periods before the target period	0.24
4	HbA1c, 1 period before the target period	0.13
5	Target-period blood pressure	0.08
6	Oral treatment was prescribed, 2 periods before the target period	0.07
7	Age	0.05
8	Target-period point-of-care glucose	0.05
9	Target-period erythrocyte mean corpuscular hemoglobin (MCH RBC Qn)	-0.04
10	Referred to an endocrinologist, 2 periods before the target period	0.04

correctly predicted prescriptions in the test set. In these tables, $\hat{\mathbf{y}}_A$ denotes the predicted future HbA1c from regression method A and the first column lists the variables included in the feature vector $\phi(\mathbf{z})$. (We use the abbreviations defined in Table 5.1 for the various methods.) For example, the row with features $\hat{\mathbf{y}}_{LA}$ and $\hat{\mathbf{y}}_{EN}$, uses a 2-dimensional feature vector with elements the predicted future HbA1c from LASSO and Elastic Net regression. Table 5.3 suggests that when using the predicted effects alone as features for prescription learning, regression methods with higher accuracy in Table 5.1, e.g., Random Forest, do not necessarily result in more accurate learning of prescriptions.

In Table 5.4, we expand the set of features provided to the prescription learning algorithm. Specifically, $\hat{\mathbf{y}}_{Algos}$ is a feature vector formed by including the predicted future HbA1c from various methods, that is, $\hat{\mathbf{y}}_{Algos} = (\hat{\mathbf{y}}_{LA}, \hat{\mathbf{y}}_{EN}, \hat{\mathbf{y}}_{RF}, \hat{\mathbf{y}}_{WkNN})$. The first row of Table 5.4 uses as features in $\phi(\mathbf{z})$ the vector $\hat{\mathbf{y}}_{Algos}$ and the average HbA1c values during the target period and the two periods preceding it. The second row of Table 5.4 adds to $\phi(\mathbf{z})$ the prescription history, and, finally, the last row adds the remaining features from the patient's

Table 5.3: Accuracy of prescription policy learning with predicted treatment effects as input features.

Features	Accuracy (Mean)	Accuracy (std)
$\hat{\mathbf{y}}_{LA}$	0.63	0.004
$\hat{\mathbf{y}}_{EN}$	0.63	0.004
$\hat{\mathbf{y}}_{RF}$	0.46	0.004
$\hat{\mathbf{y}}_{WkNN}$	0.46	0.004
$\hat{\mathbf{y}}_{LA}, \hat{\mathbf{y}}_{EN}$	0.65	0.005
$\hat{\mathbf{y}}_{LA}, \hat{\mathbf{y}}_{EN}, \hat{\mathbf{y}}_{RF}$	0.66	0.005
$\hat{\mathbf{y}}_{LA}, \hat{\mathbf{y}}_{EN}, \hat{\mathbf{y}}_{RF}, \hat{\mathbf{y}}_{WkNN}$	0.66	0.005

EHR. The results suggest that the regressed prescription effects (as done in [Bertsimas et al., 2017] for example) may not constitute a rich enough feature set to learn accurate prescriptions; adding further information from the patient's record can significantly improve accuracy.

Table 5.4: Accuracy of prescription algorithm learning with more input features.

Features	Accuracy (Mean)	Accuracy (std)
$\hat{\mathbf{y}}_{Algos}$, HbA1c history	0.71	0.003
$\hat{\mathbf{y}}_{Algos}$, HbA1c and treatment history	0.81	0.003
$\hat{\mathbf{y}}_{Algos}$, HbA1c and treatment history, others	0.81	0.003

Improving the learned prescription algorithm

We next apply the constrained ℓ_1-regularized softmax regression model of Sec. 5.4 to improve the learned prescription algorithm.

To illustrate the effects of our recommended prescription algorithm, we consider the following three recommendation strategies, where the first one, rec_1, is based on the recommendation strategy from [Bertsimas et al., 2017] and is described in detail below, and

the other two are based on our algorithm improvement model.

(1) rec_1 (strategy in [Bertsimas et al., 2017]): For each sample $\mathbf{x}_i$, the recommended treatment is $u^{rec_1}(\mathbf{x}_i) = \arg\min_u \hat{y}_{t+1}(\mathbf{x}_i, u)$ only if the predicted future HbA1c $\hat{y}_{t+1}(\mathbf{x}_i, u^{rec_1}(\mathbf{x}_i))$ is significantly less than the current-period HbA1c $y_t(\mathbf{x}_i, u_{t-1}(\mathbf{x}_i))$; otherwise the prescribed treatment in the previous-period $u_{t-1}(\mathbf{x}_i)$ is being adopted. Specifically,

$$u_t^{rec_1}(\mathbf{x}_i) = \begin{cases} u_{t-1}(\mathbf{x}_i), \text{ if } \min\limits_u \hat{y}_{t+1}(\mathbf{x}_i, u) > \eta\ y_t(\mathbf{x}_i, u_{t-1}(\mathbf{x}_i)), \\ \arg\min_u \hat{y}_{t+1}(\mathbf{x}_i, u), \text{ otherwise,} \end{cases}$$

where η is a threshold regulating the conservativeness of the algorithm; it is set to $\eta = 95\%$ in our experiments.

(2) rec_2 (our strategy): As described in Sec. 5.4, for a sample $\mathbf{x}_i$ in the training set, the true next-period HbA1c $y_{t+1}(\mathbf{x}_i, u_t(\mathbf{x}_i))$ and the current-period treatment type $u_t(\mathbf{x}_i)$ are available. We first generate new current-period treatment labels based on the prescription effects predicted by regression methods. Specifically,

$$u_t^*(\mathbf{x}_i) = \begin{cases} u_t(\mathbf{x}_i), \text{ if } \min\limits_u \hat{y}_{t+1}(\mathbf{x}_i, u) > \eta\ y_{t+1}(\mathbf{x}_i, u_t(\mathbf{x}_i)), \\ \arg\min_u \hat{y}_{t+1}(\mathbf{x}_i, u), \text{ otherwise.} \end{cases}$$

This assignment, adopts $u_t^*(\mathbf{x}_i) = \arg\min_u \hat{y}_{t+1}(\mathbf{x}_i, u)$ as new labels only if the predicted HbA1c $\hat{y}_{t+1}(\mathbf{x}_i, u^*(\mathbf{x}_i))$ is significantly less than the true future HbA1c, otherwise the current-period prescribed treatment is being used.

Then, using the new labels $u_t^*(\mathbf{x}_i)$, we train a constrained ℓ_1-regularized softmax regression model using formulation (5.4), where the feature vector $\phi(\mathbf{z}_i)$ includes $\hat{\mathbf{y}}_{Algos}$, the HbA1c history, and the treatment history. Once we learn the parameters $\hat{\theta}^{\ell}$, the recommended treatment type is obtained from the maximum likelihood estimation in (5.5), that is, we set

$$u^{rec_2}(\mathbf{x}_i) = u^*(\mathbf{z}_i).$$

(3) rec_3 (a conservative variant of strategy rec_2): The recommended treatment $u^{rec_2}(\mathbf{x}_i)$ is adopted only if the predicted future HbA1c $\hat{\mathbf{y}}_{t+1}(\mathbf{x}_i, u^{rec_2}(\mathbf{x}_i))$ is significantly less than the current-period HbA1c, otherwise, the prescribed treatment of the previous period is being used. In mathematical terms,

$$u_t^{rec_3}(\mathbf{x}_i) = \begin{cases} u_{t-1}(\mathbf{x}_i), \text{ if } \hat{y}_{t+1}(\mathbf{x}_i, u_t^{rec2}(\mathbf{x}_i)) > \eta\, y_t(\mathbf{x}_i, u_{t-1}(\mathbf{x}_i)), \\ u_t^{rec_2}(\mathbf{x}_i), \text{ otherwise.} \end{cases}$$

Note that the recommendation strategy rec_1 only depends on the predicted prescription effects from the regression models, whereas the strategies rec_2 and rec_3 also rely on the prescription algorithm we learn from the data, which corresponds to some model fitting the way physicians make these prescription decisions.

Table 5.5: Relative HbA1c improvement for treatment-shifted patients by recommendations based on LASSO regression.

$u_{\mathbf{LA}}$	Orig.	$\bar{\mathbf{y}}_{LA}$	$\bar{\mathbf{y}}_{EN}$	$\bar{\mathbf{y}}_{RF}$	$\bar{\mathbf{y}}_{WkNN}$	Avg($\bar{\mathbf{y}}$)
rec_1	8.31	8.02	8.02	8.22	8.29	8.14
	(0)	(3.5%)	(3.4%)	(1.0%)	(0.2%)	(2.0%)
rec_2	8.63	7.79	7.80	7.87	7.90	7.84
	(0)	(9.74%)	(9.70%)	(8.9%)	(8.5%)	(9.2%)
rec_3	8.65	8.00	8.00	8.08	8.17	8.06
	(0)	(7.5%)	(7.5%)	(6.6%)	(5.6%)	(6.8%)

All three recommendation algorithms depend on the predicted prescription effects, hence, each regression model can result in three corresponding recommended prescription algorithms whose performance is evaluated in Tables 5.5–5.8. We compare the HbA1c effects of the three recommendation strategies on the patients whose recommended treatments are different from their original *current-period* treatments prescribed by physicians.

Tables 5.5–5.8 show the *relative* HbA1c improvement for patients whose treatment has been modified by recommendations based on the various regression models. In each of

Table 5.6: Relative HbA1c improvement for treatment-shifted patients by recommendations based on Elastic Net regression.

$u_{\mathbf{EN}}$	Orig.	$\bar{\mathbf{y}}_{LA}$	$\bar{\mathbf{y}}_{EN}$	$\bar{\mathbf{y}}_{RF}$	$\bar{\mathbf{y}}_{WkNN}$	Avg($\bar{\mathbf{y}}$)
rec_1	8.32	8.03	8.02	8.24	8.31	8.15
	(0)	(3.5%)	(3.6%)	(1.0%)	(0.1%)	(2.0%)
rec_2	8.65	7.81	7.81	7.88	7.92	7.85
	(0)	(9.7%)	(9.7%)	(8.9%)	(8.4%)	(9.2%)
rec_3	8.66	8.01	8.01	8.09	8.18	8.07
	(0)	(7.5%)	(7.6%)	(6.6%)	(5.6%)	(6.8%)

Table 5.7: Relative HbA1c improvement for treatment-shifted patients by recommendations based on Random Forest regression.

$u_{\mathbf{RF}}$	Orig.	$\bar{\mathbf{y}}_{LA}$	$\bar{\mathbf{y}}_{EN}$	$\bar{\mathbf{y}}_{RF}$	$\bar{\mathbf{y}}_{WkNN}$	Avg($\bar{\mathbf{y}}$)
rec_1	8.81	8.22	8.21	8.19	8.36	8.25
	(0)	(6.8%)	(6.8%)	(7.0%)	(5.1%)	(6.4%)
rec_2	8.82	7.91	7.91	7.90	7.99	7.93
	(0)	(10.3%)	(10.3%)	(10.5%)	(9.5%)	(10.1%)
rec_3	8.79	8.17	8.17	8.12	8.27	8.18
	(0)	(7.1%)	(7.1%)	(7.6%)	(5.9%)	(6.9%)

these tables: (a) the 2nd column shows the average future HbA1c of these patients if they follow the original physician's prescription; (b) columns 3–6 list the future HbA1c predicted by different regression models if the patients follow the recommended prescription (using strategies rec_1–rec_3); (c) the last column lists the average of these predicted treatment effects; (d) for each prescription strategy (rec_1–rec_3) we also list in parentheses the *relative* improvement in the next period HbA1c compared to the original prescription algorithm. Note that the listed relative improvement is an estimate of the true improvement based on the predicted treatment effect. Ideally, the true relative improvements can be obtained by comparing the effects of different treatments on actual patients, which can only be obtained from expensive long-term clinical trials.

As it can be seen from the results in Tables 5.5–5.8, recommendations rec_2 and rec_3

Table 5.8: Relative HbA1c improvement for treatment-shifted patients by recommendations based on WkNN regression.

$u_{\mathbf{WkNN}}$	Orig.	$\bar{\mathbf{y}}_{LA}$	$\bar{\mathbf{y}}_{EN}$	$\bar{\mathbf{y}}_{RF}$	$\bar{\mathbf{y}}_{WkNN}$	Avg($\bar{\mathbf{y}}$)
rec_1	7.80	7.69	7.68	7.77	7.42	7.64
	(0)	(1.4%)	(1.5%)	(0.3%)	(4.9%)	(2.0%)
rec_2	8.30	7.71	7.71	7.74	7.53	7.67
	(0)	(7.0%)	(7.1%)	(6.8%)	(9.3%)	(7.6%)
rec_3	8.14	7.75	7.74	7.77	7.54	7.70
	(0)	(4.8%)	(4.8%)	(4.5%)	(7.3%)	(5.4%)

consistently outperform rec_1, by as much as 10.1% in terms of the average (among various predictive models) future HbA1c values. This illustrates the efficiency of incorporating prescription learning models, rather than only considering the predicted prescription effects, in obtaining recommended prescriptions. It is also clear that rec_3, which is more conservative than rec_2, outperforms rec_1 but is inferior to rec_2 at least in terms of the predicted future HbA1c. Since we have not performed a clinical trial, it not possible to assess what will the true effect in HbA1c be, and it may be the case that the more conservative rec_2 may protect the model against regression inaccuracies; clearly this is not visible when we use the regression models to make predictions.

Table 5.9 shows the counts and ratios of shifted treatments under the three prescription algorithm recommendations (rows) based on various regression models (columns). Compared to rec_1, rec_3 (and rec_2 in most cases) has much fewer records where the recommendations are different from the physicians' prescriptions.

Analysis of the recommended prescriptions, including a manual review

To enable a more intuitive comparison of the original and our prescription algorithm, in Fig. 5·1 we analyze the distributions of patient treatments under the original prescription algorithm and our recommendation rec_3, when using the Elastic Net regression model. In the figure, "orig" denotes the original physicians' algorithm and "rec" our algorithm,

Table 5.9: Counts and Ratios of patients with shifted treatment.

	Shifted Treatments	$u_{\mathbf{LA}}$	$u_{\mathbf{EN}}$	$u_{\mathbf{RF}}$	$u_{\mathbf{WkNN}}$
rec_1	Counts	13,807	14,066	8,889	15,868
	(Ratio)	(36.57%)	(37.26%)	(23.54%)	(42.03%)
rec_2	Counts	12,258	12,479	12,185	14,299
	(Ratio)	(32.47%)	(33.05%)	(32.27%)	(37.87%)
rec_3	Counts	9,584	9,696	8,001	11,265
	(Ratio)	(25.38%)	(25.68%)	(21.19%)	(29.84%)

respectively. We first group all patient records by their current-period HbA1c values with cut-points 5.7%, 6.5%, 7.0%, and 9.0%. Then, for each treatment type and each HbA1c group, we plot the counts of patient records. For the first three groups where the patient's HbA1c is under control (less than 7.0%), the percentage of using no drugs increases while the percentage of using drugs decreases. For the 4th group, where the patient's HbA1c is high but not very severe (7.0%–9.0%), the percentage of using oral drugs increases, but that of using no drugs or injectable drugs decreases. For the 5th group, where HbA1c is severe (greater than 9.0%), the percentage of both oral and injectable agents increases. This is consistent with the acceptable guidelines for diabetes care [American Diabetes Association, 2018a].

In Fig. 5·2, we compare the mean and standard deviation of some normalized demographical, vital signs and lab test feature values for patients under oral or injectable agents in both the original and the recommended prescription algorithms. The patients whose recommended treatment type is injectable have more acute disease in terms of HbA1c and blood glucose, on average, compared to their counterparts under the original prescription algorithm. This suggests that the proposed prescription algorithm reserves the injectable agents (typically insulin) for the most severe diabetes cases.

As an additional sanity check of the proposed recommended prescriptions, two M.D.s manually reviewed a sample of 1,000 randomly selected cases. Out of these 1,000 cases,

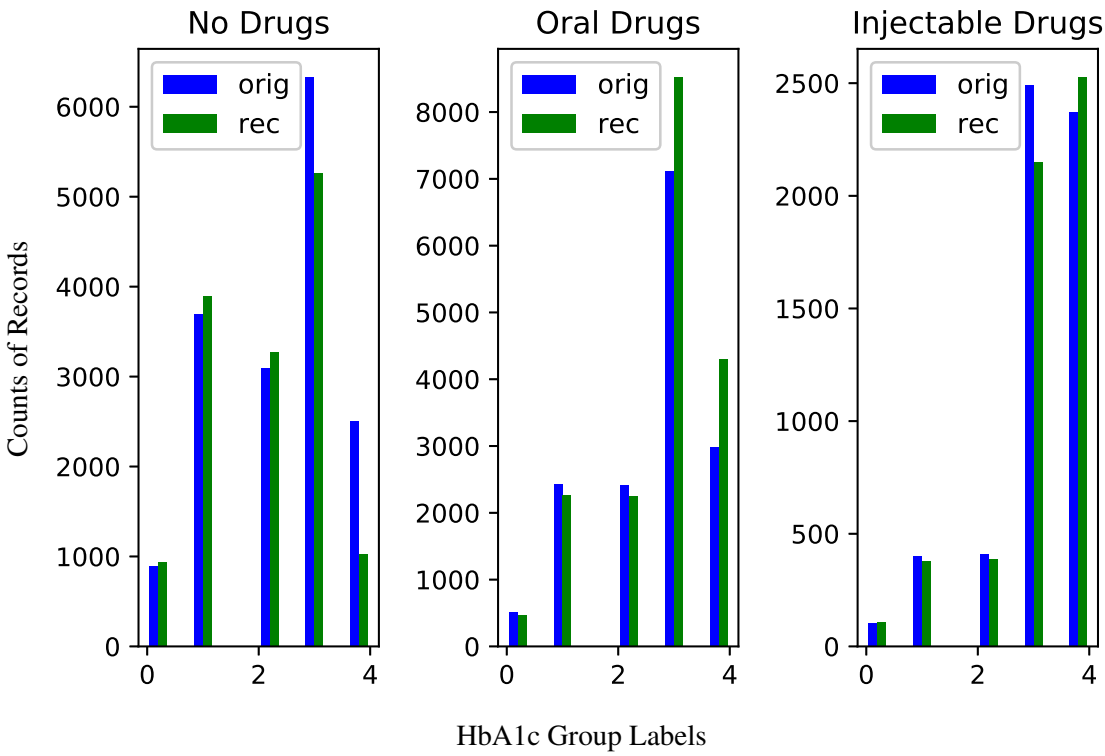

Figure 5·1: Comparison of original and recommended prescriptions under various HbA1c subgroups.

our recommended treatment is the same with the original prescription for 744 cases. For the remaining cases, the physicians agree with our prescription recommendations in 126 cases, while they agree with the original prescription in 98 cases. In total, the physicians agree with our recommendation for 870 out of 1,000 cases (87%). Furthermore, the physicians find that our recommendations are closer to the ideal prescription compared to the original prescription in the EHR in 907 cases out of 1000.

5.6 Conclusions

We have developed a data-driven framework for (1) learning a model of the prescription algorithm physicians use to treat patients with a chronic disease and (2) altering the parameters of this algorithm to improve future patient outcomes. Our framework employs

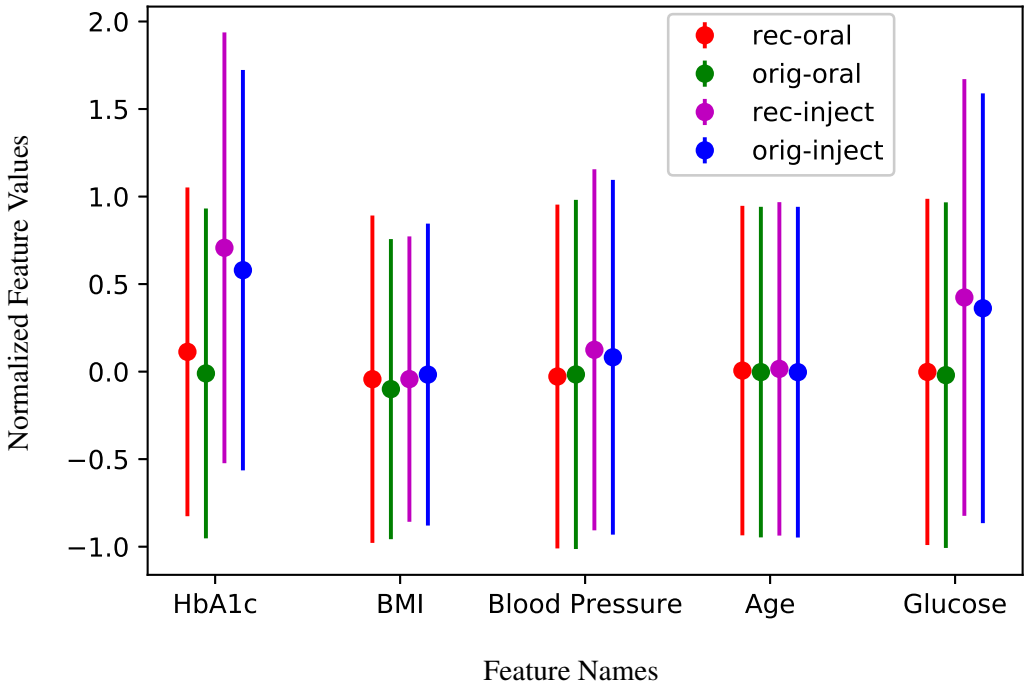

Figure 5·2: Mean and standard deviation of selected features from patients with oral or injectable prescriptions under the original and the recommended algorithm.

a combination of regression to predict future outcomes and classification to associate prescriptions with different patient conditions and characteristics. In both cases, we solve a robust learning problem which employs the use of appropriate regularizers.

The prescription algorithm we eventually obtain combines the goals of optimizing future predicted outcomes and incorporating information about what physicians typically do to treat a patient. Based on the experimental results we obtained, this leads to significant improvement over earlier approaches that only sought to optimize future outcomes. Specifically, we applied our framework to a large dataset ($n \approx 38{,}000$) containing prescriptions for type 2 diabetes patients. Compared to alternative approaches, our algorithm improves future HbA1c by as much as 10%, on average over various predictive models. In particular, the patients whose pharmacological treatment follows the recommended (optimized)

algorithm see a reduction in the average HbA1c from 8.65% to 8.06%. We note that this reduction is commensurate with the effect of some of the medications on the market to treat diabetes.

As a sanity check of the prescriptions recommended by the algorithm, two physicians evaluated a random 1000 cases. They were in agreement with the recommendations in 87% of the cases and in more than 90% of the cases they found the algorithm's recommendations closer to their ideal prescriptions than the original prescriptions in the patient record.

The framework we developed is not intended to replace primary care and specialty physicians who provide excellent care for their patients. There are, however, many patients who live in remote places and/or do not have access to specialists and consistent medical care. An algorithm like ours can be used to automatically process EHRs and make recommendations which can be compared to the ones on record (which may be non-existent). Substantial differences can be detected and lead to alerts, helping to put in place mechanisms to direct attention to these patients, improve their glycemic control, and help prevent costly hospitalizations and life-threatening complications. Our proposed framework can be easily adapted to improving personalized prescriptions for other chronic diseases beyond diabetes.

Chapter 6

Summary and Future Work

6.1 Summary

Early detection and treatment improvement can slow down the progression of the disease and result in better health outcomes and huge savings. Diabetes is the fastest growing chronic condition causing a number of preventable hospitalizations, and it is also associated with serious complications, such as heart disease and stroke, retinopathy, kidney failure, and lower-limb amputation.

The first objective of this research is to consider the problem of predicting diabetes-related hospitalizations in a target year using information in the EHRs of the patients. We develop a novel clustering and classification framework (ACC) that jointly discriminates between hospitalized and non-hospitalized patients and discovers clusters of patients with key factors, different in each cluster, that lead to hospitalization. The identification of the clusters has the significant advantage of interpretability, which is crucial in the medical domain. We have proved convergence of the new algorithm and established theoretical generalization guarantees. The experimental results, on both simulated and actual data from a healthcare setting, demonstrate the superiority of our approach compared to alternative approaches, in terms of prediction accuracy and discovery of interpretable clusters. With a 20% false alarm rate, we can correctly predict almost 81% of the diabetes-related hospitalized patients. The proposed algorithm has wider applicability and the potential to be applied to other medical case studies, helping, for example, discover cohorts of patients with similar underlying issues and devising cohort-specific predictive models.

In the second part, we design predictive models for estimating the success rate of IVF based on age, egg characteristics, sperm characteristics, reproductive hormone measurements, and lifestyle-related variables. We assess the sensitivity of the models to specific predictive variables. In the first-cycle IVF success rate prediction problem, our work achieves an AUC of 70.95%. Even using only the five most important predictors, the predictive model can achieve an accuracy (AUC) of 67.02%. Based on an analysis of the importance of the variables in the model, we found many of the most important predictors consistent with the existing literature; for example, "Egg & Embryo", "Age" and "Lab Tested Hormone Value" variables are the most influential for predicting the IVF success rate. In addition, for predicted non-pregnant subjects, we predicted whether the subjects have no embryos implanted due to embryo abnormalities, or are not pregnant despite implantation, achieving an AUC of 91.03%.

In the third part of this thesis, we investigate the problem of learning a policy and an associated transition probability model in an MDP based on demonstrations. We propose two regularized logistic regression models to estimate the parameterized conditional transition probabilities and original policy. Theoretical results are established for obtaining out-of-sample generalization bounds on the difference in target parameters and their estimates. In addition, we derive a bound on the regret of the policy estimates under the learned transition probability model. The proposed algorithms can be applied in many real world MDP estimation problems, such as mining Electronic Health Record data. The theoretical results are validated in a disease progression application. The numerical results show satisfactory performance of both the policy and transition dynamics and illustrate the benefits of using robust learning techniques. The learned conditional transition probabilities and the prescription policy are useful for analysis of chronic disease progression and drug effects.

In the final part, a prescription learning and improvement framework is proposed with an application to Type II diabetes. The framework employs a combination of treatment ef-

fect regression, learning of the physicians' prescription policy, and policy optimization. We designed a classification model which demonstrates the capability of accurately predicting the physicians' prescription policy using the patients' EHRs. By incorporating information from both the estimated current prescription policy and the treatment effect regression models, we provide an approach for improving the current prescription policy, which is robust to the influence of low-accuracy regression models and leads to improved outcomes in the experiments with real datasets. The proposed pharmacological therapy recommendations demonstrate better performance than the state-of-art deterministic algorithm. The proposed framework can be easily adapted to improving personalized prescriptions for other diseases.

6.2 Future Work

The proposed joint clustering and classification (JCC) formulation is a non-convex optimization problem, and the ACC algorithm only guarantees convergence to a local optimal solution. A potential extension is to formulate the JCC problem into a convex optimization problem, which can guarantee convergence to a global optimal solution.

When the disease state can be observed, our proposed MDP model can be used to characterize the progress of the disease. However, in many practical health care applications, certain disease states (such as severity) may not be observed, and the partially observable Markov Decision Process (POMDP) has the potential to model disease progression involving treatment (control) and hidden sequential disease states.

In addition to the problem of learning and improving prescription policies, obtaining the optimal prescription policy is a challenging and meaningful problem. One method is to learn the MDP model first, and then use the value-iteration or policy-iteration methods for optimization. Reinforcement learning (RL) is a more promising solution to solve such sequential decision problems and obtain the optimal policy when the disease state space is huge and the reaction feedback to the treatment is delayed.

References

Abadeh, S. S., Esfahani, P. M. M., and Kuhn, D. (2015). Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, pages 1576–1584.

Abdollahian, M. and Das, T. K. (2015). A MDP model for breast and ovarian cancer intervention strategies for BRCA1/2 mutation carriers. *IEEE journal of biomedical and health informatics*, 19(2):720–727.

American Diabetes Association (2003). Implications of the united kingdom prospective diabetes study. *Diabetes care*, 26(suppl 1):s28–s32.

American Diabetes Association (2018a). 8. pharmacologic approaches to glycemic treatment: Standards of medical care in diabetes—2018. *Diabetes Care*, 41(Supplement 1):S73–S85.

American Diabetes Association (2018b). Economic costs of diabetes in the us in 2017. *Diabetes Care*, 41(5):917–928.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

Asoh, H., Akaho, M. S. S., Kamishima, T., Hasida, K., Aramaki, E., and Kohro, T. (2013). An application of inverse reinforcement learning to medical records of diabetes treatment. In *ECMLPKDD2013 Workshop on Reinforcement Learning with Generalized Feedback*.

Bertsimas, D., Kallus, N., Weinstein, A. M., and Zhuo, Y. D. (2017). Personalized diabetes management using electronic medical records. *Diabetes care*, 40(2):210–217.

Bipartisan Policy Center (2012). What is driving us health care spending? america's unsustainable health care cost growth. `https://www.kff.org/wp-content/uploads/sites/2/2012/10/bpc_health_care_cost_drivers_brief_sept_2012.pdf`.

Bishop, C. M. (2006a). *Pattern recognition and machine learning*. Information science and statistics. Springer, New York.

Bishop, C. M. (2006b). *Pattern recognition and machine learning*. Springer.

Bjercke, S., Fedorcsak, P., Åbyholm, T., Storeng, R., Ertzeid, G., Oldereid, N., Omland, A., and Tanbo, T. (2005). IVF/ICSI outcome and serum LH concentration on day 1 of ovarian stimulation with recombinant FSH under pituitary suppression. *Human Reproduction*, 20(9):2441–2447.

Blank, C., Wildeboer, R. R., DeCroo, I., Tilleman, K., Weyers, B., de Sutter, P., Mischi, M., and Schoot, B. C. (2019). Prediction of implantation after blastocyst transfer in in vitro fertilization: a machine-learning perspective. *Fertility and Sterility*, 111(2):318–326.

Bousquet, O., Boucheron, S., and Lugosi, G. (2004). Introduction to statistical learning theory. In *Advanced lectures on machine learning*, pages 169–207. Springer.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). Classification and regression trees. belmont, ca: Wadsworth. *International Group*, 432:151–166.

Brisimi, T. S., Xu, T., Wang, T., Dai, W., Adams, W. G., and Paschalidis, I. C. (2018). Predicting chronic disease hospitalizations from electronic health records: An interpretable classification approach. *Proceedings of the IEEE*, 106(4):690–707.

Broekmans, F., Kwee, J., Hendriks, D., Mol, B., and Lambalk, C. (2006). A systematic review of tests predicting ovarian reserve and IVF outcome. *Human Reproduction Update*, 12(6):685–718.

Broer, S. L., van Disseldorp, J., Broeze, K. A., Dolleman, M., Opmeer, B. C., Bossuyt, P., Eijkemans, M. J., Mol, B.-W. J., Broekmans, F. J., on behalf of the IMPORT study group, Broer, S., van Disseldorp, J., Broeze, K., Dolleman, M., Opmeer, B., Anderson, R., Ashrafi, M., Bancsi, L., Caroppo, L. E., Copperman, A., Ebner, T., Eldar Geva, M., Erdem, M., Greenblatt, E., Jayaprakasan, K., Fenning, R., Klinkert, E., Kwee, J., Lambalk, C., La Marca, A., McIlveen, M., Merce, L., Muttukrishna, S., Nelson, S., Ng, H., Popovic-Todorovic, B., Smeenk, J., Tomás, C., Van der Linden, P., van Rooij, I., Vladimirov, I., Bossuyt, P., Eijkemans, M., Mol, B., and Frank, B. (2013). Added value of ovarian reserve testing on patient characteristics in the prediction of ovarian response and ongoing pregnancy: an individual patient data approach. *Human Reproduction Update*, 19(1):26–36.

Calderón, J. M. and Jaimes, L. G. (2018). A fuzzy control-based approach for the selection of health interventions. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE.

Casanova, R., Saldana, S., Chew, E. Y., Danis, R. P., Greven, C. M., and Ambrosius, W. T. (2014). Application of random forests methods to diabetic retinopathy classification analyses. *PLOS one*, 9(6).

Centers for Disease Control and Prevention and others (2017). *National diabetes statistics report, 2017*. Atlanta, GA. https://dev.diabetes.org/sites/default/files/2019-06/cdc-statistics-report-2017.pdf.

Centers for Disease Control and Prevention et al. (2018). 2016 assisted reproductive technology fertility clinic success rates report. https://www.cdc.gov/art/reports/2016/fertility-clinic.html.

Centres for Disease Control and Prevention (2019). National Center for Health Statistics – Infertility Statistics. Technical report.

Chen, R. and Paschalidis, I. (2019). Selecting optimal decisions via distributionally robust nearest-neighbor regression. In *Advances in Neural Information Processing Systems*, pages 748–758.

Chen, R. and Paschalidis, I. C. (2018). A robust learning approach for regression models based on distributionally robust optimization. *The Journal of Machine Learning Research*, 19(1):517–564.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Choi, B., Bosch, E., Lannon, B. M., Leveille, M.-C., Wong, W. H., Leader, A., Pellicer, A., Penzias, A. S., and Yao, M. W. (2013). Personalized prediction of first-cycle in vitro fertilization success. *Fertility and Sterility*, 99(7):1905–1911.

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J. (2016). Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cotch, M. F., Pastorek, J. G., Nugent, R. P., Hillier, S. L., Gibbs, R. S., Martin, D. H., Eschenbach, D. A., Edelman, R., Carey, J. C., Regan, J. A., et al. (1997). Trichomonas vaginalis associated with low birth weight and preterm delivery. *Sexually transmitted diseases*, 24(6):353–360.

Cover, T. A. and Thomas, J. A. (2006). *Elements of Information Theory*. John Wiley & Sons.

Dai, W., Brisimi, T. S., Xu, T., Wang, T., Saligrama, V., and Paschalidis, I. C. (2015). A joint clustering and classification approach for healthcare predictive analytics. In *Extended abstract in 2nd Workshop on Data Mining for Medical Informatics: Predictive Analytics (conjuncted with American Medical Informatics Association Annual Symposium)*.

Davis, J., Burnside, E. S., de Castro Dutra, I., Page, D., Ramakrishnan, R., Costa, V. S., and Shavlik, J. W. (2005). View learning for statistical relational learning: With an application to mammography. In *IJCAI'05: Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 677–683. https://dl.acm.org/doi/10.5555/1642293.1642402.

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654.

Delaney, A., Jensen, J. R., and Morbeck, D. (2012). Fertility Testing - How Laboratory Tests Contribute to Successful Infertility Treatments. *Clinical Laboratory News*.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

Eliasson, U., Heldal, R., Pelliccione, P., and Lantz, J. (2015). Architecting in the automotive domain: Descriptive vs prescriptive architecture. In *2015 12th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 115–118. IEEE.

ESHRE Guideline Group on Good Practice in IVF Labs, De los Santos, M. J., Apter, S., Coticchio, G., Debrock, S., Lundin, K., Plancha, C. E., Prados, F., Rienzi, L., Verheyen, G., et al. (2016). Revised guidelines for good practice in ivf laboratories (2015). *Human Reproduction*, 31(4):685–686.

Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Fu, Z., Robles-Kelly, A., and Zhou, J. (2010). Mixing linear svms for nonlinear classification. *IEEE Transactions on Neural Networks*, 21(12):1963–1975.

Fujimoto, V. Y., Luke, B., Brown, M. B., Jain, T., Armstrong, A., Grainger, D. A., Hornstein, M. D., for Assisted Reproductive Technology Writing Group, S., et al. (2010). Racial and ethnic disparities in assisted reproductive technology outcomes in the united states. *Fertility and sterility*, 93(2):382–390.

Geller, T., David, Y. B., Khmelnitsky, E., Ben-Gal, I., Ward, A., Miller, D., and Bambos, N. (2019). Learning health state transition probabilities via wireless body area networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Georga, E. I., Protopappas, V. C., Ardigo, D., Marina, M., Zavaroni, I., Polyzos, D., and Fotiadis, D. I. (2012). Multivariate prediction of subcutaneous glucose concentration in type 1 diabetes patients based on support vector regression. *IEEE journal of biomedical and health informatics*, 17(1):71–81.

Gianaroli, L., Magli, M. C., Gambardella, L., Giusti, A., Grugnetti, C., and Corani, G. (2013). Objective way to support embryo transfer: a probabilistic decision. *Human Reproduction*, 28(5):1210–1220.

Glujovsky, D., Pesce, R., Fiszbajn, G., Sueldo, C., Hart, R. J., and Ciapponi, A. (2010). Endometrial preparation for women undergoing embryo transfer with frozen embryos or embryos derived from donor oocytes. *Cochrane database of systematic reviews*, (1).

Goyal, A., Aprilia, E., Janssen, G., Kim, Y., Kumar, T., Mueller, R., Phan, D., Raman, A., Schuddebeurs, J., Xiong, J., et al. (2016). Asset health management using predictive and prescriptive analytics for the electric power grid. *IBM Journal of Research and Development*, 60(1):4–1.

Gu, Q. and Han, J. (2013). Clustered support vector machines. In *Proceedings of the sixteenth international conference on artificial intelligence and statistics*, pages 307–315.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

Györfi, L. and Kohler, M. (2007). Nonparametric estimation of conditional distributions. *IEEE Transactions on Information Theory*, 53:1872 – 1879.

Güvenir, H. A., Misirli, G., Dilbaz, S., Ozdegirmenci, O., Demir, B., and Dilbaz, B. (2015). Estimating the chance of success in IVF treatment using a ranking algorithm. *Medical & Biological Engineering & Computing*, 53(9):911–920.

Hafiz, P., Nematollahi, M., Boostani, R., and Jahromi, B. N. (2017). Predicting Implantation Outcome of In Vitro Fertilization and Intracytoplasmic Sperm Injection Using Data Mining Techniques. *International Journal of Fertility & Sterility*, (3).

Hanawal, M. K., Liu, H., Zhu, H., and Paschalidis, I. C. (2018). Learning policies for markov decision processes from data. *IEEE Transactions on Automatic Control*, 64(6):2298–2309.

Hannan, T. J. (1999). Detecting adverse drug reactions to improve patient outcomes. *International Journal of Medical Informatics*, 55(1):61–64.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

Hinneburg, A., Aggarwal, C. C., and Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In *26th Internat. Conference on Very Large Databases*, pages 506–515.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*, volume 398. John Wiley & Sons.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

International Diabetes Federation (2015). Diabetes Atlas. `www.diabetesatlas.org/component/attachments/?task=download&id=116`.

Iyengar, G. N. (2005). Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280.

Jain, Y. K. and Bhandare, S. K. (2011). Min max normalization based data perturbation method for privacy protection. *International Journal of Computer & Communication Technology*, 2(8):45–50.

Jayaprakasan, K., Pandian, D., Hopkisson, J., Campbell, B., and Maalouf, W. (2014). Effect of ethnicity on live birth rates after in vitro fertilisation or intracytoplasmic sperm injection treatment. *BJOG: An International Journal of Obstetrics & Gynaecology*, 121(3):300–307.

Jiang, H., Russo, C., and Barrett, M. (2006). Nationwide frequency and costs of potentially preventable hospitalizations, 2006: Statistical brief# 72. Rockville, MD: Agency for Health Care Policy and Research (US). `https://www.hcup-us.ahrq.gov/reports/statbriefs/sb72.jsp`.

Kayyali, B., Knott, D., and Van Kuiken, S. (2013). *The big-data revolution in US health care: Accelerating value and innovation*. Mc Kinsey & Company. `https://digitalstrategy.nl/wp-content/uploads/E2-2013.04-The-big-data-revolution-in-US-health-care-Accelerating-value-and-innovation.pdf`.

Kent, D., Banerjee, S., and Chernova, S. (2018). Learning sequential decision tasks for robot manipulation with abstract markov decision processes and demonstration-guided exploration. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE.

Khalifa, M. and Zabani, I. (2016). Utilizing health analytics in improving the performance of healthcare services: A case study on a tertiary care hospital. *Journal of infection and public health*, 9(6):757–765.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., and Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, 13:8–17.

Krishnapuram, B., Carin, L., Figueiredo, M. A., and Hartemink, A. J. (2005). Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):957–968.

La Marca, A., Sighinolfi, G., Radi, D., Argento, C., Baraldi, E., Artenisio, A. C., Stabile, G., and Volpe, A. (2010). Anti-Mullerian hormone (AMH) as a predictive marker in assisted reproductive technology (ART). *Human Reproduction Update*, 16(2):113–130.

Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, 9(1):1–20.

Lee, S.-I., Lee, H., Abbeel, P., and Ng, A. Y. (2006). Efficient L1 regularized logistic regression. In *AAAI*, volume 6, pages 401–408.

Lever, J., Krzywinski, M., and Altman, N. S. (2016). Points of significance: Classification evaluation. *Nature Methods*, 13(8):603–604.

Liu, Y.-Y., Ishikawa, H., Chen, M., Wollstein, G., Schuman, J. S., and Rehg, J. M. (2013). Longitudinal modeling of glaucoma progression using 2-dimensional continuous-time hidden markov model. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 444–451. Springer.

Liu, Z. et al. (2011). A method of svm with normalization in intrusion detection. *Procedia Environmental Sciences*, 11:256–262.

Ludwick, D. A. and Doucette, J. (2009). Adopting electronic medical records in primary care: lessons learned from health information systems implementation experience in seven countries. *International Journal of Medical Informatics*, 78(1):22–31.

Lukaszuk, K., Kunicki, M., Liss, J., Lukaszuk, M., and Jakiel, G. (2013). Use of ovarian reserve parameters for predicting live births in women undergoing in vitro fertilization. *European Journal of Obstetrics & Gynecology and Reproductive Biology*, 168(2):173–177.

Massachusetts General Hospital Fertility Center (2019). In vitro fertilization basics. `https://www.massgeneral.org/assets/MGH/pdf/obgyn/fertility/handbooks/mass-general-fertility-center-in-vitro-fertilization-basics.pdf`.

Mathur, S. and Sutton, J. (2017). Personalized medicine could transform healthcare. *Biomedical reports*, 7(1):3–5.

Matignon, L., Laurent, G. J., and Le Fort-Piat, N. (2006). Reward function and initial values: better choices for accelerated goal-directed reinforcement learning. In *International Conference on Artificial Neural Networks*, pages 840–849. Springer.

Niederberger, C., Pellicer, A., Cohen, J., Gardner, D. K., Palermo, G. D., O'Neill, C. L., Chow, S., Rosenwaks, Z., Cobo, A., Swain, J. E., Schoolcraft, W. B., Frydman, R., Bishop, L. A., Aharon, D., Gordon, C., New, E., Decherney, A., Tan, S. L., Paulson, R. J., Goldfarb, J. M., Brännström, M., Donnez, J., Silber, S., Dolmans, M.-M., Simpson, J. L., Handyside, A. H., Munné, S., Eguizabal, C., Montserrat, N., Izpisua Belmonte, J. C., Trounson, A., Simon, C., Tulandi, T., Giudice, L. C., Norman, R. J., Hsueh, A. J., Sun, Y., Laufer, N., Kochman, R., Eldar-Geva, T., Lunenfeld, B., Ezcurra, D., D'Hooghe, T., Fauser, B. C., Tarlatzis, B. C., Meldrum, D. R., Casper, R. F., Fatemi, H. M., Devroey, P., Galliano, D., Wikland, M., Sigman, M., Schoor, R. A., Goldstein, M., Lipshultz, L. I., Schlegel, P. N., Hussein, A., Oates, R. D., Brannigan, R. E., Ross, H. E., Pennings, G., Klock, S. C., Brown, S., Van Steirteghem, A., Rebar, R. W., and LaBarbera, A. R. (2018). Forty years of IVF. *Fertility and Sterility*, 110(2):185–324.e5.

Nilim, A. and Ghaoui, L. E. (2005). Robust control of markov decision processes with uncertain transition matrices. *Operations research*, 53(5):780–798.

Office of the National Coordinator for Health Information Technology (2016). Office-based Physician Electronic Health Record Adoption, Health IT Quick-Stat# 50. `dashboard.healthit.gov/quickstats/pages/physician-ehr-adoption-trends.php`.

Palep-Singh, M., Picton, H., Vrotsou, K., Maruthini, D., and Balen, A. (2007). South asian women with polycystic ovary syndrome exhibit greater sensitivity to gonadotropin stimulation with reduced fertilization and ongoing pregnancy rates than their caucasian counterparts. *European Journal of Obstetrics & Gynecology and Reproductive Biology*, 134(2):202–207.

Paternain, S., Bazerque, J. A., Small, A., and Ribeiro, A. (2018). Learning policies for markov decision processes in continuous spaces. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4751–4758. IEEE.

Pele, O., Taskar, B., Globerson, A., and Werman, M. (2013). The pairwise piecewise-linear embedding for efficient non-linear classification. In *Proceedings of the 30th international conference on machine learning*, pages 205–213.

Prasad, S., Kumar, Y., Singhal, M., and Sharma, S. (2014). Estradiol level on day 2 and day of trigger: a potential predictor of the ivf-et success. *The Journal of Obstetrics and Gynecology of India*, 64(3):202–207.

Pudil, P., Novovičová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125.

Purcell, K., Schembri, M., Frazier, L. M., Rall, M. J., Shen, S., Croughan, M., Grainger, D. A., and Fujimoto, V. Y. (2007). Asian ethnicity is associated with reduced pregnancy outcomes after assisted reproductive technology. *Fertility and Sterility*, 87(2):297–302.

Ram, S., Zhang, W., Williams, M., and Pengetnze, Y. (2015). Predicting asthma-related emergency department visits using big data. *IEEE journal of biomedical and health informatics*, 19(4):1216–1223.

Rouzbahman, M., Jovicic, A., and Chignell, M. (2017). Can cluster-boosted regression improve prediction of death and length of stay in the icu? *IEEE journal of biomedical and health informatics*, 21(3):851–858.

Sacks, D. B. (2012). Measurement of hemoglobin a1c: a new twist on the path to harmony. *Diabetes Care*, 35(12):2674–2680.

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS One*, 10(3):e0118432.

Schlegel, P. N. and Girardi, S. K. (1997). In vitro fertilization for male factor infertility. *The Journal of Clinical Endocrinology & Metabolism*, 82(3):709–716.

Scholkopf, B., Sung, K.-K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., and Vapnik, V. (1997). Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11):2758–2765.

Seli, E., editor (2011). *Infertility*. Wiley-Blackwell, Oxford, UK.

Shapiro, A. J., Darmon, S. K., Barad, D. H., Albertini, D. F., Gleicher, N., and Kushnir, V. A. (2017). Effect of race and ethnicity on utilization and outcomes of assisted reproductive technology in the usa. *Reproductive Biology and Endocrinology*, 15(1):44.

Smith, A. D. A. C., Tilling, K., Nelson, S. M., and Lawlor, D. A. (2015). Live-Birth Rate Associated With Repeat In Vitro Fertilization Treatment Cycles. *JAMA: The Journal of the American Medical Association*, 314(24):2654.

Sontag, E. D. (1998). VC dimension of neural networks. In *Neural Networks and Machine Learning*, pages 69–95. Springer.

Stratton, I. M., Adler, A. I., Neil, H. A. W., Matthews, D. R., Manley, S. E., Cull, C. A., Hadden, D., Turner, R. C., and Holman, R. R. (2000). Association of glycaemia with macrovascular and microvascular complications of type 2 diabetes (ukpds 35): prospective observational study. *BMJ: British Medical Journal*, 321(7258):405–412.

Sunkara, S. K., Rittenberg, V., Raine-Fenning, N., Bhattacharya, S., Zamora, J., and Coomarasamy, A. (2011). Association between the number of eggs and live birth in IVF treatment: an analysis of 400 135 treatment cycles. *Human Reproduction*, 26(7):1768–1774.

Takeda, H., Matsumura, Y., Nakajima, K., Kuwata, S., Zhenjun, Y., Shanmai, J., Qiyan, Z., Yufen, C., Kusuoka, H., and Inoue, M. (2003). Health care quality management by means of an incident report system and an electronic patient record system. *International Journal of Medical Informatics*, 69(2):285–293.

Templeton, A., Morris, J. K., and Parslow, W. (1996). Factors that affect outcome of in-vitro fertilisation treatment. *The Lancet*, 348(9039):1402–1406.

The Society for Assisted Reproductive Technology (SART) (2016). The SART Clinic Summary Report. Technical report.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley & Sons Inc., New York.

Vellido Alcacena, A., Martin Guerrero, J. D., and Lisboa, P. J. (2012). Making machine learning models interpretable. In *ESANN 2012: The 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Proceedings. Bruges (Belgium)*, pages 163–172.

Voigt, P. and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*.

Wang, J. and Sauer, M. V. (2006). In vitro fertilization (IVF): a review of 3 decades of clinical innovation and technological advancement. *Therapeutics and Clinical Risk Management*, 2(4):355–364.

Wang, S. J., Middleton, B., Prosser, L. A., Bardon, C. G., Spurr, C. D., Carchidi, P. J., Kittler, A. F., Goldszer, R. C., Fairchild, D. G., Sussman, A. J., et al. (2003). A cost-benefit analysis of electronic medical records in primary care. *The American Journal of Medicine*, 114(5):397–403.

Weiss, J., Natarajan, S., and Page, D. (2012). Multiplicative forests for continuous-time processes. In *Advances in Neural Information Processing Systems*, pages 458–466.

Wellons, M. F., Lewis, C. E., Schwartz, S. M., Gunderson, E. P., Schreiner, P. J., Sternfeld, B., Richman, J., Sites, C. K., and Siscovick, D. S. (2008). Racial differences in self-reported infertility and risk factors for infertility in a cohort of black and white women: The CARDIA Women's Study. *Fertility and Sterility*, 90(5):1640–1648.

Wen, M., Papusha, I., and Topcu, U. (2017). Learning from demonstrations with high-level side information. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*.

Wiesemann, W., Kuhn, D., and Rustem, B. (2013). Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183.

Xu, T., Brisimi, T. S., Wang, T., Dai, W., and Paschalidis, I. C. (2016). A joint sparse clustering and classification approach with applications to hospitalization prediction. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4566–4571. IEEE.

Xu, T. and Paschalidis, I. C. (2019). Learning models for writing better doctor prescriptions. In *2019 18th European Control Conference (ECC)*, pages 2454–2459. IEEE.

Xu, Y., Zhang, Y.-S., Zhu, D.-Y., Zhai, X.-H., Wu, F.-X., and Wang, A.-C. (2019). Influence of gnrh antagonist in reproductive women on in vitro fertilization and embryo transfer in fresh cycles. *Biomedical reports*, 10(2):113–118.

Yakowitz, S. et al. (1979). Nonparametric estimation of markov transition functions. *The Annals of Statistics*, 7(3):671–679.

Yoon, J., Davtyan, C., and van der Schaar, M. (2016). Discovery and clinical decision support for personalized healthcare. *IEEE journal of biomedical and health informatics*, 21(4):1133–1145.

Yu, Z., Xu, Z., Black, A. W., and Rudnicky, A. (2016). Strategy and policy learning for non-task-oriented conversational systems. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, pages 404–412.

Zarkogianni, K., Athanasiou, M., Thanopoulou, A. C., and Nikita, K. S. (2017). Comparison of machine learning approaches toward assessing the risk of developing cardiovascular disease as a long-term diabetes complication. *IEEE journal of biomedical and health informatics*, 22(5):1637–1647.

Zegers-Hochschild, F., Adamson, G., de Mouzon, J., Ishihara, O., Mansour, R., Nygren, K., Sullivan, E., and Vanderpoel, S. (2009). International Committee for Monitoring Assisted Reproductive Technology (ICMART) and the World Health Organization (WHO) revised glossary of ART terminology, 2009. *Fertility and Sterility*, 92(5):1520–1524.

Zhang, J., Liu, H., Mao, X., Chen, Q., Fan, Y., Xiao, Y., Wang, Y., and Kuang, Y. (2019). Effect of body mass index on pregnancy outcomes in a freeze-all policy: an analysis of 22,043 first autologous frozen-thawed embryo transfer cycles in china. *BMC medicine*, 17(1):114.

Zhao, Q., Edrich, T., and Paschalidis, I. C. (2014). Adaptive control of bivalirudin in the cardiac intensive care unit. *IEEE Transactions on Biomedical Engineering*, 62(2):638–647.

Zhu, H., Xu, T., and Paschalidis, I. C. (2019). Learning parameterized prescription policies and disease progression dynamics using markov decision processes. In *2019 American Control Conference (ACC)*, pages 3438–3443. IEEE.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

CURRICULUM VITAE

Tingting Xu

Education

Ph.D. Candidate in Systems Engineering — May 2020
Boston University, Boston, MA
Advisor: Professor Ioannis Ch. Paschalidis
Thesis: "Machine Learning for Effective Predictions & Prescriptions in Health Care"
Research interests: Machine Learning, Optimization Theory, Health Informatics

M.S. in Systems Modeling and Control Theory — Jul 2014
Chinese Academy of Sciences (CAS), Beijing, China

B.S. in Information and Computing Sciences — Jul 2011
China University of Mining & Technology, Beijing (CUMTB)

Selected Publications

- Xu, T., Zhu, H., & Paschalidis, I. C. Learning Parametric Policies and Transition Probability Models of Markov Decision Processes from Data, submitted.

- Xu, T., & Paschalidis, I. C. (2019, June). Learning models for writing better doctor prescriptions. In 2019 18th European Control Conference (ECC) (pp. 2454-2459). IEEE.

- Zhu, H., Xu, T., & Paschalidis, I. C. (2019, July). Learning Parameterized Prescription Policies and Disease Progression Dynamics using Markov Decision Processes. In 2019 American Control Conference (ACC) (pp. 3438-3443). IEEE.

- Brisimi, T. S., Xu, T., Wang, T., Dai, W., & Paschalidis, I. C. (2019). Predicting diabetes-related hospitalizations based on electronic health records. Statistical methods in medical research, 28(12), 3667-3682.

- Brisimi, T. S., Xu, T., Wang, T., Dai, W., Adams, W. G., & Paschalidis, I. C. (2018). Predicting chronic disease hospitalizations from electronic health records: an interpretable classification approach. Proceedings of the IEEE, 106(4), 690-707.

- Xu, T., Brisimi, T. S., Wang, T., Dai, W., & Paschalidis, I. C. (2016, December). A joint sparse clustering and classification approach with applications to hospitalization prediction. In 2016 IEEE 55th Conference on Decision and Control (CDC) (pp. 4566-4571). IEEE.

Teaching Experience

- Mentor for Research Internship in Science & Engineering Program 2018
- Teaching Fellow for Optimization Theory and Methods 2015

Awards & Services

- Grand Prize winner (1st place) in the MIT Policy Hackathon 2019
- 2nd place team winner in the cancer track of MIT Grand Hack 2019
- Certificate of achievement in Kaggle's Datathon of Women in Data Science 2019
- Moderator for "NECINA&MIT CHIEF 2018 Digital Health Conference" 2018
- Grace Hopper Celebration Student Scholarship 2018
- Outstanding Contributor in Reviewing for Knowlege-Based Systems 2017
- Distinguished Systems Engineering Fellowship at BU 2014-2015