Machine Learning for Risk Prediction and Privacy in Electronic Health Records

by

Eric Lantz

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2016

Date of final oral examination: 02/29/2016

The dissertation is approved by the following members of the Final Oral Committee:

C. David Page, Professor, Biostatistics and Medical Informatics
Mark Craven, Professor, Biostatistics and Medical Informatics
Jude Shavlik, Professor, Computer Sciences
Somesh Jha, Professor, Computer Sciences
Peggy Peissig, Center Director, BIRC, Marshfield Clinic

ACKNOWLEDGMENTS

There are many people I would like to thank for helping me throughout my graduate career.

I want to thank my advisor, David Page. David has been steadfast in his support for me throughout my tenure. His advice, mentorship, and feedback have improved the way I understand and approach my research. I am forever grateful to him for guiding me to this point and for his friendship.

I also want to thank the other members of my committee: Mark Craven, Jude Shavlik, Somesh Jha, and Peggy Peissig. I want to thank Mark and Jude for cultivating my interest in machine learning and providing expertise throughout my time here, Somesh for guiding me through my foray into privacy research, and Peggy for being an amazing collaborator and representative for Marshfield Clinic.

I would like to thank my friends and colleagues from my time in Madison. I want to thank my collaborators Kendrick Boyd, Jesse Davis, Matt Fredrickson, and Chen Zeng for working with me on the projects in this dissertation. I greatly appreciate the discussions and diversions from Aubrey Barnard, Finn Kuusisto, Jeremy Weiss, Bess Berg, Jie Liu, Deborah Chasman, Alex Cobian, Deborah Muganda-Rippchen, Ameet Soni, and the many other wonderful members of the machine learning community at UW over the years. And thanks to Vasily Dudnik and Laura Heideman for helping remind me of the world outside computer science.

I would like to acknowledge the funding sources that sustained me during my graduate work: CIBM Training Program through NIH grant 5T15LM007359, NIGMS grant R01GM097618, the Wisconsin Genomics Initiative, and NLM grant R01LM011028.

I need to thank my parents and siblings, Eric, Jane, Laura, and Stephen, for their support and love, and for being a source of stability in my life.

Finally, I owe the most to my wife Misty, who has seen me through challenging times and never stopped believing in me. Thank you for everything you have brought to my life.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

There are many opportunities for machine learning to impact clinical care. The practice of medicine generates lots of data about patients, from symptoms to diagnoses to treatments. And there is much to be gained from leveraging the data to improve outcomes. But medical data also has significant and unique privacy concerns. This dissertation focuses on aspects of two limitations of utilizing clinical data on a large scale: representation and privacy.

The data in the electronic health record is the result of interactions with the health care system. The frequency, reliability, and complexity of the records can differ wildly between patients. Through several case studies, I present different ways to represent and predict from patient data.

Medical records are sensitive data, and the privacy of the patients is an important concern. This is an issue whenever patient data is used, but becomes more important when we consider the possibility of collaborations between hospital systems. I examine the potential for differential privacy to ameliorate some of these concerns.

In the last several years, differential privacy has become the leading framework for private data analysis. It provides bounds on the amount that a randomized function can change as the result of a change in one record of a database. I examine how the trade-off between privacy and utility in differential privacy impacts a simulated clinical scenario. I also look at how to modify inductive logic programming to satisfy differential privacy. In both cases, achieving reasonable levels of privacy have significant impacts on model accuracy.

Lastly, I look at ways to expand the applicability of differential privacy. While previous works have examined differential privacy in the context of model creation, none have looked at evaluation. I present work demonstrating how to appropriately release area under the ROC curve and

average precision. There are circumstances in which existing differential privacy approaches are intractable. I present a solution to one of these cases: when a selection must be made from a large number of options. I show that the subsampled exponential mechanism preserves differential privacy while having a small theoretical penalty and often improved empirical results.

1 INTRODUCTION

Humans are very complex systems that we do not yet completely understand. A doctor is trained to choose a course of treatment based on the problems that the patient is presenting. But a drug or other intervention does not always work the same every time. Some patients need higher doses, others have bad reactions. It is difficult to determine an individual patient's short and long term risk of side effects or future disease. But new kinds of analysis have the potential to make more confident predictions, utilizing information that isn't readily apparent in the doctor's office and learning from more patients than one physician can see in a lifetime.

This is the promise of precision medicine, where treatment decisions take into account many factors beyond the list of current symptoms. While physicians currently do attempt to tailor treatments to each of their patients, they have limited information on how all the possible factors should impact their decision. If a patient has a history of asthma, does that change which medication they should receive for bronchitis? Or the dose? If so, by how much? These questions are difficult to answer, and require analyzing many patients to come to reliable conclusions.

Precision medicine is often presented in terms of the impact of genetic variants. While genetics are an important and emerging source of patient information, the use of other sources, like clinical, dietary, sensor, and biomarker data are possible. Clinical history is a particularly important component, and will be the primary focus in this dissertation. First, it has the most direct relationship with ongoing medical conditions and symptoms. Second, unlike other data sources that might become useful once they become more widely collected, clinical data already exist for a large portion of the population. The transition of this data to electronic form increases its utility for secondary analysis.

The importance of medical data to patient health is highly connected to

the related issue of medical data privacy. Information about our medical histories is one of the most sensitive personal data that must be stored and maintained by third parties[1]. We know that our own care is improved by providing medical history to our medical providers. There are a variety of regulations covering the storage and communication of medical data, requiring removal of identifiers like name and zip code before data release. But these techniques are vulnerable to an adversary with outside information that can be used to link up records.

We focus on a more rigorous vision of statistical privacy. Differential privacy is a privacy definition that has become popular in recent years, owing in part to its resistance to attacks based on outside information. Rather it deals with the requirement that each element in a dataset must have a small impact on the final results, regardless of the contents of that element. Because this is a worst-case analysis, it provides stronger guarantees than an analysis that based on average-case performance. However, it also means that it is a high bar for an algorithm to meet.

From a high level perspective, the work in this thesis tries to bring together two related threads. The first is to improve the performance of machine learning models on clinical data, taking into account the incompleteness of the source data. The second is to make algorithms that satisfy differential privacy more practical, finding a better operating point between privacy and utility. These threads connect in a vision of a system for providing accurate predictions for clinical tasks with privacy guarantees that increase our ability to share data and models between institutions.

1.1 Thesis Statement

Machine learning has tremendous potential applications in medical records, enabling automatic individualized risk and dose prediction. These tech-

[1] Financial data also falls into this category

niques must take into account the structure of electronic medical records and the incomplete view of patient health they provide. As the power of machine learning increases with more data, the importance of patient privacy also increases. While appropriate policies are a necessary part of a successful system, differential privacy offers a promising algorithmic guarantee on patient privacy. In order to be applied to medical data, differentially private methods must be analyzed in terms of their effect on privacy, computation, and utility in training and testing data.

This dissertation supports the following thesis:

> The application of machine learning for clinical data holds great promise, but presents unique privacy risks throughout the machine learning process. Differential privacy holds promise for mitigating these risks, but present methods seriously limit utility.

While some of the chapters deviate from the motivating example of electronic health records, they provide analysis of techniques based on differential privacy that are building blocks to developing practical methods with privacy guarantees. The overarching goal is to enable the learning of high performance predictive models for risk and dose prediction that advance the state of the art in medical practice while mitigating the privacy implications of taking part in a large study using sensitive personal information.

1.2 Dissertation Organization

The thesis chapters can be summarized as follows.

Chapter 2 contains background information on concepts used throughout the thesis, such as random forests and differential privacy.

Chapter 3 explores several examples of applying machine learning to tasks in the medical domain, exploring the difficulties and potentials in this type of data. As we discover genetic variants that influence treatment, we have the opportunity to produce a dosing algorithm that takes the variants into account. The first study presents work conducted as part of a large consortium working to help determine the proper dose of the anticoagulant warfarin (Consortium, 2009). Another advance in medical technology is the continued uptake and use of electronic health records (EHR)[2]. Moving patient information off of paper and into systems that can be thoroughly searched and stored in a structured manner has the potential to reveal information about the real-world practice of health care that would otherwise be impossible to obtain. The chapter presents three case studies that show how EHR data can predict medical outcomes. We look at the increased risk of myocardial infarction (MI, a type of heart attack) caused by a class of pain relievers (Cox-2 inhibitors). Then we revisit warfarin to look at producing a dosing algorithm from the electronic health records directly. We applied a similar procedure to predicting the onset of atrial fibrillation and flutter (AF/F) and associated morbidities. The model is not only useful in predicting the disease, but also points to risk factors that have not been previously explored.

Chapter 4 The previous studies have exposed the difficulties in expressing the data from electronic health records in a way that is amenable to use in standard machine learning algorithms. One problem is the thousands of codes for diagnoses, medications, procedures, and tests used in the records, many quite rarely. We adopt methods from natural language processing to produce embeddings from EHR data in an unsupervised fashion.

[2]The term "electronic medical record" or EMR is also commonly used. Sometimes the terms are used interchangeably, though sometimes their definitions differ slightly. For example, one might consider data from an exercise tracking armband part of an EHR but not an EMR.

The second major portion of this thesis deals with differential privacy, a recent framework for privacy-preserving computations. Differential privacy provides guarantees on the amount a computation can change as the result of a single record in a database. This thesis explores several projects that relate differential privacy to machine learning.

Chapter 5 looks at the effect of differential privacy in a realistic application of attempting to preserve genetic privacy, based on the warfarin dosing scenario from (Consortium, 2009). There are tradeoffs involved in privacy guarantees in relation to medical accuracy such that privacy can compromise patient outcomes.

Chapter 6 explores inductive logic programming, the machine learning algorithm used in (Davis et al., 2008) to predict heart attack from EHR data. ILP and its statistical variants have been used in other medical domains, including breast cancer (Kuusisto et al., 2013). We develop a variant of an inductive logic programming algorithm that preserves differential privacy, but show that it has serious scaling limitations.

Differential privacy not only applies to building models, but also to testing and validating them. Even if we use differentially private methods to learn a model, reporting the performance of the model can leak information. To our knowledge this has not been addressed or even observed until now. **Chapter 7** explores producing statistics such as the area under the receiver operating characteristic curve and average precision such that differential privacy is preserved. In **Chapter 8**, we show that existing mechanisms for ensuring differential privacy can become computationally infeasible when the space of choices are combinatorial in size. We create a new mechanism that works in these scenarios, and provide rigorous proofs of both its soundness and accuracy.

2.1 Electronic Health Records

After decades of existence on paper, medical records increasingly are being stored electronically. This shift has happened piecemeal, with different parts of the medical system adopting computerized recording of events over a period of more than thirty years. Systems for prescription ordering, imaging, testing, and other functions are becoming more integrated, tracking patients throughout their interactions with the healthcare system. Some health care providers were early adopters of EHR technology, while others were much later to fully adopt them. This fact, along with interoperability issues between different EHR vendors, have limited the ability to perform analyses on large groups of patients, particularly across systems.

Records serve dual purposes: the recording of clinical history to aid care over multiple patient visits, and the documentation of events for purposes of management and billing. This dual role has influenced the structure of current electronic systems. One important consequence has been a push towards standardization. There are several ontologies developed to standardize the types of diagnoses and procedures a patient might receive. These ontologies, along with similar efforts in drugs and laboratory tests, simplify the computerized analysis of EHRs.

However, it is important to not treat the EHR as a completely accurate representation of reality. For example, lab test results may vary due to lab conditions and personnel. There could be transcription errors when older paper charts are converted into digital format. Patients switch doctors and clinics over time, so a patient's entire clinical history is unlikely to reside in one database. A gap in a patient history could be due to lack of illness, transfer out of service area, or a patient-hospital interaction could have gone unreported. Furthermore, things like the use of over-the-counter

A.

PatientID	Gender	Birthday
P1	M	3/22/63

B.

PatientID	Date	Physician	Symptoms	Diagnosis
P1	1/1/01	Smith	palpitations	hypoglycemic
P1	2/1/03	Jones	fever, aches	influenza

C.

PatientID	Date	Lab Test	Result
P1	1/1/01	blood glucose	42
P1	1/9/01	blood glucose	45

D.

PatientID	SNP1	SNP2	…	SNP500K
P1	AA	AB		BB
P2	AB	BB		AA

E.

PatientID	Date Prescribed	Date Filled	Physician	Medication	Dose	Duration
P1	5/17/98	5/18/98	Jones	prilosec	10mg	3 months

Figure 2.1: A simplified hypothetical clinical database. Table **A** contains demographic information about each patient. Table **B** lists symptoms and disease diagnoses from patient visits. Table **C** contains lab test results. Table **D** has single nucleotide polymorphism (SNP) data for patients. Table **E** has drug prescription information.

drugs may not appear in the clinical history.

From a technical perspective, personalized medicine presents many challenges for machine learning and data mining techniques. After all, EHRs are designed to optimize ease of data access and billing rather than learning and modeling. Each type of data (e.g. drug prescription information, lab test results, etc.) is stored in a different table of a database. This can mean that pieces of information relevant to determining a patient's status may be spread across many tables, requiring a series of links through even more tables. For example, see Figure 2.1. Methods need to be flexible to the many types of information being provided, which can be temporal (such as disease diagnoses) or not (such as genetics or family history).

The EHR data used throughout this dissertation comes from our col-

laborators at Marshfield Clinic. Headquartered in Marshfield, WI, the Marshfield Clinic system consists of dozens of facilities located throughout north central Wisconsin. Marshfield has been on the forefront of clinical use of EHRs; their custom system has been in use for more than thirty years. The combination of long record history, system comprehensiveness within the service area, and a relatively low turnover population, make Marshfield an ideal partner for this research. Anonymized records were transferred to a secure data warehouse used for conducting research, with no patient records leaving Marshfield's network.

2.2 Medical Prediction

Forecasting is as old as medicine, as doctors try to determine which interventions will be most beneficial to each patient. Beyond the guidelines for clinical care, clinical trials exist to determine the efficacy of a particular treatment. In clinical practice, doctors often follow clinical prediction rules, which provide guidelines to vary treatment based on current conditions. In this thesis, medical prediction refers to the more precise task of producing an algorithm that makes a numeric statement about an individual patient based on data from patients with similar features.

Simple prediction algorithms are often designed to be relatively easy to remember and calculate by hand. For example, the Glasgow coma score is a score between 3 and 15 based on a rubric of assessments of eye, motor, and verbal function. The Framingham risk score predicts heart attack risk by assigning points based on age, cholesterol, smoking, and blood pressure. The Ranson criteria for pancreatitis is a count of which of 11 measurements are out of normal range.

But models can become more complex, utilizing statistical and machine learning techniques. This complexity often brings improved accuracy, but it becomes more difficult to integrate predictions into clinical practice. This

motivates some of the work in this dissertation, as models acting directly on the EHR are more amenable to existing workflows than models that assume all measurements are taken concurrently.

When applying machine learning to different medical tasks, it is important to recognize that there is considerable variation in the "predictability" of the tasks. While the best possible results are unknown, some conditions are more deterministic and thus easier to predict. Comparison of different models by different researchers is difficult, as the data cannot be easily shared. The sometimes conflicting goals of model accuracy and patient privacy motivate later chapters on private learning.

2.3 Machine Learning Models

The studies in this thesis utilize a wide variety of methods for supervised machine learning. Most of the methods are explained in their individual chapters, but two are revisited in multiple chapters, so it is worthwhile to explain them here.

2.3.1 Random Forests

Random forests build on two more basic machine learning techniques: decision trees and bagging. They are an example of a larger class of ensemble methods, where multiple machine learning models are combined to improve performance.

Decision trees attempt to separate the data into subsets based on feature values. This separation is done recursively, dividing the data into smaller and smaller subsets until some termination condition is reached (ideally the subset contains only examples of a single class). At each node, a scoring criteria is used to decide how to split the data. While alternatives exist (Menze et al., 2011), this typically involves selecting the single feature that best separates the data according to class (or regression value). In

a binary decision tree, all splits are binary tests of each example (e.g. $x_1 = \mathrm{True}$ or $x_2 > 5$). If an example satisfies this condition, it moves down the left branch of the tree. Otherwise, it continues down the right branch. Trees have the advantage of being quite interpretable, and are capable of using data with a mix of continuous and discrete features. However, they are quite prone to overfitting without appropriate pruning to limit how large the tree becomes.

Bagging is a technique based on the idea that any particular dataset is a sample from a much larger distribution of possible examples. We might like to draw multiple datasets from this distribution, and run our machine learning algorithm on each one. In practice this is rarely possible, so bagging tries to simulate the behavior by producing variants of the original training set. If the dataset has n examples, create a new one that consists of n examples drawn from the original with replacement (i.e. each example can be drawn multiple times). This has the effect of putting random poisson-distributed weights on the original dataset, modifying the marginal distributions of the features. Each time this is repeated, the resulting machine learning models may give different answers, so voting or averaging is used to produce an ensemble prediction.

Random forests combine both these techniques, along with one more source of randomness. At each split point, only a randomly chosen fraction of the features are possible to be selected. This causes the algorithm to make slightly suboptimal decisions, but creates diversity among the different bagging models.

Random forests have been widely used for their excellent performance and ease of use. They can handle correlated and redundant features better than many linear models, and don't require feature preprocessing or normalization. The interpretability of decision trees is lost somewhat due to the potentially large ensemble, but the resulting models enjoy additional robustness.

2.3.2 ILP

Inductive logic programming uses first-order logical formalisms to represent examples and background knowledge. A binary classification model in ILP is a hypothesis that entails the positive examples and does not entail the negative examples. The hypothesis consists of clauses that form a generalized subset of the clauses in the data.

Given a set of positive and negative examples and background knowledge about the examples, an ILP algorithm conducts a search in the hypothesis space via inverse entailment. The hypothesis space contains each of the fully grounded examples, i.e. all the background knowledge relevant to each example. The search also includes clauses where the groundings are replaced by variables, making them more general and thus potentially entailing more examples.

More technically, let $M^+(T)$ be the minimal Herbrand model of a definite clause T. The problem of inductive logic programming is formulated in Definition 2.1.

Definition 2.1. *(Inductive logic programming (Muggleton and de Raedt, 1994)* [1]*): Given two languages,*

- $\mathcal{L}_1$*: the language of database (examples).*
- $\mathcal{L}_2$*: the language of hypotheses.*

Given a consistent set of background knowledge $B \subseteq \mathcal{L}_1$*, find a hypothesis* $H \in \mathcal{L}_2$*, such that:*

1. *Validity:* $\forall h \in H$, h *is true in* $M^+(B)$.
2. *Completeness: if general clause* g *is true in* $M^+(B)$*, then* $H \models g$
3. *Minimality: there is no proper subset* G *of* H *which is valid and complete*

[1] This formulation uses non-monotonic semantics.

ILP hypotheses make for relatively interpretable models, and readily handle data that is not structured in a single table. However, they face difficulties when the hypothesis space is large or the relationships are probabilistic in nature.

2.4 Differential Privacy

Approaches to data privacy in medicine have traditionally consisted of two primary techniques: suppression and generalization. Suppression simply refers to removing elements from the dataset. This can mean removing an entire column (like social security numbers) or rare values (8 foot tall patients). Generalization is combining values in a set or range (e.g. changing $age = 43$ to $age = 40 - 49$). The decisions of when to use these techniques were often ad-hoc.

Privacy models such as k-anonymity (Sweeney, 2002) attempted to formalize the degree to which suppression and generalization should be employed on a dataset. However, these methods were still shown to be vulnerable to attacks when an adversary had outside information (Li et al., 2012). Differential privacy was developed to take a different perspective on privacy preservation. Rather than focus on the elements in the data, differential privacy bounds the results of calculations on the data.

In differential privacy, the presence or absence of a record in the database is guaranteed to have a small effect on the output of an algorithm. As a result, the amount of information an adversary can learn about a single record is limited. For any databases $D, D' \in \mathbb{D}$, let D and D' be considered neighbors if they differ by exactly one record (denoted by $D' \in nbrs(D)$). Differential privacy requires that the probability that an algorithm outputs the same result on any pair of neighboring databases is bounded by a constant ratio.

Definition 2.2. *(ϵ, δ-differential privacy (Dwork et al., 2006)): For any input*

database $D \in \mathbb{D}$*, a randomized algorithm* $f : \mathbb{D} \to \mathcal{Z}$ *where* $\mathcal{Z} = \mathtt{Range}(f)$ *is* ϵ*-differentially private iff for any* $\mathcal{S} \subseteq \mathcal{Z}$ *and any database* $D' \in \mathtt{nbrs}(D)$*.*

$$\Pr(f(D) \in \mathcal{S}) \leqslant e^{\epsilon} \Pr(f(D') \in \mathcal{S}) + \delta \tag{2.1}$$

In the special case where $\delta = 0$, the stronger guarantee of ϵ-differential privacy is met.

Mechanisms for ensuring differential privacy rely on the *sensitivity* of the function we want to privatize. Sensitivity is the largest difference between the output on any pair of neighboring databases.

Definition 2.3. *(Sensitivity (Dwork et al., 2006)): Given a function* $f : \mathbb{D} \to \mathbb{R}^d$*, the sensitivity (*$\Delta$*) of* f *is:*

$$\Delta_f = \max_{D' \in \mathtt{nbrs}(D)} \|f(D) - f(D')\|_1 \tag{2.2}$$

A sequence of differentially private computations also ensures differential privacy, but for a different value of ϵ. This is called the composition property of differential privacy as shown in Theorem 2.4.

Theorem 2.4. *(Composition (Dwork et al., 2006)): Given a sequence of computations* $\mathbf{f} = f_1, \ldots, f_d$*, with* f_i *meeting* ϵ_i*-differential privacy, then* $\mathbf{f}$ *is* $(\sum_{i=1}^{d} \epsilon_i)$*-differentially private.*

When the outcome domain is real-valued, it is possible to add noise directly to the non-private value. Using noise drawn from the Laplace distribution (sometimes called the double exponential distribution) to perturb any real-valued query gives the following result:

Theorem 2.5. *(Laplace noise (Dwork et al., 2006)): Given a function* $f : \mathbb{D} \to \mathbb{R}$*, the computation*

$$f'(D) = f(D) + \text{Laplace}\left(\frac{\Delta_f}{\epsilon}\right) \tag{2.3}$$

guarantees ϵ*-differential privacy.*

The geometric mechanism is a discrete variant of the Laplacian mechanism. Ghosh et al. Ghosh et al. (2009) propose the *geometric mechanism* to guarantee ϵ-differential privacy for a single count query. The geometric mechanism adds noise Δ drawn from the two-sided geometric distribution $G(\epsilon)$ with the following probability distribution: for any integer σ,

$$\Pr(\Delta = \sigma) \sim e^{-\epsilon|\sigma|} \tag{2.4}$$

For domains that are not real-valued, the exponential mechanism can be used to select among outputs.

Theorem 2.6. *(Exponential mechanism (McSherry and Talwar, 2007)): Given a quality function* $q : (\mathbb{D} \times \mathcal{Z}) \to \mathbb{R}$ *that assigns a score to each outcome* $z \in \mathcal{Z}$ *for a given database, and base measure* μ *over* $\mathcal{Z}$*, a randomized algorithm,* $M : \mathbb{D} \to \mathcal{Z}$*, that outputs* z^* *with probability*

$$\Pr(M(D) = z^*) = \frac{e^{\epsilon\, q(D,z^*)}\mu(z^*)}{\int_{\mathcal{Z}} e^{\epsilon\, q(D,z)}\mu(z')dz} \tag{2.5}$$

is $2\epsilon\Delta_q$*-differentially private.*

These theorems provide a toolbox that can be applied to create algorithms that preserve differential privacy. While differential privacy can apply to producing noisy versions of the original data, the applications in this dissertation focus on other kinds of "queries". In particular, we are interested in machine learning models that do not leak information about the training data.

3 MEDICAL RISK AND DOSE PREDICTION

This chapter contains four case studies in medical prediction. The first is the prediction of heart attack among patients taking COX-2 inhibitors, a type of pain reliever. The method presented is SAYU, a statistical variant of ILP. The second is a large multinational study to determine patient-specific doses for warfarin, an anti-coagulant. Compared to the other case studies, each patient has a small, standardized set of features, including genetic markers. Warfarin as an example is also analyzed in later chapters.

Next, we look at predicting warfarin dose from the EHR directly. Lastly, prediction of atrial fibrillation and flutter, another cardiovascular condition. In both cases a similar random forest method is shown to be superior. The lessons from these case studies influence the latter privacy and future work.

Versions of these studies were published in Davis et al. (2008); Consortium (2009), and Lantz et al. (2015b).

3.1 Predicting Myocardial Infarction for patients on Cox-2 Inhibitors

3.1.1 Cox-2 inhibitors

Non-steroidal anti-inflammatory drugs, known as NSAIDs, are used to treat pain and inflammation. NSAIDs work by blocking cyclooxygenase (Cox), an enzyme responsible for the formation of prostanoids. Prostanoids are a class of molecules important in vasoconstriction and inflammation. There are three variations of the Cox molecule: Cox-1, Cox-2, and Cox-3. While all versions of Cox are involved in the same general process, there are slightly different effects when they are inhibited by medications. Non-selective NSAIDs, such as Aleve™and Advil™,

inhibit both Cox-1 and Cox-2. While these medications can effectively alleviate pain, prolonged use may result in gastrointestinal problems as a consequence of blocking the Cox-1 pathway (Simmons et al., 2004). The selective Cox-2 inhibitor hypothesis is that a drug that only (i.e., selectively) blocks the Cox-2 pathway will have the same benefits of traditional NSAIDs, while eliminating the side effects. It was confirmed that selective Cox-2 inhibitors resulted in fewer gastrointestinal side effects than their non-selective cousins. To this end, drugs such as Vioxx™, Bextra™and Celebrex™were introduced to the American drug market between 1998 and 2001. They were widely prescribed, resulting in several billion dollars in sales in the following years. However, they became implicated in cardiac side effects, roughly doubling the risk of myocardial infarction (MI) (Kearney et al., 2006). Beginning in 2004, Vioxx™and Bextra™were removed from the market due to these risks, and Celebrex™was restricted.

3.1.2 Data

Our data comes from Marshfield Clinic, an organization of hospitals and clinics in northern Wisconsin. This organization has been using electronic health records since 1985 and has electronic data back to the early 1960's. Furthermore, it has a reasonably stationary population, so clinical histories tend to be very complete. The database contained several thousand patients who had taken Cox-2 inhibitors, 492 of whom later had an MI. From the non-MI group, we subsampled 650 patients for efficiency reasons. We included information from four separate tables: lab test results (e.g. cholesterol levels), medications taken (both prescription and non-prescription), disease diagnoses, and observations (e.g. height,weight and blood pressure).

3.1.3 Methods

The goal of our retrospective case study can be defined as follows:

- Given: Patients who received Cox-2 inhibitors and their clinical history as found in their electronic health record
- Do: Predict whether the patient will have a myocardial infarction

The task of predicting which patients went on to have an MI after being prescribed Cox-2 inhibitors is an analog to the task of determining which patients should have not been prescribed them in the first place. There is always an underlying risk of MI in any population. The algorithm will predict MI for some subset of the patients. The goal is that the rate of MI in the remaining Cox-2 patients should be reduced to the rate of the population as a whole. At that point, the remaining patients receiving Cox-2 inhibitors would not have a greater risk of MI than they did before taking the drug.

One of the central issues we needed to address for our case study is what data should we include in our analysis. The first attempt might be to just exclude data after a patient on Cox-2 inhibitors has an MI. But this method still raises important issues. We will have uniformly more data for the non-MI cases, which introduces a subtle confounding factor in the analysis. For example, consider a drug recently introduced into the market. Under this scheme, patients on selective Cox-2 inhibitors who had MIs before the drug came on the market could not have taken the drug. Thus, it could introduce a spurious correlation between taking the drug and not having an MI.

Consequently, it is necessary to cut off data for each patient at the first Cox-2 prescription. This is also more in line with our idea of how the algorithm would be used in practice to disqualify certain patients from taking Cox-2 inhibitors at all.

The structured nature of patient clinical histories represents several problems for standard machine learning algorithms. In addition to the problems of multiple tables, the rows within a single table can be related. For example, the diagnosis table will contain one entry, or row, for each disease that a patient has been diagnosed with over the course of his life.

Propositionalization is common technique for converting relational data into a suitable format for standard learners (Lavrač and Džeroski, 1992; Pompe and Kononenko, 1995). Such work re-represents each example as a feature vector, and then uses a feature-vector learner to produce a final classifier. If the temporal aspect of the data is utilized in feature construction, the number of possible features becomes very large. Consequently, we created the following propositional features:

- One binary feature for each lab test, which is true if the the patient ever had that lab test.
- One binary feature for each drug, which is true if the patient ever took the drug.
- One binary feature for each diagnosis, which is true if the patient had this disease diagnosis.
- Three aggregate features (min, max, avg) for each type of observation. E.g. the patient's highest blood pressure.

This resulted in 3620 features.

3.1.4 Evaluation

The primary objective of the empirical evaluation is to demonstrate that machine learning techniques can predict which patients on Cox-2 inhibitors are substantial risk for MI. We tried many different types of algorithms, which can be divided into (i) propositional learners, (ii) relational learners, and (iii) statistical relational learners.

We looked at a wide variety of feature vector learners. From Weka, we used naïve Bayes and linear SVM (Witten and Frank, 2005) as well as our own implementation of tree-augmented naïve Bayes (Friedman et al., 1997). Additionally, we used decision trees, boosted decision trees and boosted rules algorithms from the C5.0 (Quinlan, 1987) package. The disadvantage of using these techniques is that they require propositionalizing the data, so we must collapse the data into a single table.

Relational learning allows us to directly operate on the multiple relational tables (Lavrač and Džeroski, 2001). We used the inductive logic programming (ILP) system Aleph (Srinivasan, 2001), which learns rules in first-order logic. ILP is appropriate for learning in multi-relational domains because the learned rules are not restricted to contain fields or attributes for a single table in a database. As introduced in Section 2.3.2, the ILP learning problem can be formulated as follows:

- Given: background knowledge B, set of positive examples E^+, set of negative examples E^- all expressed in first-order definite clause logic.
- Learn: A hypothesis H, which consists of definite clauses in first-order logic, such that $B \wedge H \models E^+$ and $B \wedge H \not\models E^-$.

In practice, it is often not possible to find either a pure rule or rule set. Thus, ILP systems relax the conditions that $B \wedge H \models E^+$ and $B \wedge H \not\models E^-$. ILP offers another advantage in that domain experts are easily able to interpret the learned rules. However, it does not allow us to represent uncertainty except by tuning set error rate.

Statistical relational learning (SRL) combines statistical and rule learning to represent uncertainty in structured data. We used the SAYU system, which combines Aleph with a Bayesian network structure learner (Davis et al., 2007a). SAYU uses Aleph to propose features to include in the Bayesian network. Aleph passes each rule it constructs to SAYU, which

converts the clause to a binary feature and adds it to the current probabilistic model. Next, SAYU learns a new model incorporating the new feature, and evaluates the model. If the model does not improve, the rule is not accepted, and Aleph constructs the next clause. In order to decide whether to retain a candidate feature f, SAYU needs to estimate the generalization ability of the model with and without the new feature. SAYU does this by calculating the area-under the ROC curve (AUC) on a tuning set. By retaining Aleph, SAYU also offers the advantage that the constructed features are comprehensible to domain experts.

We performed ten-fold cross validation. For SAYU, we used five folds as a training set and four folds as a tuning set. We used tree-augmented naïve Bayes as the probabilistic learner. We scored each candidate feature using AUC. A feature must improve the tune set AUC by two percent to be incorporated into the model. Additionally, we seeded the initial SAYU model with the 50 highest scoring propositional features by information gain on that fold (Davis et al., 2007a).

Table 3.1: Average area under the ROC curve (AUC) for the four best methods.

Naïve Bayes	TAN	Boosted Rules	SAYU-TAN
0.7428	0.7470	0.7083	0.8076

Figure 3.1 shows the ROC curves for the four best methods. All methods do better than chance on this task. Table 3.1 shows the average AUC for each method. SAYU clearly dominates the proposition learners for false positive rates less than 0.75. To assess significance, we performed a paired t-test on the per-fold AUC for each method. No significant difference exists between any of the propositional methods. Aleph, the relational method did extremely poorly and is not shown here. SAYU did significantly better than all the propositional methods.

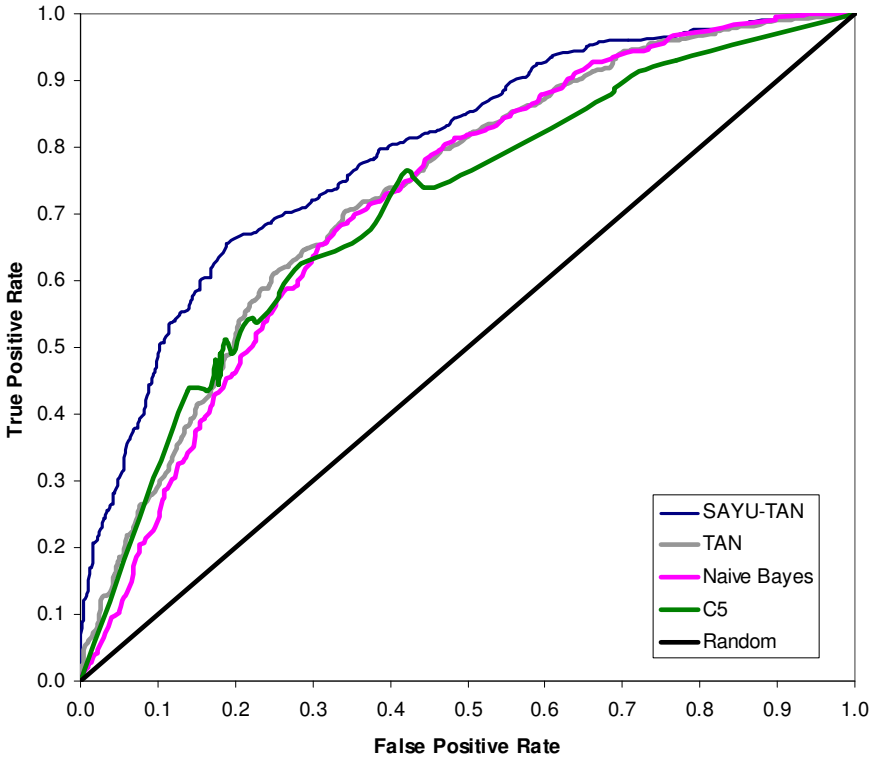


Figure 3.1: ROC Curves for the four best performing methods

3.1.5 Discussion

However, this limited case study suffers from several drawbacks. Namely, is our model predicting predisposition to MI?

Several other issues exist which are much harder to quantify. In particular, it is highly likely that the training data contains false negatives. For example, some patients may have a future MI related to Cox-2 use.

We found that using techniques from SRL lead to improvements over both standard propositional learning techniques as well as relational learning techniques. On this task, the ability to simultaneously handle multiple relations and represent uncertainty appears to be crucial for good performance.

SAYU-TAN mostly likely confers a benefit by integrating the feature induction and model construction into one, coherent process. This confirms results (Landwehr et al., 2005, 2006; Davis et al., 2005, 2007b) that have empirically demonstrated that an integrated approach results in superior performance compared to the traditional propositionalization framework of decoupling these two steps.

The ability to learn a comprehensible model is very important for our collaborators and the medical community in general. Preserving this property while developing the methods is important.

3.2 Predicting Warfarin Dose in Multicenter Consortium

3.2.1 Warfarin

Warfarin, with brand name Coumadin™, is the most prescribed anticoagulant medication, used chronically for conditions such as deep vein thrombosis, cardiac valve replacement, and atrial fibrillation (Glurich et al., 2010). By reducing the tendency of blood to clot, at appropriate dosages it can reduce risk of clotting events, particularly stroke. Since the introduction of warfarin as a medication in 1954, other anticoagulants have been developed, but warfarin remains popular due to its low cost and well-studied therapeutic effects. It acts as an antagonist of vitamin K, a crucial cofactor in the clotting process. The primary reason warfarin is not utilized more for the maintenance of conditions like atrial fibrillation is its position as a leading cause of adverse drug events, primarily bleeding.

It is also one of the most difficult medications to determine the proper dose for a patient. Proper dosages can range more than 10 fold within a population. Underestimating the dose can result in adverse events from the condition the drug was prescribed to treat, such as cardiac arrhythmia.

Overestimating the dose can, just as seriously, lead to uncontrolled bleeding events. Because of these risks, patients starting on warfarin typically must visit a clinic many times over the days and weeks of initiation before a stable dose is found. Warfarin is one of the leading causes of drug-related adverse events in the United States (Kim et al., 2009). Determining a patient's dose before initiation thus has the possibility of both reducing adverse events for the patient and reducing the number of clinic visits required to reach a stable dose.

The effect of warfarin is clinically determined by measuring the time it takes for blood to clot, called prothrombin time. This measure is standardized as an international normalized ratio (INR). Based on the patient's indication for warfarin, a clinician determines a target INR range. After initiation, the dose is modified until the desired INR range is reached and maintained.

Genetic variability among patients is known to play an important role in determining the dose of warfarin that should be used when oral anticoagulation is initiated. Polymorphisms in two genes, VKORC1 and CYP2C9, are associated with the mechanism with which the body metabolizes the drug, which in terms affects the dose required to reach a given concentration in the blood. VKORC1 encodes for vitamin K epoxide reductase complex. Warfarin is a vitamin K antagonist. CYP2C9 encodes for a variant of cytochrome P450, a family of proteins which oxidizes a variety of medications. A review of this literature is given in Kamali and Wynne (2010). Practical methods of using a patient's genetic information had not previous to the following study been evaluated in a diverse and large population. The International Warfarin Pharmacogenetics Consortium (Consortium, 2009) was formed in order to collect genetic and clinical information from a large, geographically diverse sample of patients and formulate an algorithm to improve initiation dosing of warfarin.

Over 5000 patients were recruited for the study from 19 warfarin initi-

ation centers. Participants included centers in Brazil, Israel, Japan, Singapore, South Korea, Sweden, Taiwan, the United Kingdom, and the United States. Each patient was genotyped for at least one single nucleotide polymorphism (SNP) in VKORC1, and for variants of CYP2C9. In addition, other information such as age, height, weight, race, and certain other medications was collected.

The goal of the project was to determine an initiation dose for each patient that was as close as possible to the eventual stable dose. I participated in the project design where the criteria for methods were decided. In the first phase of the project, many methods were evaluated by calculating the mean absolute error (MAE) using 10-fold cross validation on 4043 patients. The best performing methods would then be used in phase two, where they were evaluated against a held-aside set of 1009 patients.

3.2.2 Data Preparation

There are several SNPs in the VKORC1 gene, and different patients in the study were genotyped at different sites. Fortunately, these SNPs are in high linkage disequilibrium, meaning their values are highly correlated. From the patients that had multiple SNPs genotyped, we confirmed that the rest could be imputed with a high degree of accuracy. One of the SNPs was chosen as the reference for VKORC1, and its value could be imputed for all patients that had at least one VKORC1 SNP genotyped.

As mentioned earlier, a clinician chooses a target INR for warfarin treatment based on indication. In creating the models, some medical collaborators were a bit dismayed that the feature for the target INR was typically eliminated by any feature selection method run on the data. It makes sense to think that in order to raise the INR (to slow clotting), one must take more warfarin. After all, this is exactly what is done when a patient's dose is adjusted through the initiation phase. However, the data only showed a very weak relationship between the two. Part of the

weakness in the signal was the limited range of INR among the patients in the study. Most had a target INR between 2 and 3. With this information, it was decided to exclude patients with a target INR outside this range. This limited the claimed applicability of the study, but strengthened the conclusions regarding the included group.

In addition to trying to predict warfarin dose, we used models to predict square root of dose and natural logarithm of dose. Predictions were converted back to dose before calculating model statistics. Both of these transformations improved results; the final model used a square root transform.

One interesting consideration was how to model missing data. For categorical data, models often perform better if missingness is represented by a unique value. This also avoids the need to impute the value or discard the example, as may be necessary for some learning algorithms. However, it is useful to think of how this algorithm would be used in the future. While a patient's use of a particular drug might be unknown in our data, it would not be unknown to a physician entering patient information into an implementation of the algorithm. It has also been shown that it is better to learn an additional model on a reduced set of features if it is known that some will be missing in future examples rather than imputing a value (Saar-Tsechansky and Provost, 2007). However, for simplicity, missingness in categorical variables such as SNPs or race were marked as a separate value in this study.

3.2.3 Prediction Algorithms

Several algorithms were used in an attempt to predict warfarin dose (or a transformation). The first, linear regression, is a standard approach to real-valued prediction. It is well justified theoretically and can be used to both predict an output value and quantify the strength of the associations with features of the data. It has a long history of use in medical and

epidemiological applications.

Another category of models that seem appealing to solving the warfarin dose problem are regression trees. It had been noticed for some time that there seemed to be racial differences in the amount of warfarin different patients needed. Regression trees allow the data to be separated into groups by a decision tree, then assign a different linear regression equation to each leaf of the tree. This provides additional freedom in the model parameters than ordinary linear regression as the features in the tree can have a nonlinear effect on the predicted variable. There were other features in the dataset where such a division could prove useful, such gender or medication use.

Support vector regression is an extension of the support vector machine used for classification. The regression line chosen is the one that best balances the "flatness" of the model and the prediction error. The trade-off between the two factors is controlled by a parameter C. Flatness refers to the size of the vector needed to describe the regression line. In addition, this formulation of SVR uses an ϵ-insensitive penalty function, where there is no penalty for errors that are less than ϵ. Performing the optimization on the original features is called linear SVR.

Support vector machines can also take advantage of the "kernel trick", in which features are mapped into a higher dimensional space, allowing nonlinear relationships to be represented. Instead of computing a dot product between two vectors (examples), a kernel function $K(\mathbf{x}, \mathbf{y})$ is used. In the radial basis function (RBF) kernel, the kernel has the equation $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2)$. The corresponding feature space is an infinite-dimension Hilbert space, but the regularization of the model prevents this from being a problem as the features are not explicitly calculated.

3.2.4 Evaluation

Currently, clinicians typically start all patients on a fixed dose of warfarin, then adjust the dose over several weeks according to the patient's response. This research does not try to address the issue of how to perform the adjustment, but to simply do a better job of calculating a patient-specific starting dose. Therefore our first baseline is to start all patients with a fixed dose of 35 mg/week (5 mg/day)[1].

The second goal of the study is to determine the advantage gained by having genetic information available. Therefore the second baseline is a model constructed using without VKORC1 and CYP2C9. Acquiring genotypes has additional costs, and there needs to be a benefit above using what is clinically available.

Three evaluation metrics are presented. Mean absolute error (MAE) was the primary criterion used for evaluating models, and measures the average amount that a prediction deviates from the known final dose. R^2_{adj} measures the amount of variation in the outcome variable that is explained by the features, adjusting for the fact that additional features can often improve R^2 by chance. A final metric was the percentage of patients for which the predicted dose was within 20% of the actual dose, providing an idea of how many patients had a prediction that was "close enough" to the correct dose. Recall that although some models were trained on transformed dose, these metrics are calculated on original dose.

The best performing model was a SVR model trained with a RBF kernel (MAE= 8.2). However, the margin was not significant against the next two models, linear regression and linear SVR (MAE= 8.3). All three were trained against square root of final dose. Using model trained with a nonlinear kernel presented some difficulty for distribution within the

[1]Some studies have shown a shorter time to stable dose using a more aggressive 70 mg/week starting dose strategy (Kovacs et al., 2003). However, this baseline would perform even worse in our comparison than 35 mg/week, as the average final dose in our dataset was 31 mg/week

medical community. There was concern that the model could be viewed negatively as a "black box". This issue of interpretability is one that is certain to persist as newer quantitative methods are used in medical applications. The coefficients for both linear models were almost identical, so the linear regression model was presented as it was a more familiar technique to the audience.

The results of the three methods are shown in table 3.2. The differences between all pairs of algorithms are significant at $P < 0.001$ using McNemar's test of paired proportions. The final clinical and pharmacogenetic (clinical + genetic) algorithms are displayed in tables 3.3 and 3.4, respectively.

Table 3.2: Warfarin dosing model comparison

	Training		Validation	
	MAE	R^2_{adj}	MAE	R^2_{adj}
Pharmacogenetic	8.3	0.47	8.5	0.43
Clinical	10.0	0.27	9.9	0.26
Fixed dose	13.3	0	13.0	0

Table 3.3: Warfarin clinical algorithm

	4.0376	
-	0.2546	Age in decades
+	0.0118	Height in cm
+	0.0134	Weight in kg
-	0.6752	Asian race
+	0.4060	Black or African American
+	0.0443	Missing or Mixed race
+	1.2799	Enzyme inducer status
-	0.5695	Amiodarone status
=		Square root of dose

Table 3.4: Warfarin pharmacogenetic algorithm

	5.6044	
-	0.2614	Age in decades
+	0.0087	Height in cm
+	0.0128	Weight in kg
-	0.8677	VKORC1 A/G
-	1.6974	VKORC1 A/A
-	0.4854	VKORC1 genotype unknown
-	0.5211	CYP2C9 *1/*2
-	0.9357	CYP2C9 *1/*3
-	1.0616	CYP2C9 *2/*2
-	1.9206	CYP2C9 *2/*3
-	2.3312	CYP2C9 *3/*3
-	0.2188	CYP2C9 genotype unknown
-	0.1092	Asian race
-	0.2760	Black or African American
-	0.1032	Missing or Mixed race
+	1.1816	Enzyme inducer status
-	0.5503	Amiodarone status
=		Square root of dose

In looking at the percentage of patients properly predicted within 20%, we split the patients into low ($\leqslant$ 21 mg/week), intermediate ($>$ 21 and $<$ 49 mg/week), and high dose ($\geqslant$ 49 mg/week) groups. Figure 3.2A shows that in the validation cohort, the pharmacogenetic algorithm accurately identified larger proportions of patients who required 21 mg of warfarin or less per week and of those who required 49 mg or more per week to achieve the target international normalized ratio than did the clinical algorithm (49.4% vs. 33.3%, $P < 0.001$, among patients requiring $\leqslant$ 21 mg per week; and 24.8% vs. 7.2%, $P < 0.001$, among those requiring $\leqslant$ 49 mg per week). The same was true if the training set and validation set were combined (figure 3.2B).

An interesting difference between the clinical and pharmacogenetic

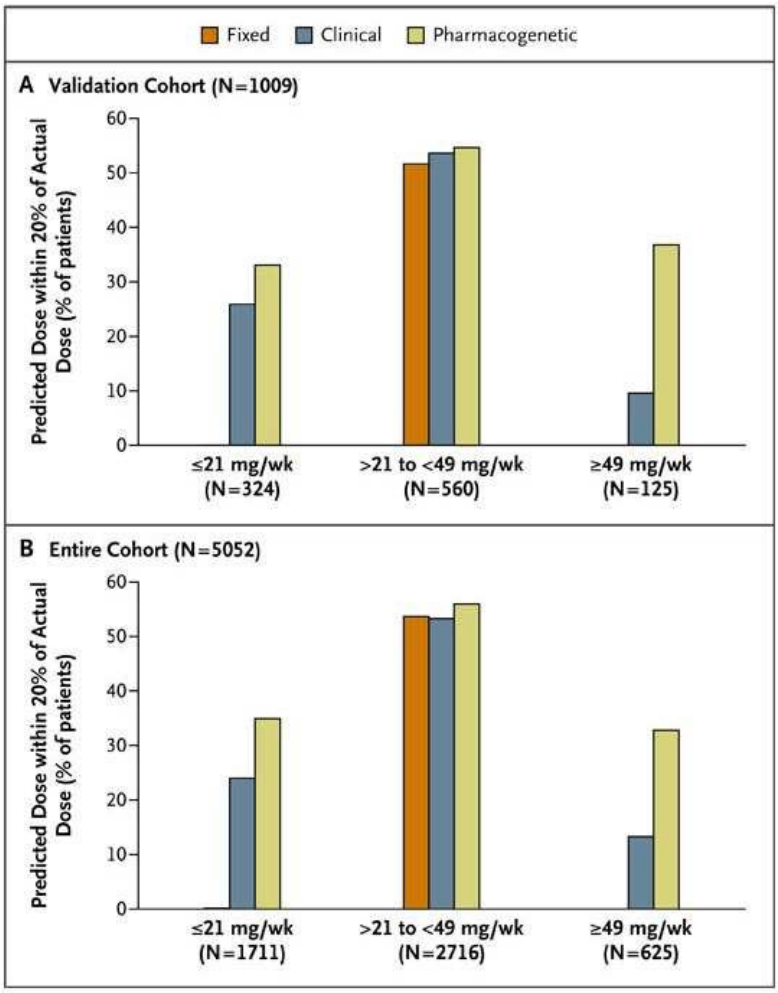


Figure 3.2: Percentage of patients in low, intermediate, and high dose groups that are predicted within 20% of actual dose. Figure from (Consortium, 2009)

model is the change in the coefficients representing race. The sign of the the coefficient for Black or African American race changes from positive (indicating a higher dose) to negative (indicating a lower dose). In fact, the variables for race could be removed from the pharmacogenetic model with little effect on the error. This effect is caused by the addition of VKORC1 to the model. It appears that race is actually an imperfect surrogate for the value of the VKORC1 SNP. This helps explain why it was not helpful to estimate separate race-specific models for dose prediction, as was attempted as part of the study.

The use of a pharmacogenetic algorithm for estimating the appropriate initial dose of warfarin produces recommendations that are significantly closer to the required stable therapeutic dose than those derived from a clinical algorithm or a fixed-dose approach. The greatest benefits were observed in the 46.2% of the population that required 21 mg or less of warfarin per week or 49 mg or more per week for therapeutic anticoagulation.

3.3 Using Electronic Health Records to Predict Therapeutic Warfarin Dose

3.3.1 Introduction

As discussed in Section 3.2.1, the commonly prescribed medication warfarin is prone to misdosing. Several factors have been identified that help explain the variation in warfarin dosages between patients including age, height, weight, race, liver disease, smoking, and drugs such as amiodarone, statins, azoles, and sulfonamide antibiotics (Gage et al., 2008). In addition, genetic variants to three genes, VKORC1, CYP2C9, and CYP4F2, have been shown to play a role (Caldwell et al., 2008). While genetic factors do improve the accuracy of dosing algorithms beyond that achieved with

clinical information alone (Consortium, 2009) as shown in the previous section, the added cost and time required for genetic tests has thus far limited their use outside of trials.

Due to their size and breadth, electronic health records (EHRs) are an important source of clinical information for producing predictive models. However, they differ in important ways from the data typically used for prospective clinical trials (Hripcsak and Albers, 2013). The amount of data available for each patient is variable and information in the record is used for purposes, such as billing, that are not part of clinical care. EHRs are observational in nature, and therefore it can be difficult to separate correlations from causative factors.

While acknowledging these challenges, access to EHRs allows the evaluation of predictive utility for a wide variety of clinical factors typically not available in a clinical trial. It is hypothesized that additional clinical factors beyond those previously described might contribute to prediction of therapeutic dose. In order to possibly identify these factors, we do not limit our learning algorithm to a small number of predetermined factors, but instead allow it to utilize any that might be useful.

3.3.2 Methods

The study population consists of 4560 patients who had undergone warfarin initiation at the Marshfield Clinic anti-coagulation service between 2007 and 2012 and attained stable dose within 210 days. The final stable dose was used as the prediction variable. Each patient in the cohort had electronic data concerning diagnoses, prescriptions, procedures, vital signs, laboratory tests, and note-extracted UMLS terms. The study was approved by the institutional review boards at Marshfield Clinic and University of Wisconsin-Madison. The details of the population are shown in 3.5.

Most previous warfarin prediction papers have used linear regression

Table 3.5: Cohort summary. Standard deviations shown in parentheses.

Number of patients	4560
Age at initiation (yrs)	66.0 (15.5)
Warfarin stable dose (mg/wk)	35.8 (16.6)
EHR interactions (days)	137.4 (111.5)

to create the models (often with a logarithmic or square root transform of the dose). Linear models work well with a small set of preselected features, and have high interpretability of feature coefficients. However, linear regression was not effective for our task, due in part to the high degree of collinearity between the features and the number of features exceeding the number of patients (though results improved slightly with regularization). Instead, we employ a random forest regression model (Breiman, 2001), introduced in Section 2.3.1 Random forests consist of many decision trees, each of which is trained on a different bootstrap sample of the original data. In addition, when the decision tree chooses a split variable, it may only choose from a random subset of the features. In EHR data, it is likely that several features represent the same underlying event (for example, a particular drug, procedure, and diagnosis might be equally indicative of a condition that a patient had treated), and ensemble methods like random forests tend to perform well in this case.

Study data were restricted to data available prior to initial warfarin initiation date. Patient history was divided into multiple overlapping windows covering 1, 3, and 5 years prior to warfarin initiation and the number of occurrences of each window was counted. For example, if a patient had prescriptions of azithromycin 10, 8, and 2 years before initiation, the features $\mathtt{azith}_{\mathtt{1yr}}$, $\mathtt{azith}_{\mathtt{3yr}}$, $\mathtt{azith}_{\mathtt{5yr}}$, and $\mathtt{azith}_{ever}$ would have values of 0, 1, 1, and 3, respectively. For numeric values, such as blood pressure values, features were created for the minimum, maximum, and mean values within each window. Features that were nonzero in fewer

than 0.5% of patients were removed, resulting in 21632 features.

We examine two metrics to compare our models. First is the mean absolute error (MAE) of the predicted warfarin dose to actual warfarin dose. Second is R^2, a measure of the amount of variation in warfarin dose accounted for by the model. Results presented are out-of-bag estimates, in which the trees in the model that were not trained with a given example are used to evaluate it. This is possible because bootstrap samples cause some examples to be unused in building the tree and has been shown to be consistent with cross-validation results (Breiman, 2001). Random forests were constructed with 500 trees and 1/3 of the variables available per split.

3.3.3 Results

Electronic health records do not contain the same amount of information for each patient. Patients with less information available could be more difficult to predict. We examined the potential of such effects using a somewhat crude measure of interactions with the health system. Our utilization metric is the number of unique days in which the patient has any records in the system in any category. This metric is biased somewhat towards medications, as medication refills typically occur more frequently than clinic visits. Figure 3.3 shows a sharp decline in mean average error between the third and fourth deciles. Patients in the first three deciles have fewer than 64 days of utilization.

Our subsequent analyses will use a "high utilization" subset of 3202 patients (i.e., patients in deciles 4-10 of the utilization metric). We compare the model using all of our features to EHR-based analogs of two previous clinical models. The Subset model uses all features in our dataset analogous to the clinical algorithm (age, gender, height, weight, race, amiodarone, and enzyme inducers) from (Consortium, 2009). The Subset+ algorithm adds features used in other clinical algorithms (Gage et al., 2008),

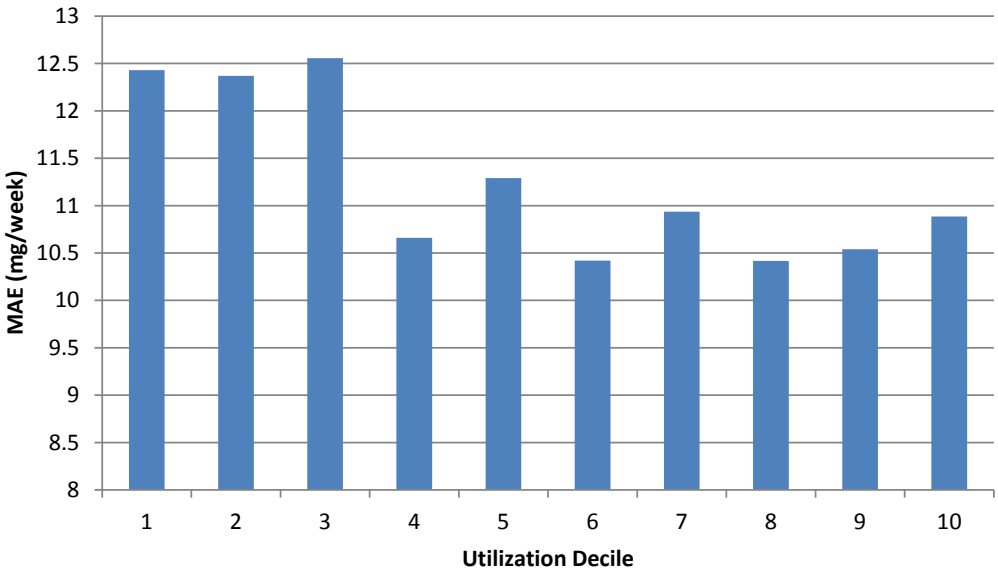


Figure 3.3: Mean average error of the random forest model as utilization changes. Patients are split into deciles based on EHR utilization, from low (1) to high (10).

such as tobacco use, diabetes, statins, sulfonamides, and liver disease. The results in terms of MAE and R^2 are shown in Figure 3.4. As can be seen, the complete model outperforms the others.

Despite the fact that random forests produce a highly complex non-linear model, it is possible from the resulting forest model to determine the importance of individual features. This is done by measuring the impact on the accuracy of the model if a feature changes value. If the model's performance decreases when the feature is assumed to change, it is determined to be important. Table 3.6 shows the most important features for the full model.

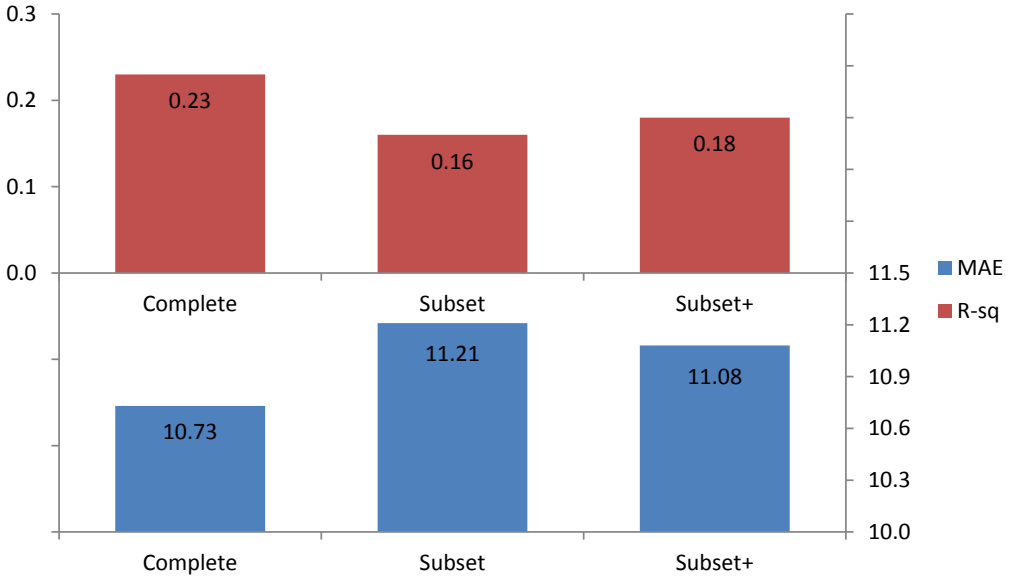


Figure 3.4: Model performance on high utilization patients using different feature sets. Complete uses all available features. Subset includes EHR analogs of features from (4). Subset+ additionally contains features used in other models (Gage et al., 2008). Top chart shows explained variance (R^2). Bottom shows MAE. The differences between all pairs of models are statistically significant at $p < 0.01$ using a two-sided paired t-test on the absolute error.

3.3.4 Discussion

It is useful to compare these results to other warfarin dosing algorithms, despite the methodological differences in using EHR versus trial data. The linear model from the IWPC (Table 3.2) reports a validation MAE of 9.9 and a R2 of 0.26. The clinical model in Gage et al. (2008) reports an MAE of 10.5 and an R2 of 0.17. Ramirez et al. (2012) reports an MAE of 11.8 and an R2 of 0.24 on their European-American cohort. Our Subset and Subset+ seem to perform worse than these models, but the Complete model is competitive, in as much as the two types of trials can be directly compared even though the patient sets and data collected are different.

Table 3.6: Most important features in high utilization full model. Entries with a (*) are abbreviated in this list. Both weight and height appear multiple times at the top of the list over different time windows (ever, 3 years, etc.) and only the impact of the most important variant is shown.

Feature	% Error impact
Age	15.8
Weight	3.2(*)
Height	2.5(*)
Irregular Bleeding (NOS)	2.4
Gender	2.4
Potassium Chloride	2.3
Race	1.9
Blood Creatinine Lab Test	1.7

Much of the interest in the prediction of warfarin dose concerns the use of genetic factors to aid prediction (Consortium, 2009; Ramirez et al., 2012). The addition of genetic factors improved the R^2 of the linear IWPC model from 0.26 to 0.43 (see Table 3.2). However, genetic information is still quite rare in EHRs. Pharmacogenetic testing is being explored for use for several conditions, but is not yet widespread. When available, genetic information can readily be incorporated into the framework presented here.

The important features listed in Table 3.6 contain a few previously unused factors, such as previous irregular bleeding and use of potassium chloride. These might be of interest for independent study. But many of the top features are the ones used in other clinical models. There is however a long tail of features with small importance scores, which may help explain why the Complete model outperforms the Subset and Subset+ models. When the features are highly correlated and many have small predictive ability, there can be value to incorporate them into the model to realize a cumulative benefit in performance.

Because the model operates on the EHR, it is amenable to integration into clinical decision support systems. Previous algorithms might require more information or tests to be performed on the patient at the time of warfarin initiation, as they assume the data are collected as it would be in a clinical trial. EHR-resident models take into account the existence of correlated events in the medical history that can be used to make predictions with or without the most recent information and take into account current practice in entering information into the EHR.

There is increasing support for improved clinical-data-only algorithms to guide warfarin initiation processes (Scott and Lubitz, 2014) following the completion of two large clinical trials that showed improved outcomes from pharmacogenetic dosing over fixed dosing (Pirmohamed et al., 2013), but failed to demonstrate a difference in outcomes between pharmacogenetic and clinical dosing (Kimmel et al., 2013). As more clinical data are stored and accessible in electronic records, personalized models that take into account the various ways that people interact with the health care system can help translate the results of trials to clinical use through decision support.

3.4 Predicting Incident Atrial Fibrillation/Flutter and Its Associated Risk of Stroke and Death

3.4.1 Atrial Fibrillation/Flutter

AF/F causes electro-mechanical dysfunction of both atria of the heart, a process which typically causes a rapid and irregular rhythm of the lower chambers. These hemodynamic abnormalities alter blood flow and decrease cardiac output, thus causing symptoms of palpitations, shortness

of breath, easy fatigability, and heart failure, among others. AF/F is associated with a fivefold increase in the risk of stroke and twofold increase in mortality (Greenlee and Vidaillet, 2005; Vidaillet et al., 2002). Despite recent advances in available treatments, AF/F continues to adversely impact quality of life, cost, and survival.

AF/F is also important to study because there are effective and relatively low cost anticoagulation treatments available that could prevent two-thirds of AF/F-related strokes. In the US alone, it is estimated there are more than 1.2 million cases of atrial fibrillation annually (Colilla et al., 2013) and rising (Naccarelli et al., 2009). Atrial fibrillation affects 10% of people over 80 years of age and the lifetime risk is one in four (Lloyd-Jones et al., 2004).

Each year there are approximately 700,000 strokes in the US that contribute directly or indirectly to 165,000 annual deaths (Ingall, 2004); stroke consistently ranks among the top 5 causes of death, usually trailing only heart disease and cancer. Enhanced capabilities to identify individuals at high risk to develop AF/F could allow earlier intervention with effective low cost treatments, like warfarin. Early intervention is highly desirable as this is likely to lower costs while also reducing morbidity and mortality. Risk models with clinically-actionable prediction accuracies that support such interventions early, combined with primary prevention efforts to reduce incidence of predisposing comorbidities, such as diabetes and obesity, are key factors in achieving better health and lower cost outcomes.

This project sought to accurately predict AF/F and subsequent stroke and mortality using coded EHR data without manual specification of a limited feature subset. Previous algorithms for determining atrial fibrillation risk (Schnabel et al., 2009, 2010) have utilized a small number of clinical features based on clinical knowledge. Some of these, such as PR interval, are often not available in a coded format in patients not part of clinical trials such as those used in previous studies. We desire a predictive

model that can be applied to observational data in which there is wide variety in the patient interactions with the medical system.

Compared to prospective clinical trials, observational studies utilizing EHR data have advantages and disadvantages inherent to clinical practice. One such limitation is the potential of sources of error that can affect the reliability of the records, such as the irregular time intervals between visits. Within a pool of patients, the set of patients who truly have a given condition is not necessarily the same as the set of patients labeled with an occurrence of a given diagnostic code (such as ICD-9) in the EHR. In order to establish reliable risk models, it was crucial to develop reliable and valid phenotypes for AF/F and acute stroke.

3.4.2 Materials and Methods

The AF/F study was designed to identify subjects whose new diagnosis of AF/F would be expected to be associated with a substantially increased risk in morbidity and mortality. We focused on this group because it represents the largest subset of patients with new onset AF/F and because these individuals would benefit most from early diagnosis and treatment of AF/F to reduce their increased risk of stroke and death. Because patients whose AF/F onset resulted from known transient or reversible predisposing comorbidities are not thought to be at such an elevated stroke risk, we excluded patients from the cohort if the otherwise qualifying episode of ECG-documented AF/F occurred within 90 days of a health event known to induce transient (short-lived and nonrecurring) AF/F (see exclusions). The study was approved by the Marshfield Clinic institutional review board.

Subjects representing new onset AF/F cases were selected based on the criteria in Table 3.7. Forty AF/F case subjects and forty control subjects were selected at random for manual review by trained staff and a board certified cardiologist. All reviewed subjects (both cases and controls) met

their respective inclusion criteria, and none demonstrated any evidence of previous history of AF/F. After applying the criteria for identifying acute ischemic stroke among the AF/F cases, we manually evaluated 51 cases meeting the criteria and found that 48, or 94%, were true acute ischemic strokes.

Table 3.7: Cohort selection criteria

Cohort	Inclusion Criteria
AF/F subjects	2 diagnoses of AF/F (ICD 427.31 or 427.32) + 1 of the diagnoses from cardio specialist + positive ECG
Exclusions	Coronary artery bypass surgery (CPT 33510-33536), Cardiac valve surgery (CPT 33400-33475), Open procedure for AF/F ablation (CPT 33250-33261), Percutaneous transcatheter ablation (CPT 33650-33651), Major trauma (ICD-9 codes 860.0-869.1), or hyperthyroid treatment initiation within 90 days
AF/F controls	0 diagnoses of AF/F + negative ECG
Stroke subjects (among AF/F subjects)	1 diagnosis of ischemic stroke (ICD 434) + Either second stroke diagnosis or death within 7 days
Exclusions	Subarachnoid hemorrhage (ICD 430), Intracerebral hemorrhage (ICD 431), or Unspecified intracranial hemorrhage (ICD 432)

Marshfield Clinic actively seeks and initiates processes to obtain accurate patient death information. These processes include: scanning local newspaper and internet obituaries, gathering data from insurers and hospitals, consulting with home health agencies and patient care registries, and electronically identifying patients over 110 years of age with no clinical contact "presumed dead".

The study cohort was determined using a highly structured process that ensured that both AF/F cases and controls reflected the same age and gender composition and met standards for encounters with Marshfield

Clinic providers to ensure that adequate medical data were available for case and control determination and for modeling. We elected to focus our analyses on the subset of expected incident AF/F cases that had at least annual Marshfield Clinic provider encounters in the 36-months prior to AF/F onset.

The final AF/F analysis data set consisted of 8,054 unique patient records, including 3,762 with new onset AF/F (46.7%) and 4,292 (53.3%) non-AF/F control records. Frequency counts of other study outcome measures (i.e., stroke and mortality at 1- and 3-years) are reported in Table 3.8. AF/F cases span a 16 year period; the earliest new onset AF/F case in the analysis set was recorded in January 1995 while the most recent AF/F case was recorded in December 2010.

Table 3.8: Summary of AF/F Case and Control Cohorts and Outcome Measures Model Source Data

	Total	Stroke	1-Year Mortality	3-Year Mortality
AF/F Cases	3,762	299	383	526
Controls	4,292	100	516	668
Total	8,054	399	899	1,214

A major potential advantage of the study is the availability of relatively comprehensive clinical and phenotypic data for large AF/F cohort and control groups. A series of steps were taken to process the AF/F data. For each patient, only data prior to AF/F diagnosis (or similar age threshold for control patients) were used to produce predictive features. We censored one month before the recorded date of the patient's first AF/F diagnosis; this month "cushion" is needed because sometimes diagnoses are dated by final date of diagnosis confirmation rather than actual date of the event. Several types of data available in the electronic medical record were used, including gender, age, diagnoses, drugs, laboratory tests, procedures, and vital signs. In order to prepare data for analyses, we propositionalized the

data so that each patient was described by an equal number of features. It was important to capture the temporal nature of the data; therefore, for each drug, diagnosis, procedure, and test, a feature was created to characterize the frequency of that event for a patient in the last year, last 3 years, last 5 years, and at any time in the record. Vital signs were processed similarly except aggregated to include min, max, and mean readings. Diagnoses and procedures were also generalized into categories in order to provide an additional form of aggregation, using existing ICD-9 and CPT hierarchies. Features were pruned if they occurred in fewer than 5 patients total because such data are not informative. Over 25,000 features were used for analyses for each patient.

Many machine learning algorithms were evaluated, including support vector machines, decision trees, logistic regression, and random forests. In addition, several methods for feature selection and model averaging were employed. For all prediction tasks, random forests (Breiman, 2001) had the best performance, so their results are presented.

Random forests (see Section 2.3.1) are a method for inducing a collection of decision trees to predict an outcome. The method differs from standard decision tree learning in three important ways. Each decision tree is trained with a different bootstrap sample, or subset of the original examples drawn with replacement, causing some examples to be drawn multiple times and others to be not used at all. The unused examples for each tree are used to make "out-of-bag" predictions of model accuracy. Each time the algorithm is deciding on which feature to use to create a split in the tree, only a small random subset of the features is considered. Unlike most tree learning algorithms, the trees are not pruned. The predictions of the various trees are used to form a weighted average, with tree weights based on the out-of-bag examples, in order to make a final prediction on a new (test) example.

We report several performance metrics, primarily the area under the

receiver operating characteristic curve (AUC); from the ROC curve sensitivity, specificity, and accuracy at a wide variety of operating points can easily be identified. AUC corresponds to the probability that an algorithm will (correctly) give a higher score to a randomly chosen case patient than to a randomly chosen control patient. All results presented in the following section are based on out-of-bag estimates. Ten-fold cross validation was also performed and results were very similar. The random forest implementation used was the `randomForest` package in the R statistical environment.

3.4.3 Results

3.4.3.1 Predicting Onset of AF/F

A random forest approach with a large number of decision trees (250) yielded an AUC of 0.70, as shown in Table 3.9. The ROC curve in Figure 3.5 shows, for example, that we can obtain a specificity of just over 0.70 (false positive rate under 0.30) at a sensitivity (true positive rate) of around 0.60. Alternatively, if one wanted to identify only those at very high risk of developing AF/F (e.g., potential candidates for targeted preventive strategies), we could identify roughly 30% of the patients who will have new onset AF/F (true positive rate of 0.30) with a false positive rate of only 0.10.

Table 3.9: Performance of the prediction models.

	AUC	Sens	Spec	Acc	True +	False -	False +	True -
AF/F onset	0.70	0.71	0.56	0.64	2,101	1,661	1,228	3,064
Stroke	0.60	0.55	0.61	0.58	183	116	136	163
1 yr Morality	0.73	0.69	0.66	0.67	252	131	120	263
3 yr Mortality	0.74	0.72	0.64	0.68	336	190	146	380

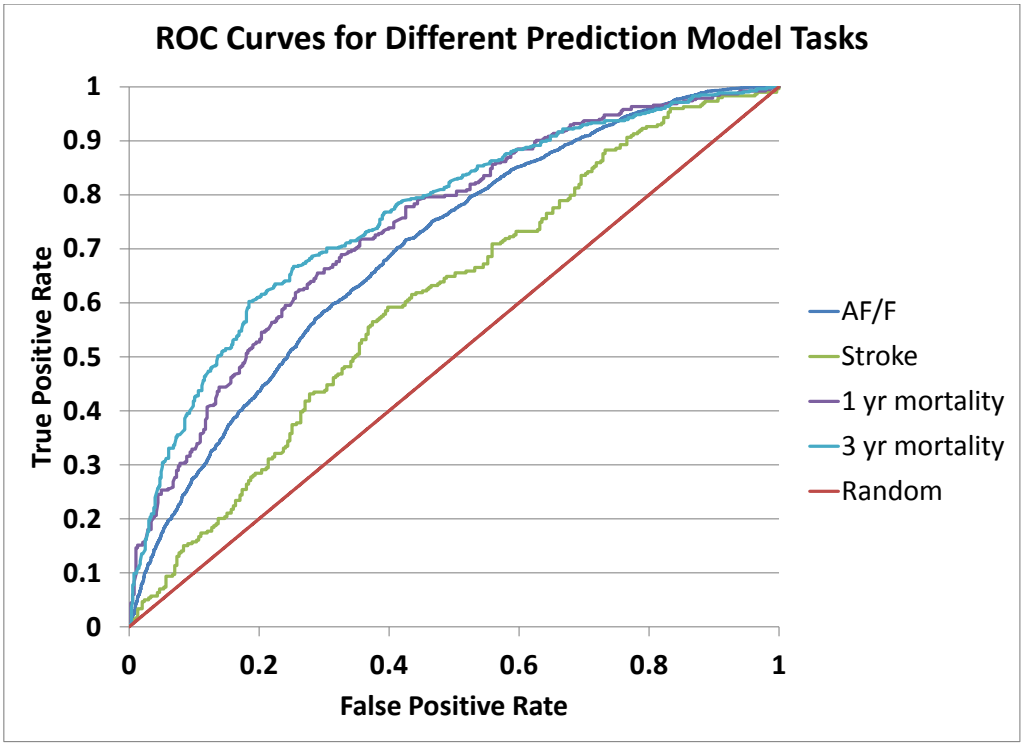


Figure 3.5: ROC curves for different AF/F models.

A random forest of 250 trees tends to be much more complex than a linear model, making it more difficult than with linear models to determine the contribution of each individual feature by visual inspection of the model; nevertheless, it is possible to rank the features by their contribution to the overall model. This is done by randomly permuting the feature values of the out-of-bag examples in a given tree in the random forest and measuring the average change in error rate. Table 3.10 lists the ten features with the largest such importance score for predicting AF/F. It is worth noting that the importance scores for this model do not decay quickly; over 3000 features have non-negligible importance scores. The top features are presented to provide some insight into the model, but this insight is necessarily incomplete due to model complexity.

3.4.3.2 Stroke Prediction Conditional on AF/F

We developed models to predict acute stroke subsequent to a patient's onset of AF/F. Because an acute stroke occurred in less than 10 percent of the cases with new onset AF/F during the period of observation, there was a significant imbalance in our case and control numbers for this analysis (i.e., about one case for every 10 controls). In response, we engaged algorithms using cost-sensitive classification that implement weighting schemes where cases are given more weight than controls by the learning algorithm. Best performance was obtained for most metrics by subsampling, where a random subset of the controls were chosen to bring the numbers of cases and controls into better balance. As is apparent from the Figure 3.5 and the sample operating point in Table 3.9, model predictions are only marginally better than random (the light blue diagonal line). The most important features are shown in Table 3.10.

Following up on the relatively poor performance on the stroke task, we constructed a model based on a subset of the original feature set which included several features that have been previously shown to correlate with increased stroke risk. Those features included age, gender, prothrombin time (INR) level, warfarin usage, and several echocardiograph measurements (left atrial size, left ventricular ejection fraction, and left ventricular wall thickness). Random forests were again used, although a smaller number of trees were constructed. This model had an AUC of 0.61 and an accuracy of 0.58, performing similarly to the model with the full set of features.

3.4.3.3 Mortality Prediction Conditional on AF/F

We developed models to predict death at both 1- and 3-year intervals following AF/F diagnosis. There were 384 patients for whom there was a record of death from any cause within one year of AF/F diagnosis in our cohort, and 527 patients who died within 3 years. (Note: The number

of controls in these models is reduced due to censoring issues.) Again, as was the pattern in our other models, random forests on subsampled data produced the best performance. These results suggest that either of these predictive models could be used to identify which AF/F patients may benefit from more aggressive therapies designed to potentially improve life expectancy. The most important features are shown in Table 3.10.

3.4.4 Discussion

Inspection of the most important features shows several patterns. One expected finding is that the same feature may appear multiple times with similar importance at different time granularities (e.g. `Other forms of Heart Disease` in the AF/F model). This is somewhat a byproduct of how the features are constructed; the list of patients who have had a procedure in the last 3 years will be highly correlated with the patients who have had the same procedure in the last 5 years. This co-linearity in the features may be one explanation for the success of the random forest model on these data, as it does not suffer in the face of co-linear features as much as other methods such as logistic regression. Another expected finding is the presence of features that are correlated but obviously not causative of the outcome. For example, getting a certain laboratory test is not the cause of increased risk of stroke, but may be conducted in cases where related underlying comorbidities may be suspected. Unfortunately, this phenomenon makes it more challenging to interpret the list of important features.

Many of the diagnosis and procedure codes in the top ten are groups of similar codes (i.e. `Diseases of the Circulatory System` rather than `Benign Secondary Hypertension`). The availability of these features helps the model aggregate a large number of related conditions at a single node in the decision tree, utilizing lower levels of the tree to provide exceptions if necessary. One surprising absence from the list are medications, which

despite their obvious preventative or causative effect on health outcomes do not occur frequently at the top of the rankings with this metric. For example, the top drug in the AF/F model is furosemide at #14, and for the stroke model is rofecoxib at #44. Furosemide is a diuretic used in treatment of high blood pressure and heart failure, comorbidities known to increase the risk of developing AF/F. Rofecoxib is a nonsteroidal anti-inflammatory agent (a selective COX-2 inhibitor) previously used to treat osteoarthritis, acute pain conditions, and dysmenorrhea (a medical condition that causes severe pain during menstruation). Rofecoxib was voluntary withdrawn by the manufacturer in 2004 due to safety concerns of acute increased risk of cardiovascular events (including heart attack and stroke, see Section 3.1.1).

The performance of the stroke model constructed with the hand-selected set of features has several implications. First is that equivalent performance to a crafted set can be achieved without using prior knowledge about the condition being predicted. On the other hand, the thousands of additional features are not improving the accuracy of the model in this case. Of note is that none of the features included in the smaller model, except for age, appear at all in the top 200 most important features of the larger model, yet they achieve similar performance. This implies that perhaps the EHR data contains many redundant expressions of the underlying hidden patient health state that we are trying to measure.

Improved AF/F risk models are of continuing clinical interest (Dewland et al., 2013; Kramer and Zimetbaum, 2013). Most previous AF/F risk models (Schnabel et al., 2009, 2010) are based on features shown to be related to cardiovascular disease through prospective studies such as the Framingham Heart Study. Researchers collect data thought to be related to the condition of interest at regular intervals in order to build their model. To apply to a new patient, all of the features required for the model must be measured in order for a prediction to be made. Our model doesn't require an expert selected subset of features to be chosen and measured

in order to be applied to a new patient. Concerning the prior risk models, they achieved a C statistic of 0.78 on training data (Schnabel et al., 2009) and 0.68 on validation data (Schnabel et al., 2010). The C statistic is similar to AUC, and therefore can be interpreted as performing similarly to our model, although those studies predicted risk over a longer 10 year window.

Another related line of work utilizes the natural language notes present in many EHR systems instead of or in addition to coded data; analysis of coded data together with text notes has been previously applied to predicting AF/F onset in a smaller patient cohort (Karnik et al., 2012). That study found that the addition of text notes did not improve the accuracy of a model with coded data, indicating that our model may not be limited by not incorporating natural language features.

The fact that our technique does not rely on pre-selected features also allows it to be applied broadly to many complex and etiologically heterogeneous conditions without making prior assumptions as to which features are important. While such use of pre-selected features is common (Wu et al., 2010; Barrett et al., 2011) and sometimes necessary for computational reasons for certain methods, it must be done independently for each condition and may not increase model accuracy, particularly in observational settings.

It should be noted that the maximum achievable accuracy or AUC varies substantially from task to task. Consider that an AUC of over 0.75 can be obtained for predicting myocardial infarction from EHR (Weiss et al., 2012) while the AUC for predicting breast cancer under the Gail model and variations is estimated at only 0.63 (Gail, 2009; Meads et al., 2012). Therefore, what is important to know is the current best that can be done for each disease of interest in a specific patient cohort, so efforts can be made to improve accuracy whether through longer periods of observations, improved machine learning algorithms, improved clinical

and genetic variables, larger populations, and/or improved phenotype definitions. It is also important to consider how the model will be used or can best be used (e.g. to identify high-risk patients and take stronger preventive measures for them), and what minimum values of accuracy measures such as sensitivity and specificity are necessary for this use.

The current study has several limitations. While observational data is more readily available than prospective data, it introduces sources of systematic errors and makes it more difficult to make causal arguments. The large number of features with non-negligible importance scores hurts the interpretability of the model. Since the stroke and death cohorts were subsets of the original AF/F cohort, they are not as large as the AF/F model. Lastly, it is not clear how the model would perform in other populations from different healthcare delivery systems.

Table 3.10: Most important features in the prediction models

	Feature	Time	Err Red.
AF/F			
1	Mean Weight	year	4.18
2	Diseases of the Circulatory System (390-450)	Ever	4.10
3	Diseases of the Eye and Adnexa (360-379)	3 years	4.10
4	Other forms of Heart Disease (420-429)	5 years	3.97
5	Other forms of Heart Disease (420-429)	3 years	3.95
6	Minimum Body Mass Index	5 years	3.95
7	Mean Body Mass Index	3 years	3.82
8	Other forms of Heart Disease (420-429)	year	3.79
9	Phosphorous Level	Ever	3.77
Stroke			
1	Patient age		2.87
2	Urinalysis - Specific Volume	Ever	2.71
3	Minimum Weight	Ever	2.27
4	Ophthalmological Medical Exam (92012)	Ever	2.26
5	Urinalysis - Fine Granular Casts	Ever	2.22
6	Urine Bile	Ever	2.20
7	Diseases Blood and B-F Organs (280-289)	Ever	2.16
8	Maximum Systolic Blood Pressure	3 years	2.15
9	Cholesterol Percentile	Ever	2.14
1 Year Mort			
1	Minimum Body Mass Index	3 years	2.69
2	Persons Without Reported Diagnosis (V70-V82)	3 years	2.53
3	Mean Height	Ever	2.51
4	Mean Corpuscular Volume Normal	Ever	2.50
5	Minimum Weight	Ever	2.47
6	Mean Corpuscular Hemoglobin Test	Ever	2.41
7	Evaluation and Management (99212-99215)	Ever	2.40
8	Triglycerides Normal	Ever	2.38
9	Minimum Systolic Blood Pressure	3 years	2.31
3 Year Mort			
1	Glucose High or Critical High	Ever	2.91
2	Blood Urea Nitrogen High or Critical High	3 years	2.83
3	Charlson Index		2.76
4	Cataract NOS (366.9)	Ever	2.70
5	Anion Gap Test	Ever	2.66
6	Glucose Test	5 years	2.65
7	Subsequent Hospital Care 25 min (99232)	Ever	2.64
8	Blood Urea Nitrogen	3 years	2.59
9	Diseases of the Blood Organs (280-289)	year	2.58

4 DERIVING SEMANTIC RELATIONSHIPS FROM HETEROGENEOUS CODED EHR EVENTS USING WORD EMBEDDING

Informed by the results of the previous chapter in finding a representation of the EHR amenable to machine learning, in this chapter we explore a technique to quantify the relationships between events in the EHR without using any outside knowledge. The electronic health record (EHR) contains a collection of events describing the medications, diagnoses, procedures, and tests that were recorded for a patient (see Section 2.1). These events are coded in a variety of different ontologies, with different levels of specificity. When using the EHR for predictive tasks (like those in Chapter 3), each type of event is often treated as independent, while there is a complex relationship between events both within and across ontologies. We adapt a recently developed word embedding model for textual data called skip-gram that allows us to account for the temporal similarity between events. Each event code is represented by a high dimensional vector, forming clusters of codes used in similar ways without being provided any additional medical knowledge. We demonstrate some interesting properties of this embedding, and describe how it can be used in EHR prediction tasks.

4.1 Introduction

While an increasing amount of medical information is available in EHRs, utilizing it remains a challenge. A number of ontologies are used for coding different types of data, including ICD-9, ICD-10, CPT, NDC, SNOMED, and many more. Some ontologies try to represent underlying relationships between the events. As an example, ICD-9-CM organizes diagnosis codes by the type of disorder or the area of the body effected, creating a hierarchy.

However, this may group together disorders with very different etiologies, while ignoring systemic disorders. Hierarchical relationships are typically limited in scope and don't apply between ontologies. For example, when making an evaluation about a patient, how similar is a diagnosis code for a broken arm to a procedure code for a applying an arm cast?

Another problem with large ontologies is that many of the events occur very rarely. When codes are treated as independent from each other, rarer codes are often removed from analysis because they only occur in a small fraction of the population. This removes a lot of potentially useful information from the analysis.

One way to approach the relatedness between codes is to look at cooccurrence within a patient. This approach would likely do a good job of finding a correlation between urinary tract infections and amoxicillin, for example. However, it would not necessarily find the similarity between different antibiotics, as typically only one is used to treat a particular illness. Instead, we want a method that places together codes that occur in similar circumstances among different patients.

Essentially, we want a method that allows the calculation of a similarity score between any two codes, among all utilized ontologies. The crucial insight is to view the events in a medical history as analogous to words in a sentence, treating temporal order as equivalent to word order. Text has many of the same issues as coded events, such as relationships between words that more complex than hierarchies (e.g. part of speech tagging) can capture.

Much traditional natural language processing (NLP) has utilized a representation called bag of words (Manning et al., 2008), where a feature is created for each word (or pair of words) and set to 0 if it does not appear in the document and 1 (or a count) if it does. A similar encoding is often used for coded events in EHR-based models. An alternative scheme called distributed representation (also known as word embedding) has been

adopted in many recent models because it does a better job of capturing relationships between words (Baroni et al., 2014).

We present a model for creating neural network distributed representations that has had significant success in NLP tasks. Then we propose a timeline sampling technique that adapts the method to EHR events. Using only the order of events in a patient medical record, the model is able to learn relationships among over 30,000 event codes across 5 different ontologies.

4.1.1 Distributed Representations

The typical way to represent a word in a text corpus is called a one-hot representation. You have a vocabulary V of words in your corpus, so a word can be represented as a vector of length $|V|$ that is zero everywhere except for the one position representing the current word. In this way a document can be represented using bag of words (BOW), which is the sum of all the word vectors. Alternately, it is the count of every word in the vocabulary in the document. The model is readily extended to groups of words (bigrams, trigrams).

One major problem with the BOW representation is that it ignores any relationship between the words themselves. More precisely, it assumes that all words are equally distant from all other words. Another problem is the removal of any information representing word order. An alternative representation that attempts to improve upon these factors is the distributed representation. Each word is a point in a high dimensional space (though a lower space than $|V|$) in such a way that words that are "closer" in semantics are closer in this space. This is called a word embedding, as the words are transformed from a sparse high dimensional space to a dense lower dimensional space.

There are several possible ways to create a distributed representation, as well as different ways to describe closeness, but the most popular ones

use neural networks to examine the context (e.g. preceding and following words) around each word. These models use a sliding window along the text where some words in the window are used to predict the others. This transforms an unsupervised task (there is no prediction goal defined) into a supervised task.

The skip-gram model was recently introduced to calculate distributed representations (Mikolov et al., 2013a). Compared to previous neural network language models, it is very fast to train while maintaining high performance in semantic prediction tasks. Using the current word, the model tries to predict the words before and after it in the window. The architecture of the model is shown in Figure 4.1 with a window of 2 words on either side. To start, we choose the parameter s for the size of the distributed vectors. Each box represents in the figure represents a vector of size s. The vectors are initialized randomly for each word. The arrows represent fully connected neural network layers, i.e. s^2 weights connecting all pairs of elements in both vectors. Using the current weights, the model will produce a prediction for the context words that differs from their current representation. The model uses standard backpropagation to correct the distributed representation of the current word given these errors. A similar model called continuous bag of words has the opposite set up; the average of the context words are used to predict the current word. Note that skip-gram does not contain a non-linear hidden layer like many neural network models. This allows the network to more readily scale to larger corpora and dimensionality s, as shown in Mikolov et al. (2013a).

The result is that words that are used in similar contexts get mapped to similar points in the embedding space. For example, the vector representation of `Paris` is close to that of `Rome` because both are used in similar contexts. Similarly, synonyms and plurals often have close embeddings.

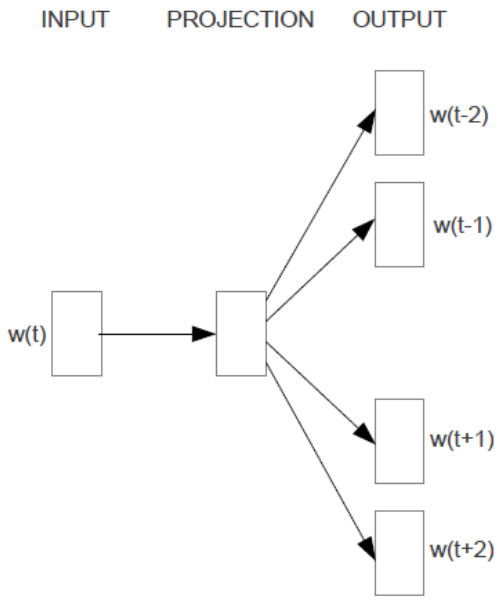


Figure 4.1: Diagram of the skip-gram architecture. The word at the current position t is used to predict the surrounding words. Figure from Mikolov et al. (2013a).

4.1.2 EHR Event Embedding

When using EHR data for prediction tasks, a representation very similar to bag of words is used, where a variable represents the number of times a patient is documented with a particular code (Wu et al., 2010; Lantz et al., 2015b). However, like words in a sentence, the EHR consists of a series of tokens. We can treat the events that take place in a patient timeline as a sentence of tokens that follow temporal patterns. While this event sequence doesn't have the syntactic rules of English sentences, it does contain subsequences that encode medical practice. For example, a patient may have a lab test, followed by a diagnosis, followed by a prescription. Using timelines from many different patients, we would expect the different prescriptions that a patient might receive for a particular diagnosis to

have similar distributed representations.

However, EHR timelines have an additional problem not seen in text: that multiple events can occur at the same time. This cooccurrence problem is somewhat dependent on the granularity of the available data and the fidelity of recording. Data recorded on a daily or weekly basis will have more cooccurrence than data with exact timestamps. Even if to-the-minute timing is available, the true temporal order may be confused by clinical process effects that delay the recording of some types of events (e.g. lab tests recorded when they are completed versus ordered, emergency room actions not recorded until after the incident). In our experiments we use the time granularity of a day to balance these effects.

Nonetheless we have multiple events occurring at a time point, unlike in text. If all events happening at a given time were just listed in the timeline, this would artificially induce an implied temporal order that is not justified. For example, suppose a lab test with several components was ordered in response to a diagnosis. The skip-gram model should be trained with the context that the diagnosis precedes each of the lab tests, not that one lab test precedes another.

In order to overcome this, we propose a mechanism to sample from the patient timeline so that all events occur at separate timepoints. For each unique timepoint represented the timeline, select at random one event that occurred at that timepoint, resulting in a subset of events. Sampling can be repeated, resulting in slightly different sampled timelines each time, but preserving the temporal ordering of the events. This is illustrated in Table 4.1. It is possible that some events (and combinations of events) will not be sampled in this process.

We utilized a large set of anonymized EHR data from Marshfield Clinic in Marshfield, WI. There were 1.1 million unique patients. We included data from medications (as drug/combination names), diagnoses (as ICD-9-CM codes), procedures (as CPT codes), lab tests (internal codes), and vitals

Day 1	Day 2	Day 3	Day 4	Day 5
A	{B, C}	{D, E, F}	G	{H, I}
Possible Samples				
ACDGH, ABFGH, ABDGI, ABEGI, ACFGI, ACEGH				

Table 4.1: Illustration of the sampling process. The first two rows show a timeline of 5 days for one patient with the letters representing events occurring that day. The last row shows several possible sampled timelines.

measurements (internal codes). After removing codes that occurred fewer than 10 times, there were 36,582 event codes. The sampling procedure described above was performed 10 times per patient, resulting in a corpus of 795 million clinical events.

The skip-gram model was implemented in `word2vec`[1]. Default parameters were used, with the size of the vectors (s) set to 100.

4.2 Results

After running the model, each event code is a point in a s-dimensional space. We can examine how the codes group by looking at the closest codes to a query code. The distance used is cosine distance, defined as $\frac{A \cdot B}{\|A\|\|B\|}$. The 10 closest codes for four sample queries are shown in Table 4.2.

Many of the relationships shown in the table are easy to understand and match our intuitions. In the `hydrocodone` example, the top results include brand name drugs that contain it (`vicodin`, `norco`, `lortab`). The rest are other oral analgesics, with the exception of `cyclobenzaprine`, which is a muscle relaxant that relieves pain for those with muscle spasms.

For the ICD-9 410.0 query, the model correctly demonstrates the relationship with other myocardial infarction codes for other areas of the

[1] http://code.google.com/p/word2vec/

RX: hydrocodone	DX: 410.0 MI anterolateral wall
RX: vicodin	DX: 410.2 MI inferolateral wall
RX: apap	DX: 410.1 MI anterior wall
RX: oxycodone	DX: 410.7 subendocardial infarc
RX: norco	DX: 410.4 MI inferior wall
RX: tramadol	DX: 410.3 MI inferoposterior wall
RX: lortab	DX: 411.8 other ischemic HD
RX: oxycontin	DX: 411.0 post-MI syndrome
RX: cyclobenzaprine	PX: coronary bypass (1 artery)
RX: darvocet	PX: coronary bypass (3 arteries)
RX: bextra	DX: 412.0 previous MI
PX: treat head injury	**LAB: prothrombin time test**
DX: 803.30 skull fracture	PX: prothrombin time proc
DX: 852.31 subdural hemorrhage	LAB: prothrombin time test
DX: 801.62 skull fracture	RX: warfarin sodium
PX: treat sinus fracture	DX: V58.61 current anticoagulant use
DX: 803.05 skull fracture	RX: warfarin
DX: 803.46 skull fracture	LAB: prothrombin time test
DX: 852.42 extradural hemorrhage	RX: coumadin
DX: 800.42 skull fracture	RX: enoxaparin sodium
DX: 864.15 liver laceration	DX: 427.31 atrial fibrillation
DX: 800.61 skull fracture	PX: 453.50 chronic venous embolism

Table 4.2: Four example codes and the ten closest codes by cosine distance. Prefixes: RX = prescription, DX = ICD-9 diagnosis, PX = CPT or ICD-9 procedure, LAB: laboratory test.

heart. There is also a relationship between MI and two types of coronary bypass surgery.

The neighbors of a generic procedure code for `treat head injury` are a series of diagnosis codes referring to specific types of skull fractures and cranial hemorrhages. Other codes are for `treat sinus fracture` and a less directly related diagnosis code for `liver laceration`.

The `prothrombin time test` is used to measure the effectiveness of anti-coagulant medications to ensure the patient is on the proper dose. We see

it is associated with procedure codes for the same test, along with drugs and diagnoses codes for anticoagulants. The last two items are diagnoses that are indications for anticoagulant use.

Mikolov et. al. (Mikolov et al., 2013a) discuss how the embedded words often exhibit relationships that are consistent between pairs of words with a similar relationship. For example, when they perform the calculation `bigger - big + small`, the closest word by cosine distance is `smallest`. In other words, what word is similar to `small` in the same way that `big` is like `biggest`.

We see some of the same relationships in the EHR embeddings. One possible relationship is `drug` $\xrightarrow{treats}$ `diagnosis`. We can ask `adderall` $\xrightarrow{treats}$ `ADHD` as `prozac` $\xrightarrow{treats}$ `?`. Translating to vector notation, this is equivalent to the equation `ADHD - adderall + prozac`, for which the 3 closest codes are `depressive disorder`, `major depressive disorder`, and `psychotherapy services`. The results are very different from the codes closest to `prozac` itself, which are primarily other antidepressants. However, the preservation of these relationships is weaker than in the text case. `ADHD - adderall + simvastatin`, for example, returns `impaired fasting glucose` as the top diagnosis, rather than a more optimal relationship, like `hypercholesterolemia`. The relationship seems more effective within diseases of the same system, as `hypercholesterolemia - simvastatin + warfarin` is more concordant returning `mitral valve replacement` and `aortic valve replacement` as the top two matches. This discrepancy may be due to the difficulty of finding pairs of relationships that are direct analogues in the medical domain (i.e. `adderall` is not used to treat `ADHD` in the "same way" as `prozac` is used to treat depression).

It is also possible to use the distributed representations for more quantitative prediction tasks. We examine the task of predicting onset of atrial fibrillation and flutter (AFF). This dataset contains 8,054 patients, 3,762 of whom developed AFF. In order to generate features from the embeddings,

we do k-means clustering of the event embeddings where k = 1,000. Each embedding is mapped to one of the 1,000 clusters, and features represent the number of times an event in that cluster occurs. To compare, we create a dataset with a feature for each code that occurs in more than 1% of patients, including ICD-9 diagnosis hierarchies. In both datasets each feature is calculated over 4 windows indicating the counts over the last year before censor date, last 3 years, last 5 years, or over all time. Both datasets also contain features for gender, birth year, and aggregates of basic vitals (minimum, maximum, and mean of height, weight, blood pressure). In the end, the dataset using the individual codes has 25,587 features, and the skip-gram clusters has 4,123.

We used Random Forests as implemented in Weka[2], with 200 trees and 200 variables per split and 5-fold cross validation. Using area under the ROC curve as the performance metric, the individual codes dataset achieved 0.685 while the skip-gram dataset had a score of 0.688. The model with the skip-gram features has better performance, but the difference is not significant. Despite the loss of precision caused by using only the cluster memberships of the embeddings, the skip-gram-based model performs as well as the individual code-based model.

4.3 Discussion

Ontology construction has been a major theme of medical informatics for a long time. The relationships between entities in the health care system are vast, and ontologies are used for many purposes, from diagnosis to reimbursement to quality control. Attempts have been made for over 30 years to manually curate collections of codes that capture clinical practice (Schneeweiss et al., 1983). Many ontologies have some sort of structure, typically a hierarchy, between the elements. However, while the hierarchy

[2]http://www.cs.waikato.ac.nz/ml/weka

can be useful to a predictive model (Singh et al., 2014), it does not provide enough detail to estimate a complete ordering of distances between all pairs of events. The existing ontologies are not focused on providing the right granularity for predictive modeling. Distributed representations benefit from the data-driven approach in which the degree of granularity can be adjusted as needed.

The skip-gram model is quite impressive in extracting knowledge from the EHR data despite being completely unsupervised. It is able to organize drugs from the same class, diagnoses with similar causes, and procedures with similar etiologies. It can even do so across different types of events, and can be trained with hundreds of millions of patient events. However, there is room for improvement. As mentioned before, the model is not great at representing more complex relationships, like $\texttt{drug} \xrightarrow{treats} \texttt{diagnosis}$. This may be related to a problem encountered in text data, where a word can have multiple senses (Neelakantan et al., 2014). Borrowing additional techniques from NLP literature could help.

A limitation of the current implementation is an inability to take into account the data values that accompany lab tests and vital signs. It is informative to know that a person had a diagnostic test, but the test result obviously matters as well. Result values can be discretized and encoded as separate events (e.g. `blood-glucose-low`, `blood-glucose-normal`, `blood--glucose-high`), but the algorithm will not necessarily keep their distributed representations collinear.

While distributed representations have become popular in NLP (Mikolov et al., 2013b), they have scarcely been considered in coded medical data, despite the similar setup. One previous paper that discusses distributed medical codes is Tran et al. (2015). They examine an embedding that uses a modified restricted Boltmann machine to embed 1,005 ICD-10 codes for 7,578 patients. Their method does not take into account the order of the codes and doesn't use codes from different ontologies.

In addition to qualitative analysis and code clustering, there are other potential uses for distributed event representations. Though our dataset only contains ICD-9 codes, one could imagine a dataset with both ICD-9 and ICD-10 codes (Quan et al., 2005). Such a dataset would provide a data-driven method for creating automated ICD-9 to ICD-10 conversions, based on the shared codes from other ontologies. This would allow a predictive model using EHR data to use the entire history despite a change in coding scheme. Examples of this effect exist in the current model; CPT code 90760 for IV infusion was replaced by 96360, and both are near the top of each other's closest codes.

Another potential use of distributed representations is to compare the patterns of coding among different institutions. Differences in the relative distances between related codes would give an indication of patterns of use. However, care would need to be taken to control the initialization between runs. One could use the encodings learned at one institution as initialization for the other. This method could also improve the performance of predictive models transferred from one institution to another.

Performance in predictive models could also be improved by designing machine learning models that directly use the sequence of distributed representations over time. The clustering presented here is one way to utilize the distributed representation in prediction, but it does not take full advantage of all of the information by assuming that all events in a cluster are the same. Methods based on multivariate time series or recurrent neural networks could be modified to better take advantage of distributed representation for the purposes of patient-level prediction.

5 DIFFERENTIAL PRIVACY IN PHARMACOGENETICS

In the previous chapters, we have examined several types of data mining with EHR data. Now we move to the second theme of this thesis, and focus on privacy preservation via differential privacy. Differential privacy was introduced briefly in Chapter 2.4, and we will build upon those principles here and in future chapters.

We return to the examine warfarin prediction from (Consortium, 2009), but juxtaposed against the desire to preserve the privacy of the patients in the training set. While differential privacy offers guarantees on effect of a patient's record on the final output, it also has an effect on the attack scenario where an adversary is trying to reverse engineer a patient's genotype. Using warfarin as a running example allows us to examine not only the effect of differential privacy on model accuracy but also simulate adverse events were the model to be used.

This work was published as Fredrikson et al. (2014) and a version previously appeared in the thesis of Matt Fredrickson Fredrikson (2015).

5.1 Introduction

Pharmcogenetic models of warfarin (Consortium, 2009; Sconce et al., 2005; Fusaro et al., 2013; Anderson et al., 2007) or other medications use a patient's genotype, demographic background, and clinical history to guide an appropriate treatment course. These models are most often based on supervised machine learning techniques that derive models from previously-gathered clinical and genomic data about a patient population (the training cohort). The learned models can then be used by doctors to personalize treatment given the patient's known traits.

Prior works (Narayanan and Shmatikov, 2008, 2010) have shown cases of being able to de-anonymize users and other privacy risks when datasets

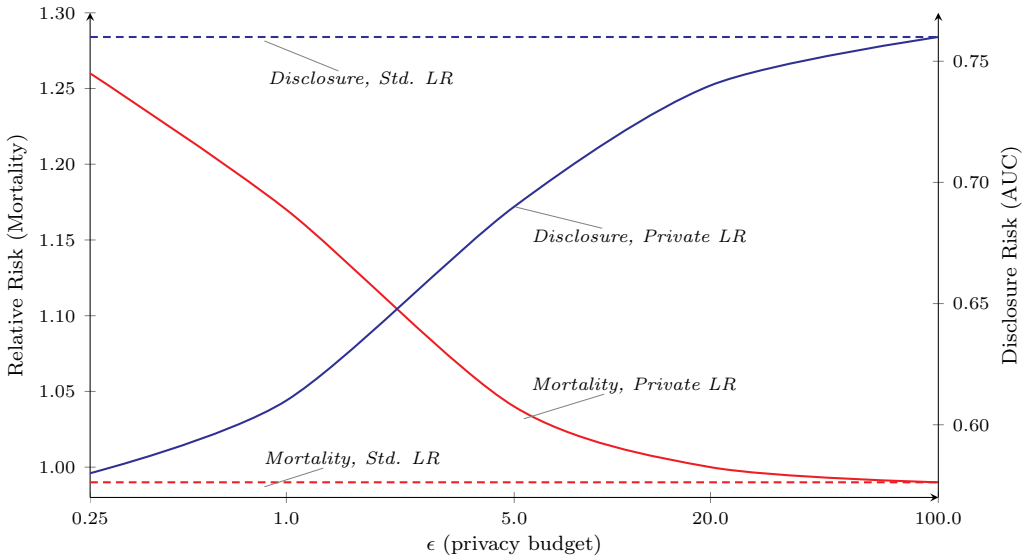


Figure 5.1: Mortality risk (relative to existing clinical procedure) for, and VKORC1 genotype disclosure risk of, ϵ-differentially private linear regression (LR) used for warfarin dosing for five values of ϵ (curves are interpolated). Horizontal dashed lines indicate mortality and disclosure risks for standard (non-private) linear regression. *Increased mortality risk is statistically significant* ($p \leqslant 0.04$) *at* $\epsilon \leqslant 5$.

are released or leaked. For biomedical research, datasets themselves are often only disclosed to researchers, yet the models learned from them are made public (e.g., published in a paper). Our focus is therefore on determining to what extent pharmacogenetic models themselves leak privacy-sensitive information, even in the absence of access to the original (private) dataset, and on whether privacy mechanisms such as differential privacy (Dwork, 2006) can be used to mitigate leaks, should they exist.

As discussed in previous chapters, warfarin is an anticoagulant widely used to help prevent strokes in patients suffering from atrial fibrillation and other cardiovascular ailments. However, it is known to exhibit a complex dose-response relationship affected by multiple genetic markers (Takeuchi et al., 2009), with improper dosing leading to increased risk of stroke

or uncontrolled bleed (Kuruvilla and Gurk-Turner, 2001; Sorensen et al., 2009). As such, a long line of work (Sconce et al., 2005; Fusaro et al., 2013; Anderson et al., 2007; Hamberg et al., 2007) has sought models that can accurately predict proper dosage and the response of the body to different levels of the drug.

Our study uses a dataset collected by the International Warfarin Pharmacogenetics Consortium (IWPC). While this particular dataset is publicly-available in a de-identified from (all participants in the studies gave informed consent about the release of de-identified data), it is equivalent to data sets used in other studies where the data must be kept private (e.g., due to lack of consent to release). We therefore use it as a proxy for a truly private dataset, and primarily focus on what privacy risks are associated with releasing models built form the dataset. The IWPC paper (Consortium, 2009) presents linear regression models from this dataset (see Tables 3.3 and 3.4), and shows that the resulting models that predict initial dose outperform the standard clinical regimen in terms of absolute distance from stable dose.

5.1.1 Model Inversion

To study the degree these models leak sensitive information about genotypes, we introduce *model inversion attacks*. In model inversion attacks, an adversary uses a machine-learning model to predict sensitive values of the input attributes. The attack takes advantage of the correlation between the target attribute, the model output, and other less sensitive attributes.

We apply model inversion to a linear model derived from the IWPC dataset and show that, when one knows a target patient's demographics and stable dosage, the inverted model performs significantly better at predicting the patient's genetic markers (up to 22% better) than guessing based on published marginal distributions.Moreover, the inverted model performs measurably better for members of the training cohort than others

(yielding an increased 5% AUC) indicating a leak of information specifically about those in the training cohort. The inversion attack is general, and is optimal in the sense of minimizing the attacker's misclassification rate given available information.

5.1.2 Effect of Differential Privacy

Differential privacy (DP) is a popular framework for statistical release mechanisms, and medical data is often mentioned potential target domain (Zhang et al., 2012; Dankar and El Emam, 2012; Vu and Slavkovic, 2009; Dwork, 2011). While differential privacy does not explicitly aim to protect attribute privacy, it would be a desirable characteristic, and some theoretical work indicates that noise can preserve attribute privacy in some settings (Kasiviswanathan et al., 2013b). We apply two recent algorithms to the IWPC dataset: the so-called *functional method* of Zhang et al. (2012) for producing private linear regression models, and Vinterbo's *privacy-preserving projected histograms* (Vinterbo, 2012) for producing differentially-private synthetic datasets, over which regression models can be trained. We chose these mechanisms as they represent the current state-of-the-art, and they apply two completely different approaches to achieving differential privacy—the former adds carefully-crafted noise to the coefficients of an objective function during optimization, and the latter is a more direct application of the well-known Laplace mechanism (Dwork, 2006). The question becomes whether there exist values of ϵ that meaningfully improve privacy yet do not degrade the utility of the pharmacogenetic application.

To answer this question, we perform the first end-to-end evaluation of DP in a medical application (Section 5.5). On one end we apply our model inverter to quantify the amount of information leaked about patient genetic markers by ϵ-differentially private versions of the IWPC model for a range of ϵ values. On the other end, we perform simulated clinical

trials via techniques widely used in the medical literature (Fusaro et al., 2013; Holford et al., 2010; Bonate, 2000; Holford et al., 2000) to quantify the impact of ϵ on patient outcomes.

Our main results, a subset of which are in Figure 5.1, show a clear trade-off between utility and privacy. While small values of ϵ do protect attribute privacy, it is not the case for larger values of ϵ (See "Disclosure, Private LR" in Figure 5.1). At higher values of ϵ, the attacker can predict well, with up to 58% accuracy (0.76 AUC), which is significantly better than the 36% accuracy one achieves without the models. Our simulated clinical trials reveal that for $\epsilon \leqslant 5$ the risk of fatalities or other negative outcomes increases significantly (up to $1.26\times$) compared to the current clinical practice, which uses non-personalized, fixed dosing and so leaks no information at all (See the line labeled "Mortality, private LR" in Figure 5.1). Our analyses show that, in this setting where utility is extremely important, there exists no value of ϵ for which differential privacy can be reasonably employed.

5.2 Background

5.2.1 Warfarin and Pharmacogenetics

Stable warfarin dose is assessed clinically by measuring the time it takes for blood to clot, called prothrombin time. This measure is standardized between different manufacturers as an international normalized ratio (INR). Based on the patient's indication for (i.e., the reason to prescribe) warfarin, a clinician determines a target INR range. After the fixed initial dose, later doses are modified until the patient's INR is within the desired range and maintained at that level. INR in the absence of anticoagulation therapy is approximately 1, while the desired INR for most patients in anticoagulation therapy is in the range 2–3 (Brace, 2001). INR is the re-

sponse measured by the physiological model used in our simulations in Section 5.5.

Polymorphisms in two genes, VKORC1 and CYP2C9, are associated with the mechanism with which the body metabolizes the drug, which in turn affects the dose required to reach a given concentration in the blood. Since each person has two copies of each gene, there are several combinations of variants possible. Following (Consortium, 2009), we represent VKORC1 polymorphisms by single nucleotide polymorphism (SNP) rs9923231, which is either G (common variant) or A (uncommon variant), resulting in three combinations G/G, A/G, or A/A. Similarly, CYP2C9 variants are *1 (most common), *2, or *3, resulting in 6 combinations.

5.2.2 Dataset

The dataset used is summarized in Chapter 3.2.1. While this dataset is publicly available, the type of data contained in the IWPC dataset is equivalent to many other medical datasets that have not been released publicly (Sconce et al., 2005; Hamberg et al., 2007; Anderson et al., 2007; Carlquist et al., 2006), and are considered private. We divided the data into two cohorts based on those used in IWPC (Consortium, 2009). The first (training) cohort was used to build a set of pharmacogenetic dosing algorithms. The second (validation) cohort was used to test privacy attacks as well as draw samples for the clinical simulations. To make the data suitable for regression we removed all patients missing CYP2C9 or VKORC1 genotype, normalized the data to the range [-1,1], scaled each row into the unit sphere, and converted all nominal attributes into binary-valued numeric attributes. Our eventual training cohort consisted of 2644 patients, and our validation cohort of 853 patients.

5.2.3 Privacy and Genetics

Prior studies in other contexts highlight the insufficiency of de-identification alone for ensuring privacy of datasets (Narayanan and Shmatikov, 2008; Datta et al., 2012; Frankowski et al., 2006). In the genomic context specifically, various works have shown that an individual's participation in a genome-wide association study (GWAS) could be accurately deduced from publication of detailed summary statistics (Homer et al., 2008; Sankararaman et al., 2009). The United States' National Institute of Health responded by changing their policy regarding public dissemination of datasets resulting from GWAS's (National Institutes of Health, 2013). The IWPC dataset is not from a GWAS and has distinct privacy concerns; data from clinical sources contains medical and oftentimes genetic information that is generally considered sensitive, and many datasets similar the IWPC's are kept private (Sconce et al., 2005; Hamberg et al., 2007; Anderson et al., 2007). Unlike patient datasets, pharmacogenetic models are generally made publicly-available. The IWPC paper published a complete description, for example, of their linear regressor, as did other studies (Carlquist et al., 2006; Anderson et al., 2007) for which the original dataset was not released.

5.3 Privacy of Pharmacogenetic Models

We consider a setting where an adversary is given access to a regression model, the warfarin dosage of an individual, some rudimentary information about the data set, and possibly some additional attributes about that individual. The adversary's goal is to predict one of the genotype attributes for that individual. In order for this setting to make sense, the genotype attributes, warfarin dose, and other attributes known to the adversary must all have been in the private data set. We emphasize that the techniques introduced can be applied more generally, but leave those

for future work.

5.3.1 Attack Model

We assume an adversary who employs an inference algorithm $\mathcal{A}$ to discover the genotype (in our experiments, either CYP2C9 or VKORC1) of a target individual α. The adversary has access to a linear model f trained over a dataset D drawn i.i.d from an unknown prior distribution p. D has domain $\mathbf{X} \times Y$, where $\mathbf{X} - X_1, ..., X_d$ is the domain of possible attributes and Y is the domain of the response. α is represented by a single row in D, $(\mathbf{x}^\alpha, y^\alpha)$, and the adversary's target attribute is $\mathbf{x}_t^\alpha$.

In addition, the adversary has access to $p_{1,...,d}$ and p_y from p[1], α's stable dose of warfarin y^α, some information π about the performance of f, and either of the following subsets of α's attributes:

- *Basic demographics*: a subset of α's demographic data, including age (binned into eight groups by the IWPC), race, height, and weight (denoted $x_{age}^\alpha, x_{race}^\alpha, \ldots$). Note that this corresponds to a subset of the non-genetic attributes in D.
- *All background*: all of α's attributes except CYP2C9 or VKORC1 genotype.

The adversary has black-box access to f.

5.3.2 Privacy of Disclosing De-identified Training Data

Before discussing the privacy risks of releasing a pharmacogenetic model, we first demonstrate the risks that follow from simply releasing a de-identified version of the original dataset D. This will serve as a point of

[1]These are often published along with the model or in other studies.

Algorithm 1: Inferring genotype given de-identified D.

Data: Dataset D, background $\mathbf{b}^{\mathrm{p}}$, dosage y^{p}
Result: Prediction of CYP2C9, VKORC1 variant for p

$\mathrm{D}_{\mathrm{p}} \leftarrow$ rows in D matching $\mathbf{b}^{\mathrm{p}}, \mathrm{y}^{\mathrm{p}}$;
if $|\mathrm{D}_{\mathrm{p}}| = 0$ **then**
    $\mathit{mode}_{\mathit{cyp2c9}} \leftarrow$ Most probable CYP2C9 polymorphism;
    $\mathit{mode}_{\mathit{vkorc1}} \leftarrow$ Most probable VKORC1 polymorphism;
    return $(\mathit{mode}_{\mathit{cyp2c9}}, \mathit{mode}_{\mathit{vkorc1}})$;
end
foreach *polymorphism* (ν_1, ν_2) *of CYP2C9, VKORC1* **do**
    $\mathit{matching}_{1,2} \leftarrow$ rows in D_{p} matching $\nu_{1,2}$;
    $\tilde{\mathrm{P}}_{\mathit{cyp2c9}}[\nu_1] \leftarrow |\mathit{matching}_1|/|\mathrm{D}_{\mathrm{p}}|$;
    $\tilde{\mathrm{P}}_{\mathit{vkorc1}}[\nu_2] \leftarrow |\mathit{matching}_2|/|\mathrm{D}_{\mathrm{p}}|$;
end
$\mathit{cyp2c9} \leftarrow \arg\max_{\nu} \tilde{\mathrm{P}}_{\mathit{cyp2c9}}[\nu_1]$;
$\mathit{vkorc1} \leftarrow \arg\max_{\nu} \tilde{\mathrm{P}}_{\mathit{vkorc1}}[\nu_2]$;
return $(\mathit{cyp2c9}, \mathit{vkorc1})$;

comparison in our discussion of linear models and their private counterparts. We answer the question: *if an adversary were given full access to the dataset, to what extent could he infer the genotype of patients in the dataset versus patients not in the dataset?* We answer this question quantitatively, under both of the background information assumptions discussed above: the adversary has only basic demographic information or the adversary knows everything but genotype. Our attack is similar to the linkage attacks explored by Narayanan and Shmatikov (2008) in the context of sparse dataset de-anonymization, using the empirical joint probability distribution on genotype given by D as a scoring function.

Our adversary, given the de-identified dataset D and some set of background information $\mathbf{b}^{\mathrm{p}}$ about the target p, makes full use of the information in D by computing the observed frequency of each polymorphism in rows with other columns matching the available background information. With the basic demographics, this means each row that has matching age, race, weight, height and stable dosage only, whereas with the full background

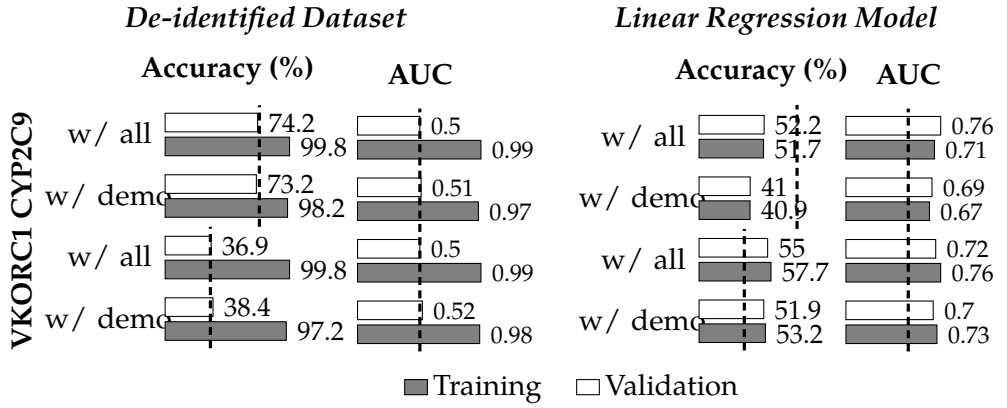


Figure 5.2: Adversary's inference performance given **(left)** raw training data and **(right)** a linear model derived from the training data. Here *w/ demo.* corresponds to inference given background demographic information, *w/ all* for inference with all attributes except the predicted genotype. Dashed lines represent baseline performance following random guessing according to prior distributions.

this means all rows excepting those for the two genetic markers. The adversary then simply picks the polymorphism with highest frequency count. This simple process is detailed in Algorithm 1.

To evaluate the algorithm, we use the IWPC dataset and its corresponding training/validation cohort split (see Section 5.2) denoted D_T and D_V, respectively. We then run Algorithm 1 on D_T with either background information setting for each patient p in D_T. We compute the CYP2C9 (resp. VKORC1) accuracy as the percentage of patients for which the algorithm correctly predicted CYP2C9 (resp. VKORC1). We also compute the area under the receiver-operator characteristic curve (AUC), following the multi-class AUC formulation given by Hand and Till (2001), which gives a more robust measure of prediction performance. Unlike accuracy, AUC is invariant to the underlying response variable distribution of the data on which it measures. This means that, for example, it will penalize an algorithm that performs well on skewed data by always guessing the

most probable response. Note that an AUC of 0.5 indicates "no better than random guessing" while a value of 1.0 indicates perfect classification. We repeat all the above for the validation set D_V.

The results are given in Figure 5.2 (**left**). The "w/ all" is the full background setting results and the "w/ demo" is the basic demographic background setting. Not surprisingly, the results show an obvious difference in the adversary's ability to infer genotype for patients in the study with knowledge of the training set, with accuracy close to 100% and AUC close to 1.0. These results indicate that the IWPC dataset exhibits a high degree of sparsity, making it possible to leverage partial information about a patient to infer their genomic information, which is in line with previous analyses of sparse datasets from other settings (such as the Netflix dataset (Narayanan and Shmatikov, 2008)). However, if the goal of the adversary is to infer genotype for an individual *not* in the training set, then Figure 5.2 (**right**) shows that regression models (discussed in the following section) yield better performance.

All this supports the conclusion that if genotype is considered sensitive in the dataset, then releasing it in de-identified form will still leak information.

5.3.3 Privacy of Disclosing Linear Models

In this section, we discuss techniques for inferring CYP2C9 and VKORC1 genotype from a linear regression model designed to predict warfarin dosing. Given a model M that takes inputs $\mathbf{x}$ and outputs a predicted stable dose y, the attacker seeks to build an algorithm $\mathcal{A}$ that takes as input some subset $\mathbf{b}$ of elements of $\mathbf{x}$, a known stable dose y, and outputs a prediction of one of the elements of $\mathbf{x}$ not in $\mathbf{b}$. In this way, $\mathcal{A}$ is partially inverting the regression model, and so we refer to the general attack technique as *model inversion* and $\mathcal{A}$ as an *inverter*. While we focus here on pharmacogenetics and using model inversion to predict a patient's genetic markers, our

techniques are applicable to any setting using linear models.

Background Recall that a linear regression model describes the relationship between a set of input variables $x_1, \ldots, x_n$ and a response variable y by a linear relation: $c_1 x_1 + \cdots + c_n x_n + d = y$ Thus, the model is described in full by its coefficients $\mathbf{c} = [c_1, \ldots, c_n]$ and intercept d. Given $\mathbf{x}^p = [x_1^p, \ldots, x_n^p]$ for patient p, one predicts the stable dosage y^p by applying the inner product of $(\mathbf{x}^p)^T$ (the transpose of $\mathbf{x}^p$) and $\mathbf{c}$, and adding d. Note that while all input variables are treated as real-valued when applying the model for prediction, linear regression is often used with nominal-valued input variables as well. For a nominal input variable taking k possible values, this is typically accomplished by breaking the variable into $k-1$ binary-valued variables, and assigning a value of 1 to at most one of the binary variables corresponding to the value of the nominal variable. For example, if a nominal variable x takes values from $\{a, b, c\}$, then we would create two binary variables x_a, x_b. To represent $x = a$, we would set $(x_a = 1, x_b = 0)$, whereas to represent $x = c$ we would use $(x_a = 0, x_b = 0)$. One of the nominal values (in this case x_c) corresponds to setting all binary variables to 0, rather than giving the value its own binary variable, in order to reduce the linear dependence between the binary variables.

Linear regression models are trained by solving a minimization problem over a training sample consisting of a set of input and response values $(\mathbf{x}^{p_1}, y^{p_1}), \ldots, (\mathbf{x}^{p_m}, y^{p_m})$:

$$\text{Find } \arg\min_{\mathbf{c},d} \sum_{1 \leqslant i \leqslant m} (y_{p_i} - (\mathbf{c} \cdot \mathbf{x}^{p_i})^T - d)^2 \tag{5.1}$$

Linear regression models rarely fit the data from which they were trained perfectly. In real settings, the model results in a certain amount of *residual*

error that is given by:

$$\mathbf{r} = [y^{p_1} - (\mathbf{c} \cdot \mathbf{x}^{p_1})^T - d, \ldots, y^{p_m} - (\mathbf{c} \cdot \mathbf{x}^{p_m})^T - d] \tag{5.2}$$

When a regression model is published, information about the residual error, such as the *residual standard error* (RSE), is typically given. In most applications of linear regression, the residuals are normally distributed about 0 with standard deviation equivalent to the RSE. We assume the adversary has knowledge of this quantity, and can thus estimate a distribution π on $\mathbf{r}$. With these quantities, the model inversion problem is stated as follows:

$$\begin{array}{ll} \textit{Given:} & p, \mathbf{b}^p, y^p, \mathbf{c}, d, \pi, P_1, \ldots, P_n, \\ \textit{Predict:} & \mathbf{x}^p_{cyp2c9}, \mathbf{x}^p_{vkorc1} \end{array} \tag{5.3}$$

where *cyp2c9*, *vkorc1* are the indices of the elements associated to the genetic markers and $\mathbf{b}^p$ corresponds to the background information given in either the demographic setting or full setting (as defined in Section 5.3.2). Recall from the Section 5.3.1 that $P_1, \ldots, P_n$ correspond to the prior probability distributions of each input variable in the training dataset. Note that $\mathbf{x}^p_{cyp2c9}, \mathbf{x}^p_{vkorc1}$ are nominal-valued, and so they actually correspond to two sets of binary variables (we omit this from our notation for simplicity).

Model inversion by maximized response likelihood In the following let $g \in \{cyp2c9, vkorc1\}$. We build a model inverter $\mathcal{A}_{mi}$ that predicts $\mathbf{x}^p_g$ by using the linear model M directly to compute a value that maximizes the probability of predicting the known response y^p given the background information $\mathbf{b}^p$. This means finding the polymorphism type v such that the following probability $\tilde{P}_{y^p}$ is maximized:

$$\tilde{P}_{y^p} = Pr[(\overline{\mathbf{x}}^p)^T \cdot \mathbf{c} + d + r = y^p \mid \overline{\mathbf{x}}^p_g = v, \overline{\mathbf{x}}^p_{bg} = \mathbf{b}^p] \tag{5.4}$$

where $\bar{\mathbf{x}}^p$ is a random variable over n-dimensional vectors with elements distributed independently according to the priors $P_1, \ldots, P_n$ and r is a random variable distributed according to the error distribution π. We condition on $\bar{\mathbf{x}}^p$ matching the adversarially-known background information as well as the target polymorphism value v.

Computing $\tilde{P}_g$ amounts to a straightforward convolution over the possible values of r and the input variables, excepting those included in $\bar{\mathbf{x}}^p_g$ and $\bar{\mathbf{x}}^p_{bg}$. In the following, let U denote the k variable indices not associated to the target genotype or the background information, and let $V = \{1, \ldots, n\} \setminus U$:

$$\begin{aligned}
\tilde{P}_{y^p} &= \frac{\Pr[(\bar{\mathbf{x}}^p)^\top \cdot \mathbf{c} + r = y_p - d, \bar{\mathbf{x}}^p_g = v, \bar{\mathbf{x}}^p_{bg} = \mathbf{b}^p]}{\Pr[\bar{\mathbf{x}}^p_g = v, \bar{\mathbf{x}}^p_{bg} = \mathbf{b}^p]} \\
&= \frac{\Pr[(\bar{\mathbf{x}}^p_U)^\top \cdot \mathbf{c}_U + r = y_p - d - (\bar{\mathbf{x}}^p_V)^\top \cdot \mathbf{c}_V] \cdot P_{v,bg}}{P_{v,bg}} \\
&= \Pr[(\bar{\mathbf{x}}^p_U)^\top \cdot \mathbf{c}_U + r = y_p - d - (\bar{\mathbf{x}}^p_V)^\top \cdot \mathbf{c}_V] \\
&= (c_{U_1} P_{U_1} * \cdots * c_{U_k} P_{U_k} * \pi)[y_p - d - (\bar{\mathbf{x}}^p_V)^\top \cdot \mathbf{c}_V]
\end{aligned}$$

where $P_{v,bg} = \Pr[\bar{\mathbf{x}}^p_g = v, \bar{\mathbf{x}}^p_{bg} = \mathbf{b}^p]$ and $*$ is the convolution operator. The second equality follows by assuming independence between the attributes. In reality these are usually not independent, but the adversary cannot do better than this assumption, having access only to independent priors on the vector components. Because we assume knowledge of $P_1, \ldots, P_n$ and π, we can easily compute the final right-hand-side quantity for each possible v of a small-domain nominal-valued variable $\mathbf{x}^p_g$ as long as the variables in $\bar{\mathbf{x}}_U$ are also small-domain nominal-valued. If these conditions do not hold, then this becomes a more challenging problem. Luckily, for us $\mathbf{x}^p_g$ corresponds to a genetic marker taking at most six values. For other numeric values, we discretize into a small number of bins.

The result is an inference algorithm that solves the problem stated

above. To evaluate its efficacy, we follow the same methodology as in Section 5.3.2 with the IWPC dataset: split it into D_T and D_V, use D_T to derive a linear model M as described above, and then run $\mathcal{A}_{mi}$ on every p in D_T with either of the two background information types. Using the same model M we then run $\mathcal{A}_{mi}$ on every p in D_V with either of the two background information types. The results are shown in Figure 5.2 (**right**). For both types of release, the adversary is able to better predict genotype (measured by AUC) than with baseline guessing (which yields AUC=0.5). However, this is not always the case when measuring performance as raw accuracy, where the regression model gives worse results on CYP2C9 (baseline accuracy is 75.6%). This means that the algorithm is worse at predicting CYP2C9 averaged over all patients, but better at predicting patients whose CYP2C9 genotype is not the most common variant. This is partially a result of the fact that our inference algorithm does not weight potential guesses by their baseline probability; because the objective function is conditioned on the the occurrence of each candidate, they are given equal prior weight. When we modified the algorithm to weight each candidate by its baseline probability as given by the prior P_i, we were able to infer CYP2C9 with approximately the same accuracy as the baseline guess, but AUC performance decreased to 0.54. This indicates that if accuracy is deemed more important than AUC performance, then our model inversion algorithm should weight its outputs according to genotype prior probabilities.

Another interesting result is that the regression model better predicts CYP2C9 for the validation set than the training set in all but one configuration. When we inspected this result more closely, it turned out to be a consequence of the particular training/validation split chosen by the IWPC. When we ran 100 instances of cross-validation, and measured the difference between CYP2C9 training and validation performance, we found that we were on average able to better predict the training cohort,

and the result held with high statistical significance ($p < 0.01$). However, we found that the superior training cohort performance for VKORC1 (measured by AUC) was *not* an unusual artifact of the IWPC training/-validation split, as 100 instances of cross-validation supported this result with similar significance. These results indicate that both the raw dataset, and linear regression models derived from it, pose both an *absolute* risk to patient privacy over baseline guessing, as well as a *relative* risk to those in the training cohort versus those in the validation cohort.

5.4 The Privacy of Differentially Private Pharmacogenetics

In the last section, we saw that non-private datasets and linear models trained on them leak information about patients in the training cohort. In this section, we explore the issue on models and datasets for which differential privacy has been applied. Applying differential privacy in practice, and understanding the meaning of its guarantees in context, can pose a challenge for several reasons. We attempt to address three commonly-cited issues (Lee and Clifton, 2011, 2012; Kifer and Machanavajjhala, 2011) in the context of pharmacogenetic warfarin dosing:

Which ϵ? Differential privacy does not purport to offer a notion of indistinguishability or semantic security; to provide some kind of utility, it gives an *approximate* guarantee related to a restricted notion of indistinguishability. ϵ is used to tune the degree of approximation, and must be configured by the owner of the dataset. What value of ϵ should one choose? Unless we understand the degree to which the types of disclosures are thwarted by different values of ϵ, it is difficult to make this judgement based on anything but the type of utility needed for the application.

Relative privacy The guarantee offered by differential privacy is relative to the disclosure risk of the application given *slightly less data*—if the risk of disclosing certain types of facts is high when the algorithm does not have access to a particular user's data, then it will only be slightly higher when given that user's data. Suppose that an analyst is considering releasing a statistical model of detailed genomic correlations drawn from a homogeneous population (e.g., an extended family, or small ethnic group). Obviously, without any type of privacy protection, the risk of disclosing a wide range of facts about the population's genome is high. If we wish to protect some of these facts, e.g., correlations considered sensitive by the participants, should we select a small ϵ, apply differential privacy, and release the model? Unfortunately, even with strong ϵ values, the risk is likely to remain high, as the contribution of any individual will not affect the output much.

Overfitting on sensitive attributes Statistical models are generally trained by maximizing their utility on one population sample available to the analyst at design time. This can lead to the well-known problem of *overfitting*, wherein the model ultimately describes unwanted deviations from the true population distribution that were present in the training sample. In a sense, one of the main goals of machine learning and statistical modeling lies in avoiding overfitting, but it remains a challenging problem. If some of the attributes used for training are sensitive, then the nature of overfitting suggests that a derived model might disclose specific information about the individuals in the training sample. Does differential privacy mitigate this problem in a useful way? If so, to what degree, and at which parameter settings? If it were generally the case, then either *i)* statisticians would apply techniques identical to differential privacy broadly, even when privacy is not a concern, or *ii)* existing algorithms developed to avoid overfitting would suffice to achieve privacy without the need

for differential privacy. Indeed, Dwork and Lei (2009) explore such a connection in their study of robust statistics and differential privacy.

In this section, we explore these issues empirically as they pertain to the use of differential privacy in pharmacogenetic Warfarin dosing. As in the previous section, we take the perspective of the adversary, and attempt to infer patients' genotype given differentially-private models and different types of background information on the targeted individual. As such, we use the same attack model from Section 5.3.1, but rather than assuming the adversary has access to D or M, we assume access to a *differentially private version of* D or M. We use two published differentially-private mechanisms with publicly-available implementations: *private projected histograms* (Vinterbo, 2012) and the *functional mechanism* (Zhang et al., 2012) for learning private linear regression models. Although full histograms are typically not published in pharmacogenetic studies, we analyze their privacy properties here to better understand the behavior of differential privacy across algorithms that implement it differently.

Our key findings are summarized as follows:

- While some ϵ values undoubtedly offer better privacy, measurable leakage occurred at all levels. Even at $\epsilon = 0.25$, private regression allowed a 16% increase in AUC performance over baseline (0.58 versus 0.5). *At the same ϵ, private histograms allowed a 44% increase over baseline AUC (0.72 versus 0.5), and a 52% increase in accuracy (55.2% versus 36.3%).*

- Private histograms disclose significantly more information about genotype than private linear regression, even at identical ϵ values. At all tested ϵ, private histograms leaked more on the training than validation set. *This result holds even for non-private regression models*, where the AUC gap reached 3.7% area under the curve, versus the 3.9% - 5.9% gap for private histograms. This demonstrates that the

relative nature of differential privacy's guarantee leads to meaningful concerns on real datasets.

- Disclosure due to overfitting occurs in both types of private models at a wide range of ϵ values. The difference in AUC between training and validation is statistically significant at *all* ϵ for private histograms, and $\epsilon > 1$ *for private linear regression in many settings*.

Our results indicate that understanding the implications of differential privacy for pharmacogenomic dosing is a difficult matter—even small values of ϵ might lead to unwanted disclosure in many cases.

5.4.1 Differentially-private histograms

We first investigate a mechanism for creating a differentially-private version of a dataset via the private projected histogram method. DP datasets are appealing because an (untrusted) analyst can operate with more freedom when building a model; he is free to select whichever algorithm or representation best suits his task, and need not worry about finding differentially-private versions of the best algorithms.

Let $\mathcal{X}_1, \ldots, \mathcal{X}_n$ be the domain of each attribute in the database D. Then $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is the domain of D, and we can model a *histogram* of D as a function $h_D : \mathcal{X} \mapsto \mathbb{R}$, where $h([x_1, \ldots, x_n])$ corresponds to the number of rows in D matching $[x_1, \ldots, x_n]$. The technique we use, *private projected histograms* (Vinterbo, 2012), constructs differentially-private versions of a dataset D by adding noise from the *Laplace distribution* with inverse scale parameter ϵ to the range of h_D. The density of this distribution is proportional to the zero-centered symmetric exponential distribution: $\text{Laplace}_\epsilon(x) \propto e^{-\epsilon|x|}$. The private projected histogram algorithm works as follows:

1. Construct a histogram h_D of D.

2. Construct a new histogram h_D^ϵ based on h_D:

$$h_D^\epsilon(\mathbf{x}) = h_D(\mathbf{x}) + \text{sample}(\text{Laplace}_\epsilon)$$

3. Filter noisy elements: set $h_D^\epsilon(x) \leftarrow 0$ for all x such that $h_D^\epsilon(x) \leqslant A\log(|D|)/\epsilon$.

4. Construct a new dataset D_ϵ by creating $\text{round}(h_D^\epsilon(\mathbf{x}))$ rows with value $\mathbf{x}$.

We use the author's publicly-available R implementation of this algorithm with $A = 0.5$, as this is the default specified in the implementation (and has been used in previous studies (Lei, 2011)). The algorithm also includes a pre-processing step that finds a subspace onto which D can be projected without sacrificing too much fidelity; this helps mitigate the potentially large computational expense of the algorithm. We do not use this functionality in our experiments, as our dataset has sufficiently small dimension to ensure reasonable computation times. We also note that the author's publicly-available implementation (and our description of it) does not include a sampling step described in the original paper (Vinterbo, 2012); this does not affect the privacy of the system, as it is a performance optimization.

Because numeric attributes are too fine-grained for effective histogram computation, we first discretize each numeric attribute into equal-width bins. In order to select the number of bins, we use a heuristic given by Lei (2011) and suggested by Vinterbo (2012), which says that when numeric attributes are scaled to the interval $[0, 1]$, the bin width is given by $(\log(n)/n)^{1/(d+1)}$, where $n = |D|$ and d is the dimension of D. In our case, this is implies two bins for each numeric attribute. We validated this parameter against our dataset by constructing 100 differentially-private datasets at $\epsilon = 1$ with $2, 3, 4$, and 5 bins for each numeric attribute, and measured the accuracy of a dose-predicting linear regression model over

each dataset. The best accuracy was given for $k = 2$, with the difference in means for $k = 2$ and $k = 3$ not attributable to noise. When the discretized attributes are translated into a private version of the original dataset, the median value from each bin is used to create numeric values.

In order to infer the private genomic attributes given a differentially-private version D_ϵ of a dataset, we can use an algorithm very close to Algorithm 1. The only difference arises due to the fact that numeric values in D_ϵ have been discretized and re-generated from the median of each bin. Therefore, the likelihood of finding a row in D_ϵ that matches any row in D_T or D_V is low. To account for this, we add a pre-processing step to Algorithm 1 that transforms each numeric attribute in the background information to the nearest median from the corresponding attribute used in the discretization step when generating D_ϵ. We then run Algorithm 1 as though D_ϵ were a non-private dataset. The results are discussed in Section 5.4.3.

5.4.2 Differentially-private linear regression

We also investigate use of the functional mechanism (Zhang et al., 2012) for producing differentially-private linear regression models. The functional mechanism works by adding Laplacian noise to the coefficients of Equation 5.1, thereby producing a noisy version of the *objective function*. This novel technique stands in contrast to the more obvious approach of directly perturbing the output coefficients of the regression training algorithm, which would require an explicit sensitivity analysis of the training algorithm itself. Instead, deriving a bound on the amount of noise needed for the functional mechanism involves a fairly simple calculation on the objective function (Zhang et al., 2012).

However, minor complications arise due to the fact that the noisy objective function may not be suitable for optimization, e.g., it may be unbounded. To address this, the authors propose two methods for ensuring

boundedness without sacrificing excessive utility, to be used in tandem. The first is based on *regularization*, a common strategy to avoid overfitting that adds a positive constant to the diagonal of the matrix representation of the objective function. The second method, *spectral trimming*, removes non-positive eigenvalues from the objective matrix. The details of these methods are not in scope, but are clearly explained in (Zhang et al., 2012), where the authors give empirical evidence of the superiority of their approach to previous methods on several datasets for ϵ values between 0.1 and 3.2.

We produce private regression models on the IWPC dataset by first projecting the columns of the dataset into the interval $[-1, 1]$, and then scaling the non-response columns of each row into the unit sphere. This is described in the paper (Zhang et al., 2012) and performed in the publicly-available implementation of the technique. It is necessary to ensure that sufficient noise is added to the objective function, i.e., the amount of noise needed is not scale-invariant. In order to inter-operate with the other components of our evaluation apparatus, we re-implemented the algorithm in R by direct translation from the authors' Matlab implementation. We evaluated the accuracy of our implementation against theirs, and found no statistically-significant difference.

Applying model inversion to the functional mechanism is straightforward, as our technique from Section 5.3.3 makes no assumptions about the internal structure of the regression model or how it was derived. However, care must be taken with regards to data scaling, as the functional mechanism classifier is trained on scaled data. When computing the convolution, all input variables must be transformed by the same scaling function used on the training data, and the predicted response must be transformed by the inverse of this function.

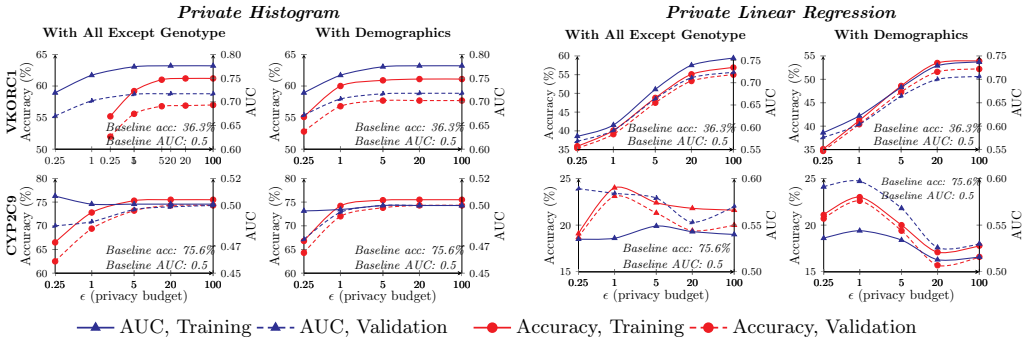


Figure 5.3: Inference performance for genomic attributes over IWPC training and validation set for private histograms (**left**) and private linear regression (**right**), assuming both configurations for background information. Dashed lines represent accuracy, solid lines area under the ROC curve (AUC).

	Alternative Hypothesis	***Genotype***	**p-value**	ϵ
DPLR	train. AUC > valid. AUC	VKORC1	< 0.02	$\geqslant 5.0$
		CYP2C9	> 0.95	*all*
	priv. AUC < non-pr. AUC	VKORC1	< 0.01	$\leqslant 5.0$
		CYP2C9	< 0.01	*all*
	train. AUC > baseline	VKORC1	< 0.01	*all*
		CYP2C9	< 0.01	*all*
DP Histogram	train. AUC > valid. AUC	VKORC1	< 0.01	*all*
		CYP2C9	< 0.01	*all*
	priv. AUC < non-pr. AUC	VKORC1	< 0.01	*all*
		CYP2C9	< 0.01	*all*
	train. AUC > baseline	VKORC1	< 0.01	*all*
		CYP2C9	> 0.99	*all*

Table 5.1: Summary statistics for genotype disclosure experiments. Alternative hypotheses are accepted when $p \leqslant 0.05$. Alternative hypotheses tested include whether AUC is greater on the training data than the validation data (for private versions of the algorithms), whether AUC on the training set is less for the private algorithm than non-private, and whether AUC is greater on the training set than baseline guessing from priors (for private versions of the algorithms).

5.4.3 Results

We evaluated our inference algorithms on both mechanisms discussed above at a range of ϵ values: 0.25, 1, 5, 20, and 100. For each algorithm and ϵ, we generated 100 private models on the training cohort, and attempted to infer VKORC1 and CYP2C9 for each individual in both the training and validation cohort. All computations were performed in R.

Table 5.1 summarizes some of the main conclusions reached by our experiments, and Figure 5.3 shows the performance results in detail. Unless stated otherwise, our hypothesis tests are one-tailed t-tests, and we consider a result significant whenever it reaches a significance level at or below 0.05. For example, the second hypothesis in Table 5.1 can be interpreted as: "Our ability to infer VKORC1 with private regression was worse (by AUC) than non-private regression when $\epsilon \leqslant 5.0$, with significance $p < 0.01$. Our experiments showed no evidence that inference was worse at $\epsilon > 5.0$." In these figures, we make several types of comparisons:

1. *Private vs. Non-Private*: We measure the difference in our ability to infer genotype in private versus non-private versions of the models. Measurements on non-private models indicate absolute disclosure risk, and differences between private and non-private indicate the degree to which differential privacy mitigates this risk.

2. *Model vs. Baseline*: We use the term *baseline disclosure risk* to indicate the ability of an adversary to infer genotype given only prior distributions on the attributes in the database. The baseline strategy corresponds to always guessing the *most likely* attribute value as given by the prior distribution. Differences in this measurement indicate the absolute leakage of the model.

3. *Training vs. Validation*: We measure the difference between our ability to infer genotype for those in the training set versus those in

the validation set. Measured differences indicate disclosure due to overfitting in the model.

Private Histograms vs. Linear Regression We found that private histograms leaked significantly more information about patient genotype than private linear regression models. The difference in AUC for histograms versus regression models is statistically significant for VKORC1 at all ϵ. As Figure 5.3 indicates, the magnitude of the difference from baseline is also higher for histograms when considering VKORC1, nearly reaching 0.8 AUC and 63% accuracy, while regression models achieved at most 0.75 AUC and 55–60% accuracy. As Table 5.1 shows, the AUC performance for VKORC1 was greater than the baseline for all ϵ. However, for CYP2C9 this result only held when assuming all background information except genotype, and only for $\epsilon \leqslant 5$; when we assumed only demographic information, there was no significant difference between baseline and private histogram performance. Private regression models performed better in terms of AUC for CYP2C9, but much worse in terms of accuracy. The shape of the performance curves for CYP2C9 on private regression is distinct from the other configurations. While we expect performance to generally improve as ϵ increases, the curves for this configuration behave non-monotonically. We suspect that the heavily skewed prior distribution of CYP2C9, and the presence of several extremely rare variants, are likely affecting the behavior of our model inverter in an unexpected way: 75% of patients have one variant, 22% have the next two most common variants, and the remaining variants account for approximately 0.7% - 1.3% of the patients. We tested this hypothesis by generating several datasets from random decision lists (Witten and Frank, 2005) that have one attribute whose distribution shares these properties, and witnessed similar behavior with the performance of the model inverter.

Disclosure from Overfitting In nearly all cases, we were able to better infer genotype for patients in the training set than those in the validation set. For private linear regression, this result holds for VKORC1 at $\epsilon \geqslant 5.0$. This is not an artifact of the training/validation split chosen by the IWPC; we ran 10-fold cross validation 100 times, measuring the AUC difference between training and test set validation, and found a similar difference between training and validation set performance ($p < 0.01$). The fact that the difference at certain ϵ values is not statistically significant is evidence that private linear regression is effective at preventing genotype disclosure at these ϵ. For private histograms, this result held for VKORC1 at all ϵ, and CYP2C9 at $\epsilon < 5$ with all background information but genotype. For CYP2C9, there were several cases in which the regression model gave higher AUC performance on the validation set (although accuracy performance was greater in all instances for $\epsilon > 5$). As shown in Figure 5.3, AUC is higher on the validation set for all ϵ in both configurations. Recall that this result was also present for non-private linear regression. We determined this to be an artifact of the training/validation split chosen by IWPC, as it did not persist across 100 instances of cross-validation, where we observed higher average performance on the training sets.

Differences in Genotype For both private regression and histogram models, performance for CYP2C9 is strikingly lower than for VKORC1. In terms of accuracy, performance when predicting CYP2C9 from private regression models was dramatically lower than baseline (17-25% versus 75.6% for baseline). This general result holds even for non-private regression models, although performance was higher (~50% accuracy). As previously discussed, this is partially a result of the fact that our inference algorithm does not weight potential guesses by their baseline probability. The underlying reason for this result is the highly-skewed distribution of CYP2C9 variants in the IWPC dataset, as discussed previously. The

skewed distribution means that a trivial classifier can achieve high accuracy for CYP2C9 by simply ignoring rare variants. The AUC performance shown in Figure 5.3 demonstrates that our inference mechanism does measurably better than such an algorithm at predicting individuals with rare variants.

5.5 The Cost of Privacy: Negative Outcomes

In addition to privacy, we are also concerned with the utility of a warfarin dosing model. The typical approach to measuring this is a simple accuracy comparison against known stable doses, but ultimately we're interested in how errors in the model will affect patient health. In this section, we evaluate the potential medical consequences of using a differentially-private regression algorithm to make dosing decisions in warfarin therapy. Specifically, we estimate the increased risk of stroke, bleeding, and fatality resulting from the use of differentially-private warfarin dosing at several privacy budget settings. This approach differs from the normal methodology used for evaluating the utility of differentially-private data mining techniques. Whereas evaluation typically ends with a comparison of simple predictive accuracy against non-private methods, we actually simulate the application of a privacy-preserving technique to its domain-specific task, and compare the outcomes of that task to those achieved without the use of private mechanisms.

5.5.1 Overview

In order to evaluate the consequences of private genomic dosing algorithms, we simulate a clinical trial designed to measure the effectiveness of new medication regimens. The practice of simulating clinical trials is well-known in the medical research literature (Fusaro et al., 2013; Holford et al., 2010; Bonate, 2000; Holford et al., 2000), where it is used to estimate

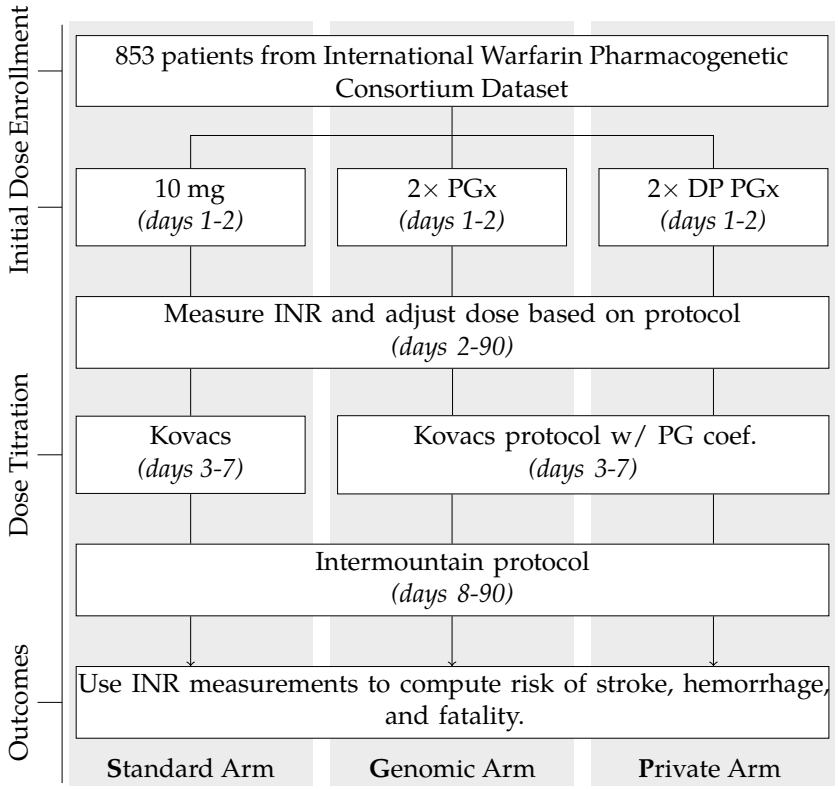


Figure 5.4: Overview of the Clinical Trial Simulation. *PGx* signifies the pharmacogenomic dosing algorithm, and *DP* differential privacy. The trial consists of three arms differing primarily on initial dosing strategy, and proceeds for 90 days. Details of Kovacs and Intermountain protocol are given in Section 5.5.3.

the impact of various decisions before initiating a costly real-world trial involving human subjects. Our clinical trial simulation follows the design of the CoumaGen clinical trials for evaluating the efficacy of pharmacogenomic warfarin dosing algorithms (Anderson et al., 2007), which is the largest completed real-world clinical trial to date for evaluating these algorithms. At a high level, we train a pharmacogenomic warfarin dosing algorithm and a set of private pharmacogenomic dosing algorithms on

the training set. The simulated trial draws random patient samples from the validation set, and for each patient, applies three dosing algorithms to determine the simulated patient's starting dose: the current standard clinical algorithm, the non-private pharmacogenomic algorithm, and one of the private pharmacogenomic algorithms. We then simulate the patient's physiological response to the doses given by each algorithm using a dose *titration* (i.e., modification) protocol defined by the original CoumaGen trial.

In more detail, our trial simulation defines three parallel *arms* (see Figure 5.4), each corresponding to a distinct method for assigning the patient's initial dose of warfarin:

1. *Standard:* the current standard practice of initially prescribing a fixed 10mg/day dose.

2. *Genomic:* Use of a genomic algorithm to assign the initial dose.

3. *Private:* Use of a differentially-private genomic algorithm to assign initial dose.

Within each arm, the trial proceeds for 90 simulated days in several stages, as depicted in Figure 5.4:

1. *Enrollment:* A patient is sampled from the population distribution, and their genotype and demographic characteristics are used to construct an instance of a *Pharmacokinetic/Pharmacodynamic (PK/PD) Model* that characterizes relevant aspects of their physiological response to warfarin (i.e., INR). The PK/PD model contains random variables that are parameterized by genotype and demographic information, and are designed to capture the variance observed in previous population-wide studies of physiological response to warfarin (Hamberg et al., 2007).

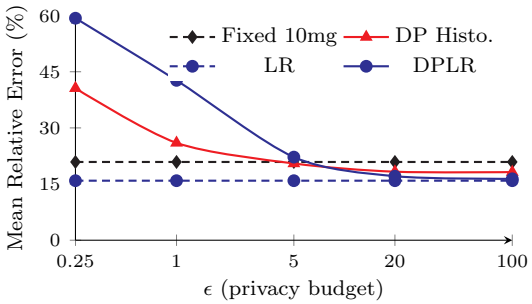


Figure 5.5: Pharmacogenomic warfarin dosing algorithm performance measured against clinically-deduced ground truth in IWPC dataset.

2. *Initial Dosing:* Depending on which arm of the trial the current patient is in, an initial dose of warfarin is prescribed and administered for the first two days of the trial.

3. *Dose Titration:* For the remaining 88 days of the simulated trial, the patient administers a prescribed dose every 24 hours. At regular intervals specified by the titration protocol, the patient makes "clinic visits" where INR response to previous doses is measured, a new dose is prescribed based on the measured response, and the next clinic visit is scheduled based on the patient's INR and current dose. This is explained in more detail in Sections 5.5.3 and 5.5.4.

4. *Measure Outcomes:* The measured responses for each patient at each clinic visit are tabulated, and the risk of negative outcomes is computed.

5.5.2 Pharmacogenomic Warfarin Dosing

To build the non-private regression model, we use regularized least-squares regression in R, and obtained 15.9% average absolute error (see Figure 5.5). To build differentially-private models, we use two techniques: the func-

tional mechanism (Zhang et al., 2012) and regression models trained on Vinterbo's private projected histograms (Vinterbo, 2012).

To obtain a baseline estimate of these algorithms' performance, we constructed a set of regression models for various privacy budget settings ($\epsilon = 0.25, 1, 5, 20, 100$) using each of the above methods. The average absolute predictive error, over 100 distinct models at each parameter level, is shown in Figure 5.5. Although the average error of the private algorithms at low privacy budget settings is quite high, it is not clear how that will affect our simulated patients. In addition to the magnitude of the error, its *direction* (i.e., whether it under- or over-prescribes) matters for different types of risk. Futhermore, because the patient's initial dose is subsequently titrated to more appropriate values according to their INR response, it may be the case that a poor guess for initial dose, as long as the error is not too significant, will only pose a risk during the early portion of the patient's therapy, and a negligible risk overall. Lastly, the accuracy of the standard clinical and non-private pharmacogenomic algorithms are moderate (~15% and 21% error, respectively), and these are the best known methods for predicting initial dose. The difference in accuracy between these and the private algorithm is not extreme (e.g., greater than an order of magnitude), so lacking further information about the correlation between initial dose accuracy and patient outcomes, it is necessary to study their use in greater detail. Removing this uncertainty is the goal of our simulation-based evaluation.

5.5.3 Dose Assignment and Titration

To assign initial doses and control the titration process in our simulation, we follow the protocol used by the CoumaGen clinical trials on pharmacogenomic warfarin dosing algorithms (Anderson et al., 2007). In the standard arm, patients are given 10-mg doses on days 1 and 2, followed by dose adjustment according to the Kovacs protocol (Kovacs et al., 2003) for

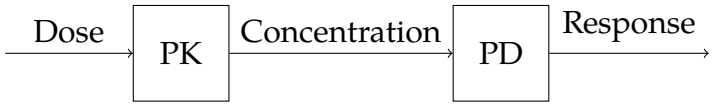


Figure 5.6: Basic functionality of PK/PD modeling.

days 3 to 7, and final adjustment according to the Intermountain Healthcare protocol (Anderson et al., 2007) for days 8 to 90. Both the Kovacs and Intermountain protocols assign a dose and next appointment time based on the patient's current INR, and possibly their previous dose.

The genomic arm differs from the standard arm for days 1-7. The initial dose for days 1-2 is predicted by the pharmacogenomic regression model, and multiplied by 2 (Anderson et al., 2007). On days 3-7, the Kovacs protocol is used, but the prescribed dose is multiplied by a coefficient C_{pg} that measures the ratio of the predicted pharmacogenomic dose to the standard 10mg initial dose:

$$C_{pg} = \frac{\text{Initial Pharmacogenomic Dose (mg)}}{5\text{ (mg)}}$$

On days 8-90, the genomic arm proceeds identically to the standard arm. The private arm is identical to the genomic arm, but the pharmacogenomic regression model is replaced with a differentially-private model.

To simulate realistic dosing increments, we assume any combination of three pills from those available at most pharmacies: 0.5, 1, 2, 2.5, 3, 4, 5, 6, 7, and 7.5 mg. The maximum dose was set to 15 mg/day, with possible dose combinations ranging from 0 to 15 mg in 0.5 mg increments.

5.5.4 PK/PD Model for INR response to Warfarin

A PK/PD model integrates two distinct pharmacologic models—pharmacokinetic (PK) and pharmacodynamic (PD)—into a single set of mathematical expressions that predict the intensity of a subject's response to drug ad-

ministration over time. *Pharmacokinetics* is the course of drug absorption, distribution, metabolism, and excretion over time. Mechanistically, the pharmacokinetic component of a PK/PD model predicts the *concentration* of a drug in certain parts of the body over time. *Pharmacodynamics* refers to the effect that a drug has on the body, given its concentration at a particular site. This includes the intensity of its therapeutic and toxic effects, which is the role of the pharmacodynamic component of the PK/PD model. Conceptually, these pieces fit together as shown in Figure 5.6: the PK model takes a sequences of doses, produces a prediction of drug concentration, which is given to the PD model. The final output is the predicted PD response to the given sequence of doses, both measures being taken over time. The input/output behavior of the model's components can be described as the following related functions:

$$\begin{aligned} \mathsf{PKPDModel}(\mathit{genotype}, \mathit{demographics}) &\mapsto \mathsf{F}_{\mathtt{inr}} \\ \mathsf{F}_{\mathtt{inr}}(\mathit{doses}, \mathit{time}) &\mapsto \mathit{INR} \end{aligned}$$

The function PKPDModel transforms a set of patient characteristics, including the relevant genotype and demographic information, into an INR-response predictor $\mathsf{F}_{\mathtt{inr}}$. $\mathsf{F}_{\mathtt{inr}}(\mathit{doses}, \mathtt{t})$ transforms a sequence of doses, assumed to have been administered at 24-hour intervals starting at $\mathit{time} = 0$, as well as a time t, and produces a prediction of the patient's INR at time t. The function PKPDModel can be thought of as the routine that initializes the parameters in the PK and PD models, and $\mathsf{F}_{\mathtt{inr}}$ as the function that composes the initialized models to translate dose schedules into INR measurements. For further details of the PK/PD model, see the appendix in the full version of the paper. [2].

Description	**Detail**	Value
Prior stroke		18%
Ischemic stroke probabilities		
w/No prior stroke		4.5%
w/Prior stroke		10.9%
RR INR in-range vs. all INRs	INR 2-3	0.46
RR INR outside range vs. in range	INR < 2.0	5.14
	INR > 3.0	0.73
Fatality resulting from stroke		
	INR $\geqslant 2.0$	8.1%
	INR < 2.0	17.5%
Bleeding probabilities		
ICH		
INR in-range	INR 2-3	0.25%
RR INR outside range	INR < 2.0	0.92
	INR > 3.0	4.31
Fatality from ICH	All INRs	51.6%
Major ECH		
Baseline probability	No treatment	0.2%
RR on warfarin	INR 2-3	2.22
RR INR outside range	INR < 2.0	2.14
	INR > 3.0	5.88
Fatality from Major ECH	All INRs	1.47%

Table 5.2: Annual cerebrovascular event probabilities for INR ranges. RR stands for relative risk, ICH for intra-cranial hemorrhage, ECH for extra-cranial hemorrhage. *Source:* Sorensen *et al.* Sorensen et al. (2009).

Alternative Hypothesis	Algorithm	p-value	ϵ
Reduced TTR	LR	> 0.99	–
	DPLR	$\leqslant 0.02$	$\leqslant 5$
	DP Histo.	< 0.01	$\leqslant 5$
Increased mortality	LR	> 0.99	–
	DPLR	$\leqslant 0.04$	$\leqslant 5$
	DP Histo.	$\leqslant 0.02$	$\leqslant 5$
Increased stroke	LR	> 0.99	–
	DPLR	< 0.01	$\leqslant 5$
	DP Histo.	$\leqslant 0.02$	$\leqslant 5$
Increased bleed	LR	$\geqslant 0.77$	–
	DPLR	$\leqslant 0.04$	$\leqslant 5$
	DP Histo.	< 0.01	$\leqslant 1$

Table 5.3: Selected summary statistics for risk estimates of pharmacogenetic algorithms versus fixed 10mg starting dose. p-values are given for rejection of the null hypothesis, which is the negation of the listed alternative hypothesis.

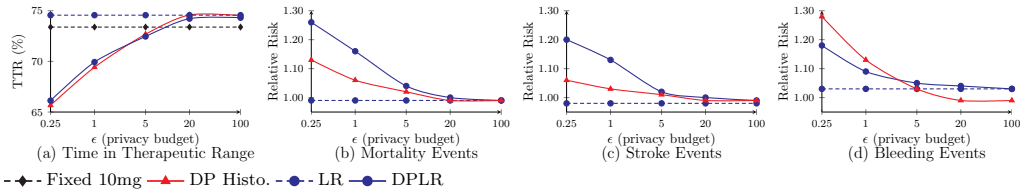


Figure 5.7: Trial outcomes for fixed dose, non-private linear regression (LR), differentially-private linear regression (DPLR), and private histograms. Horizontal axes represent ϵ.

5.5.5 Calculating Patient Risk

INR levels correspond to the coagulation tendency of blood, and thus to the risk of adverse events (Kuruvilla and Gurk-Turner, 2001). Sorensen *et al.* performed a pooled analysis of the correlation between stroke and bleeding events for patients undergoing warfarin treatment at varying INR levels (Sorensen et al., 2009). The probabilities of various events,

[2] Available at https://www.dropbox.com/s/ml2mepa8svsxxef/main.pdf

as reported in their analysis, are given in Table 5.2. We calculate each simulated patient's risk for stroke, intra-cranial hemorrhage, extra-cranial hemorrhage, and fatality based on the predicted INR levels produced by the PK/PD model. At each 24-hour interval, we calculated INR and the corresponding risk for these events. The sum total risk for each event across the entire trial period is the endpoint we use to compare the arms. We also calculated the mean *time in therapeutic range* (TTR) of patients' INR response for each arm. We define TTR as any INR reading between 1.8–3.2, to maintain consistency with previous studies (Fusaro et al., 2013; Anderson et al., 2007).

The results are presented in Figure 5.7 in terms of relative risk (defined as the quotient of the patient's risk for a certain outcome when using a particular algorithm versus the fixed dose algorithm), and Table 5.3 with summary statistics for relevant hypotheses. The results are striking: for reasonable privacy budgets ($\epsilon \leqslant 5$), private pharmacogenomic dosing results in greater risk for stroke, bleeding, and fatality events as compared to the fixed dose protocol. The increased risk is statistically significant (p-values given in Table 5.3) for both private algorithms up to $\epsilon = 5$ and all types of risk (including reduced TTR), except for private histograms, for which there was no significant increase in bleeding events with $\epsilon > 1$.

On the positive side, there is evidence that both algorithms may reduce all types of risk at certain privacy levels. Differentially-private histograms performed slightly better, improvements in all types of risk at $\epsilon \geqslant 20$. Private linear regression seems to yield lower risk of stroke and fatality and increased TTR at $\epsilon \geqslant 20$. However, the difference in bleeding risk for DPLR was not statistically significant at any $\epsilon \geqslant 20$. *These results lead us to conclude that there is evidence that differentially-private statistical models may provide effective algorithms for predicting initial warfarin dose, but only at low settings of* $\epsilon \geqslant 20$ *that yield little privacy (see Section 5.4.3).*

5.6 Related Work

The tension between privacy and data utility has been explored by several authors. Brickell and Shmatikov (2008) found strong evidence for a tradeoff in attribute privacy and predictive performance in common data mining tasks when k-anonymity, ℓ-diversity, and t-closeness are applied before releasing a full dataset. Differential privacy arose partially as a response to Dalenius' desideratum: *anything that can be learned from the database about a specific individual should be learnable without access to the database* (Dalenius, 1977). Dwork showed the impossibility of achieving this result while preserving a general notion of data utility (Dwork, 2006), and proposed an alternative goal that proved feasible to achieve in many settings: *the risk to one's privacy should not substantially increase as a result of participating in a statistical database*. Differential privacy is an attempt to formalize this goal, and constructive research on the topic has subsequently flourished.

There have also been critiques of differential privacy and its ability to meaningfully achieve Dwork's goal. Kifer and Machanavajjhala (2011) addressed several common misconceptions about differential privacy, and showed that under certain conditions, it fails to achieve a privacy goal seemingly related to Dwork's: *all evidence of an individual's participation should be removed*. Using hypothetical examples from social networking and census data release, they demonstrate that when rows in a database are correlated, or when previous exact statistics for a dataset have been released, this notion of privacy may be violated even when differential privacy is used. Part of our work extends theirs by giving a concrete examples from a real-world dataset where common misconceptions about differential privacy lead to surprising privacy breaches. We further extend their analysis by providing a quantitative study of the tradeoff between privacy and utility in a real-world application.

Others have studied the degree to which differential privacy leaks various types of information. Cormode showed that if one is allowed

to pose certain queries relating sensitive attributes to quasi-identifiers, it is possible to build a differentially-private Naive Bayes classifier that accurately predicts the sensitive attribute (Cormode, 2011). In contrast, we show that given a model for predicting a certain outcome from a set of inputs (and no control over the queries used to construct the model), it is possible to make accurate predictions in the *reverse* direction: predict one of the inputs given a subset of the other values. Lee and Clifton (2011) recognize the problem of setting ϵ and its relationship to the *relative* nature of differential privacy, and later (Lee and Clifton, 2012) propose an alternative parametization of differential privacy in terms of the *probability that an individual contributes to the resulting model.* While this may make the privacy guarantee easier for non-specialists to understand, its close relationship to the standard definition suggests that it may not be effective at mitigating the types of disclosures documented in this paper; evaluating its efficacy remains future work, as we are not aware of any existing implementations that support their definition.

The risk of sensitive information disclosure in medical studies has been examined by many. Wang et al. (2009), Homer et al. (2008), and Sankararaman et al. (2009) show that it is possible to recover parts of an individual's genotype given partial genetic information and detailed statistics from a GWAS. They do not evaluate the efficacy of their techniques against private versions of the statistics, and do not consider the problem of inference from a model derived from the statistics. Sweeny showed that a few pieces of identifying information are suitable to identify patients in medical records (Sweeney, 2000). Loukides et al. (2010a) show that it is possible to identify a wide range of sensitive patient information from de-identified clinical data presented in a form standard among medical researchers, and later proposed a domain-specific utility-preserving scheme similar to k-anonymity for mitigating these breaches (Loukides et al., 2010b). Dankar and El Emam (2012) discuss the use of differential

privacy in medical applications, pointing out the various tradeoffs between interactive and non-interactive mechanisms and the limitation of utility guarantees in differential privacy, but do not study its use in any specific medical applications.

5.7 Conclusion

We conducted the first end-to-end case study of the use of differential privacy in a medical application, exploring the tradeoff between privacy and utility that occurs when differentially-private algorithms are used to guide dosage levels in warfarin therapy. In this setting the tradeoff is between the *privacy of the patients involved in model creation* and the *safety of the patients to whom the model will be applied*; while privacy is certainly important here, *patient safety* is the tantamount concern.

Using a new technique called *model inversion*, we repurpose pharmacogenetic models to infer patient genotype. We showed that models used in warfarin therapy introduce a threat to patient privacy that is significant over the *baseline* threat following from knowledge of population-wide distributions of genotype. When models are produced to satisfy differential privacy, this effect holds to varying degrees depending on the choice of ϵ, but in most cases the models disclose sensitive information for all ϵ tested. Our model inversion technique is the first analysis to look at the problem of *unintentional* data leakage from a learned model.

We evaluated the *utility* of differential privacy by simulating clinical trials that use private models in warfarin therapy. This type of evaluation goes beyond what is typical in the literature on differential privacy, where raw statistical accuracy is the most common metric for evaluating utility. We show that differential privacy substantially interferes with the main purpose of these models in personalized medicine: for strong ϵ values, the risk of negative patient outcomes increases beyond acceptable levels.

Our results are strong evidence that there does not exist an ϵ for which differential privacy is effective with this application. More generally, our work provides a framework for assessing the tradeoff between privacy and utility for differential privacy mechanisms in a way that is directly meaningful for specific applications. For settings in which certain levels of utility performance *must* be achieved, and this tradeoff cannot be balanced, then alternative means of protecting individual privacy must be employed.

6 DIFFERENTIALLY PRIVATE INDUCTIVE LOGIC PROGRAMMING

In Section 3.1, we looked at inductive logic programming (ILP) in medical prediction. In this chapter, we focus on issues that arise in producing an ILP algorithm that satisfies differential privacy. This chapter was previously published as Zeng et al. (2014) and appeared in the thesis of Chen Zeng Zeng (2013).

6.1 Introduction

Given an encoding of a set of examples represented as a logical database of facts, an ILP algorithm will attempt to derive a hypothesized logic program which entails all the positive and none of the negative examples. Developing efficient algorithms for ILP has been widely studied by the machine learning community (Muggleton and de Raedt, 1994). However, to the best of our knowledge, a differentially private approach to ILP has received little attention.

ILP induces hypotheses from examples collected from individuals and to synthesize new knowledge from the examples. This approach naturally creates a privacy concern — how can we be confident that publishing these hypotheses and knowledge does not violate the privacy of the individuals whose data are being studied? This problem is compounded by the fact that we may not even know what data the individuals would like to protect nor what side information might be possessed by an adversary. These compounding factors are exactly the ones addressed by *differential privacy* (Dwork, 2006), which intuitively guarantees that the presence of an individual's data in a dataset does not reveal much about that individual. Differential privacy has previously been explored in the context of other machine learning algorithms (Williams and McSherry, 2010; Rubinstein

et al., 2009), including other chapters in this thesis. Accordingly, in this chapter we explore the possibility of developing differentially private ILP algorithms. Our goal is to guarantee differential privacy without obliterating the utility of the algorithm.

An obvious but important observation is that privacy is just one aspect of the problem; utility also matters, just as in the previous chapter. We quantify the utility of a differentially private ILP algorithm by its likelihood to produce a sound result. Intuitively speaking, "soundness" requires an algorithm to include a hypothesis that is correct in a sufficiently large subset of the database. We start by showing the trade-off between privacy and utility in ILP. Our result unfortunately indicates that the problem is very hard — that is, in general, one cannot simultaneously guarantee high utility and a high degree of privacy in ILP.

However, a closer investigation of this negative result reveals that if we can either reduce the hypotheses space or increase the size of the input data, then perhaps there is a differentially private ILP algorithm that is able to produce a high quality result while guaranteeing privacy. To verify this, we implement a differentially private ILP algorithm and run experiments on a synthetic dataset. Our results indicate that our algorithm is able to produce results of high quality while guaranteeing differential privacy when those two conditions are met.

6.2 Preliminaries

In ILP, as discussed in Section 2.3.2, we try to learn a theory in first-order logic that identifies positive examples, but avoids negative examples, using available background knowledge. Note that in the literature of differential privacy (Dwork, 2006), the terminology of "background knowledge" is different from the context in ILP and denotes the side information an adversary possesses to attack the privacy of a specific individual in the

underlying database. To prevent that confusion, we use the term "database" to represent the background knowledge shown in Definition 2.1. In the rest of this paper, we use $\|D\|$ to represent the number of individuals in a database D. We also refer $\mathcal{L}_2$ as the hypotheses space.

Recall Differential Privacy from Section 2.4. Here we use results from the study of *count queries* (Zeng et al., 2012), which refers to the number of examples in the dataset that match a given query. Since count queries only make sense as integers, we examine the geometric mechanism rather than the Laplacian mechanism.

Theorem 6.1. *Given* d *count queries* $\mathbf{q} = \langle q_1, \ldots, q_d \rangle$*, for any database* τ*, the database access mechanism:* $A_{\mathbf{q}}(\tau) = \mathbf{q}(\tau) + \langle \Delta_1, \ldots, \Delta_d \rangle$ *where* Δ_i *is drawn i.i.d from the geometric distribution* $G(\epsilon/S_{\mathbf{q}})$ (2.4)*, guarantees* ϵ*-differential privacy for* $\mathbf{q}$*.*

6.3 Problem Formulation

In analogy to Definition 2.2, we formulate the problem of guaranteeing differential privacy to ILP in Definition 6.2.

Definition 6.2. *(Differentially Private ILP): An ILP algorithm f is* ϵ*-differentially private iff for any pair of neighboring databases* [1] D_1 *and* D_2*, for any* $H \in \mathcal{L}_2$*.*

$$\Pr(f(D_1) = H) \leqslant e^{\epsilon} \Pr(f(D_2) = H)$$

By Definition 6.2, the output hypothesis does not necessarily satisfy the three requirements stated in Definition 2.1. The reason is that by

[1] In the differential privacy literature, databases are typically thought of as single tables. In a multi-relational setting, the proper definition of "neighboring" might change. For example, in a medical domain a neighboring database would remove one patient along with their respective prescriptions and diagnoses from other tables.

Definition 2.2, any differentially private algorithm must be randomized in nature.

6.4 Trade-off on Privacy and Utility

Although privacy is a very important problem in ILP, utility also matters; a trivial differentially private ILP algorithm can be generated by randomly outputting a hypothesis regardless of the underlying database. Though private, this algorithm is useless in practice. Therefore, we also need to quantify the utility of a hypothesis.

6.4.1 Our Utility Model

Our intuition for the utility model is to relax the requirements on hypotheses in Definition 2.1. In particular, we relax both the *validity* and *completeness* requirement, and introduce the notion of δ-*usefulness* ($0 \leqslant \delta \leqslant 1$).

Definition 6.3. *(δ-usefulness): A hypothesis* H *is* δ-useful *for the input database* D *iff* $\exists D' \subseteq D$, *and* $\|D'\|/\|D\| \geqslant \delta$ *such that*

1. *Approximate validity:* $\forall h \in H$, $h \in M^+(D')$.
2. *Approximate completeness: if a general clause* g *is true in* M^+D', *then* $H \models g$.
3. *Minimality: there is no subset of* H *which is validate and complete in* D'.

The notion of δ-usefulness quantifies the quality of a hypothesis in terms of the percentage of input database in which that hypothesis is correct. Furthermore, we define the quality of a differentially private ILP algorithm by its likelihood η to produce hypotheses of high quality. This is shown in Definition 6.4.

Definition 6.4. *((δ,η)-approximation): An ILP algorithm f is* (δ,η)-approximate *iff for any input database* D*,*

$$\Pr(f(D) \textit{ is } \delta\textit{-useful}) \geqslant 1-\eta$$

Both δ and η should be within the range of $(0,1)$ by definition. Another way to understand the notion of (δ,η)-approximation is through the idea of PAC-learning Valiant (1984) and define the notion of "approximate correctness" in terms of δ-usefulness. Next, we will quantify the trade-off between privacy and utility in ILP.

6.4.2 A Lower Bound on Privacy Parameter

Our techniques to prove the lower bound on the privacy parameter come from differentially private itemset mining Zeng et al. (2012). Perhaps this is no surprise since both frequent itemset mining and association rule mining have been closely connected with the context of ILP Dehaspe and Raedt (1997) in which frequent itemset mining can be encoded as an ILP problem. We prove the lower bound on the privacy parameter ϵ if an ILP algorithm must be both ϵ-differentially private and (δ,η)-useful. This is shown in Theorem 6.5.

Theorem 6.5. *For any ILP algorithm that is both ϵ-differentially private and (δ,η)-useful,*

$$\epsilon \geqslant \frac{\ln(\|2^n\|)(1-\eta)}{2((1-\delta)\|D\|+1)}$$

where n is the number of atoms in the language of hypotheses $\mathcal{L}_2$.

Proof. We model the language of the input database $\mathcal{L}_1$ as follows: each atom is taken from the set $I = \{a_1,\ldots,a_n\}$, and each individual's data is represented by a conjunctive clause of the atoms. We also model the

language of the hypotheses $\mathcal{L}_2$ to be all the possible conjunctive clauses over the set of atoms I.

Suppose f is an ILP algorithm that is both ϵ-differentially private and (δ, η)-useful. To better understand our proof technique, we add another atom a_{n+1} to I, and then we construct an input database D of size m by including $\delta * m$ clauses of the form $h_1 = a_1 \wedge a_2 \wedge \ldots \wedge a_n \wedge a_{n+1}$. The rest are constructed as simply $h_2 = a_{n+1}$. Since the number of all the hypotheses including a_{n+1} is 2^n, there must exist a particular hypothesis h_3 such that

$$\Pr(f(B) = h_3) \leqslant \frac{1}{2^n}$$

Without loss of generality, let $h_3 = a_1 \wedge a_2 \wedge \ldots a_k \wedge a_{n+1}$. Then, we construct another database D' from D by replacing one clause of h_1 by h_3, and then every clause of h_2 by h_3. Thus, there is a total number of $\delta m - 1$ clauses of h_1 in B' and the rest of them being h_3. It is not hard to show that h_3 is the only δ-useful hypothesis in B: any subset of B of cardinality δm must contain at least one h_3, and thus, the δ-valid hypotheses are those that can be entailed by h_3. Hence,

$$\Pr(f(D') = h_3) \geqslant 1 - \eta$$

Since D' and D differ by $2((1-\delta)m + 1)$ rows (one can think of this difference as the edit distance between two databases), by differential privacy,

$$1 - \eta \leqslant \frac{e^{\epsilon(2((1-\delta)m+1))}}{2^n}$$

Theorem 6.5 then follows. □

The result of Theorem 6.5 is similar in flavor to Ullman (2012), which proved that there is no differentially private algorithm that is able to answer $O(n^2)$ count queries in a database of size n with reasonable accuracy. That is, if an ILP algorithm can be thought of as a sequence of count queries,

and if the number of count queries exceeds a certain threshold, then the ILP algorithm cannot produce a result of high quality.

This is a discouraging result, which states that in general, it is very hard to simultaneously guarantee both differential privacy and a high utility requirement since $\|\mathcal{L}_2\|$ grows exponentially with the number of atoms. Theorem 6.5 suggests that in order to decrease the lower bound on the privacy parameter, we must either increase the size of the database $\|D\|$, or reduce the number of atoms in the hypotheses space $\mathcal{L}_2$. If a real world problem meets those two conditions, we might be able to get results of high quality while guaranteeing differential privacy. To verify this, we propose a differentially private ILP algorithm.

6.5 Differentially Private ILP Algorithm

In this section, we will first briefly describe a typical non-private ILP algorithm, inverse entailment as implemented in Aleph (Srinivasan, 2001), and then show our revisions of the non-private algorithm to guarantee differential privacy. In the rest of this section, we follow the terminologies used in Aleph in which an atom is also called a "predicate."

6.5.1 A Non-Private ILP Algorithm

The non-private ILP algorithm works as follows:

1. Select an example (selection): Select an example in the database to be generalized.

2. Build most-specific-clause (saturation (Muggleton, 1995)): Construct the most specific clause that entails the example selected, and is within language restrictions provided. This is usually a definite clause with many literals, and is called the "bottom clause."

3. Search (reduction): Find a clause more general than the bottom clause. This is done by searching for some subset of the predicates in the bottom clause that has the "best" score.

4. Remove redundant (cover removal): The clause with the best score is added to the current hypothesis, and all examples made redundant are removed.

A careful analysis of the above steps shows that the selection and reduction steps directly utilize the input data while the saturation and cover removal steps depend on the output from the previous step. Thus, as discussed in literature (Dwork, 2006), as long as we can guarantee the output from both selection and reduction is differentially private, then it is safe to utilize those output in subsequent computation. Hence, we only need to consider the problem of guaranteeing differential privacy for those two steps.

The input to the learning algorithm consists of a *target predicate*, which appears in the head of hypothesized clauses. The input database can be divided into two parts: the set of positive examples $E^+ \subseteq D$ which satisfy the target predicate, and the set of negative examples $E^- \subseteq D$ which do not. Furthermore, the bottom clause is normally expressed as the conjunctive form of the predicates, and thus we also use a "subset of the predicates" to denote the clause that is of the conjunctive form of the predicates in that subset.

6.5.2 A Differentially Private Selection Algorithm

The non-private selection algorithm is a sampling algorithm that randomly selects an individual's data to generalize. However, as discussed in (Dwork et al., 2006), no sampling algorithm is differentially private. In this paper, we propose to use domain knowledge to overcome this obstacle. That is, we utilize the domain knowledge to generate a "fake" example. We

want to emphasize that the domain information might come from external knowledge or previous interactions with the database. This information does not weaken the definition of differential privacy as stated in Definition 2.2, and we only utilize these previous known information to generate a fake example. In the worst case, this example can be expressed as the conjunction of all the predicates, which is considered as the public information. In that way, the new selection step hardly relies on the input database, and thus, it is differentially private[2].

6.5.3 A Differentially Private Reduction Algorithm

The non-private reduction algorithm actually consists of two steps: 1) the heuristic method to search for a clause, which is a subset of predicates in the bottom clause, and 2) the scoring function to evaluate the quality of a clause. Although there are many different methods to implement the reduction algorithm (Muggleton and de Raedt, 1994), in this paper we follow the standard usage in Aleph in which the heuristic method is a top-down breadth-first search and the scoring function is coverage (the number of covered positive examples minus the number of covered negative examples). The search starts from the empty set and proceeds by the increasing order of the cardinality of the clauses until a satisfying clause is found. The pseudocode of the non-private reduction algorithm is shown in Algorithm 2.

6.5.3.1 The Non-Private Algorithm

To better understand the non-private algorithm, we shall use an example to show how the top-down breadth-first search works. Suppose the bottom

[2] An alternative is to relax the privacy definition from ϵ-differential privacy to (ϵ, δ)-differential privacy. In this context, δ (unlike the symbol's use in section 6.4) refers to the probability that the algorithm violates the ϵ-differential privacy guarantee. We do not explore it here as it makes the already burdensome utility bounds much worse.

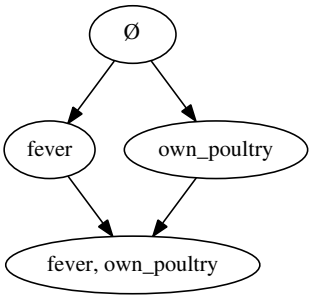


Figure 6.1: The Lattice for Clauses

clause is:

has_H9N7(A) :- fever(A), own_poultry(A)

which states that if an individual has suffered from a fever and owned certain kinds of poultry, then she is very likely to have suffered from a H9N7 virus. This bottom clause consists of two predicates, "fever" and "own_poultry." Since we are interested in the subset of the predicates in the bottom clause, we can define an order to do so. That is, for two clauses consisting of two subsets, $h_1 \preccurlyeq h_2$ iff $h_1 \subseteq h_2$, and thus, the hypotheses space can be formalized by a lattice shown in Figure 6.1.

The top-down breadth-first search algorithm starts from the clause of the empty set, which means every individual suffers from H9N7 in the underlying database, and then proceeds by the increasing order of the cardinality of the clauses until a satisfying clause is found. The quality of a clause is measured by both the number of positive examples satisfying the given clause, and the number of the negative examples violating the clause. The pseudocode of the non-private reduction algorithm is shown in Algorithm 2.

6.5.3.2 A Naïve Differentially Private Algorithm

We observe that the only part in Algorithm 2 that needs to inquire the input database is in the computation of P and N shown in line 2 and line 2,

Algorithm 2: Non-Private Reduction

Input: positive evidence E^+; negative evidence E^-; bottom clause H_b
Output: the best clause
begin
  k = number of predicates in H_b;
  $\mathcal{L}$ = the lattice on the subset of predicates in H_b;
  for $i = 1$ *to* k **do**
    for *each set* $S \in \mathcal{L}$, $\|S\| = i$ **do**
      P = the number of positive examples satisfying S;
      N = the number of negative examples satisfying S;
      $H^* = S$ if S has better coverage than the previously best clause;
    end
  end
  return H^*;
end

respectively. Therefore, as long as we can guarantee differential privacy to those two computations, then the reduction algorithm is differentially private. We do so by utilizing the geometric mechanism. Given a clause h, let q_h^+ and q_h^- be the queries that compute the number of positive examples satisfying h and the number of negative examples satisfying h, respectively. Then, as shown in Theorem 6.6, the sensitivity to evaluate a set of clauses is equal to the number of clauses in the set.

Theorem 6.6. *Given a set of clauses* $H = \{h_1, h_2, \ldots, h_n\}$*, and the corresponding evaluation queries* $\mathbf{q} = \{q_{h_1}^+, q_{h_1}^-, \ldots, q_{h_n}^+, q_{h_n}^-\}$*, the sensitivity of* $\mathbf{q}$ *is* n*.*

Proof. When we add an example, it either satisfies the clause or not, and is either positive or negative, therefore the sensitivity of $\mathbf{q}$ is at most n. Without loss of generality, suppose we add a positive example satisfying the clause. By adding an individual whose data is exactly the bottom

clause, the result of every $q^+_{h_i}$ increases by one. The theorem then follows. □

We show our differentially private reduction algorithm in Algorithm 3.

Algorithm 3: Diff. Private Reduction

Input: positive evidence E^+; negative evidence E^-; bottom clause H_b; privacy parameter ϵ
Output: the best clause
k = number of predicates in H_b;
$\mathcal{L}$ = build a lattice on the subset of predicates in H_b;
for i = *1 to k* **do**
  for *each subset* $h \in \mathcal{L}$, $\|S\| = i$ **do**
    $P' = q^+_h(E^+) + G(\epsilon/\|\mathcal{L}\|)$;
    $N' = q^+_h(E^-) + G(\epsilon/\|\mathcal{L}\|)$;
    $H^* = S$ if S has better coverage than the previously best clause w.r.t. P' and N';
  end
end
return H^*;

By Theorem 6.6, we can prove Algorithm 3 is differentially private. This is shown in Theorem 6.7.

Theorem 6.7. *Algorithm 3 is ϵ-differentially private.*

6.5.3.3 The Smart Differentially Private Reduction Algorithm

We observe that Algorithm 3 has only considered the worst-case scenario in which the number of clauses to be evaluated is the whole lattice whereas in practice, the reduction algorithm seldom goes through every clause in the lattice. This occurs when criteria are set to specify unproductive clauses for pruning (preventing evaluation of supersets) or for stopping the algorithm. Thus, the number of clauses evaluated in practice is much less than that in the whole lattice, which means the scale of the noise

added is larger than necessary. If the quality of a clause does not meet certain criterion, then there is no need to evaluate the subtree in the lattice rooted at that clause. This algorithm is shown in 4.

Algorithm 4: Smart_Diff_Private_Reduction

Input: positive evidence E^+; negative evidence E^-; bottom clause H_b; privacy parameter ϵ; levels ℓ
Output: the best clause
$\mathcal{L}$ = build a lattice on the subset of predicates in H_b;
for $i = 0$ *to* ℓ **do**
  β = the number of clauses with k predicates existing in the lattice;
  for *each subset* $h \in \mathcal{L}$, $\|S\| = i$ **do**
    $P' = q^+_h(E^+) + G(\frac{\epsilon}{\beta(\ell+1)})$;
    $N' = q^+_h(E^-) + G(\frac{\epsilon}{\beta(\ell+1)})$;
    $H^* = S$ if S has better coverage than the previously best clause;
    if P' *and* N' *does not meet the criterion* **then**
      Delete the subtrees in the lattice rooted at S;
    end
  end
end
return H^*

We have also introduced another parameter ℓ in Algorithm 4 to specify the maximal cardinality of the desired clause, which also helps to reduce the number of clauses to be evaluated. We prove Algorithm 4 is differentially private in Theorem 6.8.

Theorem 6.8. *Algorithm 4 is ϵ-differentially private*

Proof. By Theorem 6.6, the computation for each level is $(\epsilon/(\ell+1)$-differentially private, and since there is at most $\ell + 1$ levels to compute, by the composition property of differential privacy in Theorem 2.4, Theorem 6.8 then follows. □

6.5.4 Our Differentially Private ILP Algorithm

By using our differentially private selection algorithm and the smart differentially private reduction algorithm, we present our differentially private ILP algorithm in Algorithm 5. Since the output might consist of multiple clauses, we add the input parameter k which specifies the maximal number of clauses in the theory. We understand that this is not a usual setting for the usage of Aleph. However, in order to guarantee differential privacy, the most convenient way is to utilize the composition property which needs to know the number of computations in advance. We leave the problem to overcome this obstacle for future work.

By the composition property of differential privacy, we can prove that Algorithm 5 is differentially private.

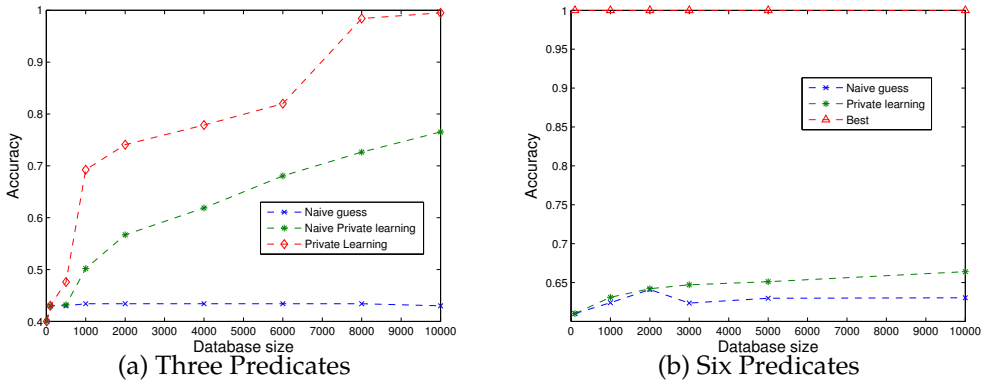
Theorem 6.9. *Algorithm 5 is ϵ-differentially private.*

Algorithm 5: Diff. Private ILP Algorithm

Input: positive evidence E^+; negative evidence E^-; privacy parameter ϵ; levels ℓ; rounds k
Output: the best theory
$T = \emptyset$;
for i = *1 to* k **do**
  H_b = Select a bottom clause in a differentially private way;
  h = Smart_Diff_Private_Reduction(E^+, E^-, ϵ/k, ℓ);
  Add h to T;
  Remove redundant examples using h;
end
return T

6.6 Experiments

In our experiments, we run our differentially private algorithm on synthetic data generated by the train generator described in (Muggleton,

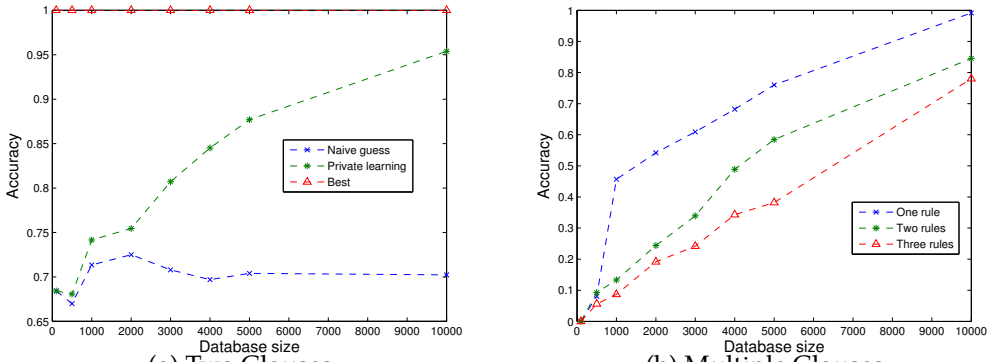


(a) Three Predicates (b) Six Predicates

Figure 6.2: One Clause with Different Number of Predicates

1995), in which the goal is to discriminate eastbound versus westbound trains based on the properties of their railcars. In all the experiments, we set the privacy parameter ϵ to be 1.0, and vary both the size of the data and the desired hypothesis to see how our algorithm performs. We measure the quality of our algorithm in terms of the accuracy of the output theory on a testing set. In all of our experiments, the naïve guess is the clause that assumes every train is eastbound.

In our first experiment, we only consider a hypothesis of one clause. We vary the number of predicates in the clause, and the results are shown in Figure 6.2. As we can see in Figure 6.2a, when there are three predicates in the desired clause, our private learning algorithm is able to learn the clause more accurately with more training data as discussed in Section 6.4. Furthermore, we also observe that our smart reduction algorithm shown in Algorithm 4 produces better results than the naïve reduction algorithm, demonstrating that reducing added noise by pruning low scoring clauses produces more accurate results. However, when increasing the number of predicates in the desired clause, the quality of our algorithm decreases. This is no surprise as the hypotheses space grows exponentially with the number of the predicates.



(a) Two Clauses (b) Multiple Clauses

Figure 6.3: Multiple Clauses with the Same Number of Predicates

We also investigate the effects on the number of clauses in a desired hypothesis, each of which consists of three predicates. As we can see, in Figure 6.3a, our private learning algorithm is able to produce high quality hypothesis with the growth in the size of the data, which is significantly better than the case of a single clause with six predicates. This is because the addition of a clause only increases the hypotheses space multiplicatively instead of exponentially. In both Figure 6.2 and Figure 6.3a we see that a large performance penalty is paid by the differentially private algorithms, as the non-private algorithm achieves perfect accuracy in all cases. Figure 6.3b shows the percentage of error reduction as more clauses need to be learned, showing the penalty due to the privacy budget being split among multiple clauses.

6.7 Conclusion

In this chapter, we have proposed a differentially private ILP algorithm. We have precisely quantified the trade-off between privacy and utility in ILP, and our results indicate that in order to satisfy a non-trivial utility requirement, an ILP algorithm incurs a huge risk of privacy breach. How-

ever, we find that when limiting the hypotheses space and increasing the size of the input data, our algorithm is able to output a hypothesis with high quality on synthetic data set. To the best of our knowledge, ours is the first one to attack this problem. With the availability of security-sensitive data such as electronic health records, we hope more and more people begin to pay attention to the privacy issues arising in the context of ILP.

There are many potential opportunities for future work. One such direction would be to formalize the notion of differential privacy with first-order logic, and discuss the tradeoff between privacy and utility in that context. Furthermore, we have only considered ILP with definite clauses, and it would be interesting to expand our work to statistical relational learning (De Raedt, 2005). Finally, since our algorithm requires one to limit the hypotheses space, it would also be interesting to investigate the feature selection problem in the context of differential privacy.

7 DIFFERENTIAL PRIVACY FOR CLASSIFIER EVALUATION

The previous two chapters have looked at the process of modifying machine learning algorithms so that they satisfy differential privacy. In the process, we discovered a gap in the existing literature on differentially private machine learning. Modifying machine learning algorithms puts the focus on training data, but information can also be revealed about testing data by publishing statistics. This chapter explores differentially private versions of classification metrics.

This chapter was published in (Boyd et al., 2015). An earlier version previously appeared in the thesis of Kendrick Boyd (Boyd, 2014).

7.1 Introduction

It has long been known that machine learning models can reveal information about the data used to train them. In the extreme case, a nearest neighbor model might store the dataset itself, but more subtle disclosures occur with all types of models. Even small changes in the training set can produce detectable changes in the model. This fact has motivated work to preserve the privacy of the training set by making it difficult for an adversary to discern information about the training data. One popular framework is differential privacy Dwork (2006), which sets bounds on the amount of change that can occur when any one training dataset row is modified.

Several authors have modified existing machine learning algorithms such that the algorithms satisfy differential privacy, e.g. Chaudhuri et al. (2011); Friedman and Schuster (2010); Rubinstein et al. (2009); Zhang et al. (2012) and Chapter 6. The machine learning method is a kind of query on the training database, and the modified algorithms output perturbed models. In doing so, the learned models can be released to the public,

and the privacy risk to the owners of the rows in the database is tightly bounded, even if the adversary has auxiliary information. However, these protections only cover the training dataset, not any latter uses of the model, such as evaluating a model's performance on another dataset. Evaluation metrics (e.g., accuracy, area under the ROC curve, average precision) on separate test datasets are a crucial part of the machine learning pipeline, and are just as important as training or interpreting the model. But when a privately trained model is evaluated on a dataset and the results are released, the privacy guarantees from training do not apply to the test dataset.

Consider a scenario in which multiple hospitals are collaborating to predict disease onset but are prevented by policy or law from sharing their data with one another. They may instead attempt to produce a model using data from one institution and test the model at other sites in order to evaluate how well the model generalizes. The institution generating the model might use a differentially private algorithm to create the model in order to protect their own patients, and then distribute the model to the other hospitals. These hospitals in turn run the model on their patients and produce an evaluation of the model performance such as accuracy or, more commonly, the area under the ROC curve (AUC). If the AUC is released, the test datasets at the latter institutions are not covered by any privacy protection that might have been used during training. The problem remains even if the training and test datasets exist at the same institution. While releasing an evaluation metric may seem to be a limited potential privacy breach, it has been demonstrated that data about patients can be reconstructed from ROC curves if the adversary has access to a subset of the test data Matthews and Harel (2013).

Our aim in this paper is to expand the scope of differential privacy in machine learning to include protection of test datasets in addition to existing work on protecting training datasets. While previous work

has examined related topics such as differentially private density estimation Wasserman and Zhou (2010); Hall et al. (2013) and parameter selection Chaudhuri and Vinterbo (2013), there has been little focus on the metrics most commonly used in machine learning evaluation: accuracy, area under the ROC curve (AUC) and average precision (AP). Without including these in the study of differential privacy, we are omitting from privacy considerations a large portion of what machine learning practitioners actually do with real data. Fundamental contributions of this paper include derivations of the local sensitivity of AUC and AP and emphasizing that accuracy, specificity, etc. can be readily made private via existing methods.

We start in Section 7.2 by providing background on evaluation metrics based on confusion matrices, such as ROC curves, as well as on differential privacy. In Section 7.3 we provide mechanisms for differentially private calculation of AUC and average precision. Finally we evaluate our mechanisms in several experiments in Section 7.4 and describe extensions to reporting symmetric binormal ROC curves in Section 7.5.

7.2 Background

7.2.1 Confusion Matrices and Rank Metrics

Evaluation for binary classification tasks begins with the confusion matrix. A confusion matrix describes the performance of a model that outputs *positive* or *negative* for every example in some test set. We characterize the test set by the number of positive examples n and the number of negative examples m. The four values in a confusion matrix are the true positives (tp), false positives (fp), true negatives (tn), and false negatives (fn). From the confusion matrix, many common evaluation metrics can be calculated. For example, accuracy is $\frac{tp+tn}{n+m}$ and precision is $\frac{tp}{tp+fp}$.

Many models output a probability or score, which creates a ranking over examples. This creates many different confusion matrices, depending on which value is chosen as a threshold. To describe the performance of these scoring classifiers, a common technique is to consider all possible thresholds and plot the trade-offs among confusion matrix metrics. For example, we have ROC curves when true positive rate ($\frac{tp}{n}$) is on the y-axis and false positive rate ($\frac{fp}{m}$) is on the x-axis. As a single summary metric, the area under the ROC curve is often used.

Area under ROC curve (AUC) is well-studied in statistics and is equivalent to the Mann-Whitney U statistic Pepe (2004).

$$\text{AUC} = \frac{1}{nm} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{I}[x_i < y_j] \tag{7.1}$$

where x_i for $1 \leqslant i \leqslant m$ are the scores on the negative examples in the test set, y_j for $1 \leqslant j \leqslant n$ are the scores on the positive examples, and $\mathbb{I}$ is the indicator function. Note that neither x_i nor y_j need be ordered.

Another metric we examine is average precision (AP), which is commonly used in information retrieval and is equivalent to the area under the precision-recall curve as $n + m \to \infty$ with constant $\frac{n}{m}$ Manning et al. (2008). AP is the average of the precision values across all values of recall. We use the following formulation:

$$\text{AP} = \frac{1}{n} \sum_{j=1}^{n} \frac{j}{j + \sum_{i=1}^{m} \mathbb{I}[x_i > y_j]} \tag{7.2}$$

with the same x_i and y_j as above and the additional assumption that the y_j's (but not x_i's) are sorted.

We assume no ties in the scores assigned to positive and negative examples when calculating (7.1) and (7.2). In case of ties, we impose a complete ordering where all negative examples are placed before the

positive examples to avoid overestimation of the curves and areas.

7.2.2 Differential Privacy

Differential privacy (see Section 2.4) is a privacy mechanism that guarantees the presence or absence of an individual's information in the database has little effect on the output of an algorithm. Thus, an adversary can learn limited information about any individual. More precisely, for any databases $D, D' \in \mathbb{D}$, let $d(D, D')$ be the number of rows that differ between the two databases. Differential privacy requires that the probability an algorithm outputs the same result on any pair of neighboring databases ($d(D, D') = 1$) is bounded by a constant ratio.

To ensure differential privacy, we must compute the *sensitivity* of the function we want to privatize. Here, sensitivity is the largest difference between the output on any pair of neighboring databases (not the performance metric $\frac{tp}{n}$).

Definition 7.1. *(Global sensitivity Dwork et al. (2006)): Given a function* $f : \mathbb{D} \to \mathbb{R}$*, the sensitivity of* f *is:*

$$GS_f = \max_{d(D,D')=1} |f(D) - f(D')| \tag{7.3}$$

This is identical to Definition 2.3 in Section 2.4. Here we call it global sensitivity because it is a worst-case bound for the change in f for an arbitrary pair of databases and to differentiate it from local sensitivity, introduced in Definition 7.2.

The approaches for obtaining differential privacy used previously in this thesis use the worst-case, global sensitivity to scale the noise. For some functions, such as median, the global sensitivity may be large, but the difference between outputs for most neighboring databases is quite

small. This motivates the work of Nissim et al. Nissim et al. (2007) to explore uses of local sensitivity.

Definition 7.2. *(Local sensitivity Nissim et al. (2007)): Given a function* $f : \mathbb{D} \to \mathbb{R}$*, the local sensitivity of* f *at* D *is*

$$LS_f(D) = \max_{d(D,D')=1} |f(D) - f(D')|. \tag{7.4}$$

Local sensitivity differs from global sensitivity because local sensitivity is parameterized by a particular database D, while global sensitivity is over all databases, i.e., $GS_f = \max_D LS_f(D)$.

Local sensitivity cannot be directly used to provide differential privacy, but a smooth upper bound can be used.

Definition 7.3. *(β-smooth sensitivity Nissim et al. (2007)): For* $\beta > 0$*, the* β*-smooth sensitivity of* f *is*

$$S^*_{f,\beta}(D) = \max_{D' \in \mathbb{D}} LS_f(D') e^{-\beta d(D,D')}. \tag{7.5}$$

Using the β-smooth sensitivity and Cauchy-like or Laplace noise provides differentially privacy as specified in the following theorem.

Theorem 7.4. *(Calibrating Noise to Smooth Bounds on Sensitivity, 1-Dimensional Case Nissim et al. (2007)): Let* $f : \mathbb{D} \to \mathbb{R}$ *be any real-valued function and let* $S : \mathbb{D} \to \mathbb{R}$ *be the* β*-smooth sensitivity of* f*, then*

1. *If* $\beta \leqslant \frac{\epsilon}{2(\gamma+1)}$ *and* $\gamma > 1$*, the algorithm* $f'(D) = f(D) + \frac{2(\gamma+1)S(D)}{\epsilon}\eta$*, where* η *is sampled from the distribution with density* $h(z) \propto \frac{1}{1+|z|^\gamma}$*, is* ϵ*-differentially private. Note that when* $\gamma = 2$*,* η *is drawn from a standard Cauchy distribution.*

2. *If* $\beta \leqslant \frac{\epsilon}{2\ln(\frac{2}{\delta})}$ *and* $\delta \in (0,1)$, *the algorithm* $f'(D) = f(D) + \frac{2S(D)}{\epsilon}\eta$, *where* $\eta \sim \texttt{Laplace}(1)$, *is* (ϵ, δ)*-differentially private.*

7.3 Private Mechanisms

Our discussion of differentially private evaluation will assume that a classification model is applied to a private database. The model could be hand-constructed by the submitter, trained on another private database in a differentially private way, or trained on a public database. Our goal is to ensure the evaluation output does not release too much information about any particular example in the private database by requiring a differentially private evaluation function.

We assume that the size of the database, $n + m$, is public information, but that the specific values of n and m are not publicly available. Though allowing n and m to be public would make our analysis for AUC and AP simpler and might achieve induced neighbors privacy Kifer and Machanavajjhala (2014), we believe that keeping the number of positive and negative examples private is a critical aspect of private model evaluation. If n and m were public information, the worst-case adversary for differential privacy that knows all but one row of the database would be able to trivially infer whether the last row is a positive or negative. Since the class label is often the most sensitive piece of information in a prediction task, releasing precise counts of positives and negatives would greatly weaken the security provided by a privacy framework. To illustrate the difference in neighboring databases, ROC curves for two neighboring databases are shown in Figure 7.1.

What types of evaluation metrics can be released privately under this framework? Any metric based on a single confusion matrix can be made private by applying the standard methods, such as Laplace noise, for differentially private counts or marginals Dwork (2006). Thus, differentially

private accuracy, recall, specificity, precision, etc. can be obtained. We focus on more complex metrics such as AUC that are both more useful for classifier evaluation Provost et al. (1998) as well as more challenging to implement privately.

7.3.1 Reidentifying AUC

Prior work has demonstrated the vulnerability of data points in ROC curves to reidentification Matthews and Harel (2013); we extend that to AUC to demonstrate that the problem remains even with the summary metric. Consider the problem of identifying the class of an excluded example given the AUC of the full dataset. Here the adversary has access to all of the data points but one, and also knows the AUC of the full data. The goal is to predict whether the final example is a member of the positive or negative class. Note that we do not assume the adversary knows where the target example should go in the ranking.

The adversary's algorithm is to attempt to place the target example at each position in the ranking, and calculate the resulting AUC under the assumption that the example is positive and again assuming it is negative. The class that produces an answer closest to the released AUC for the full dataset (or the most frequent class in the case of ties) is guessed as the class of the example. This setup is similar to the assumptions of differential privacy in terms of looking at the influence on AUC from a single example. However, it is not a worst case analysis and is concerned with identifying an attribute value of the target example not simply its presence in the original data.

Figure 7.2 shows the ability of the attacker to guess the class of the target example given a sample of data from the UCI adult dataset. One heuristic method that could be used to interfere with this attack is to round the released AUC to a smaller number of decimal places, and is illustrated in the figure. When the AUC is given to a high number of decimal places, the

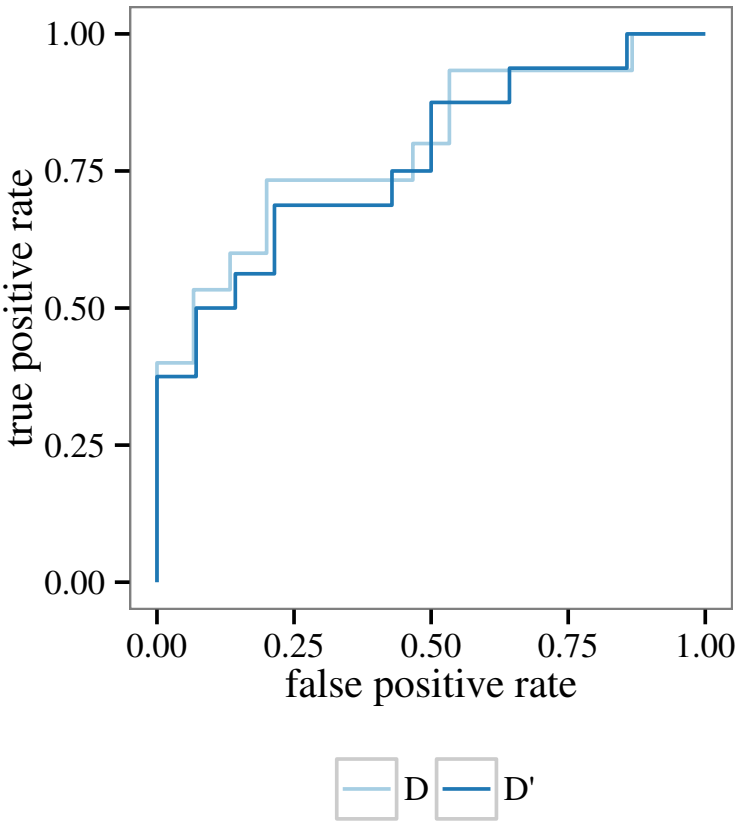


Figure 7.1: ROC curves for two neighboring databases where the difference between D and D′ is that a negative was changed to a positive and given a new score. D contains 15 positives and 15 negatives and D′ contains 16 positives and 14 negatives. AUC for D and D′ is 0.796 and 0.772, respectively.

adversary is able to recover the class value with high probability, though this ability falls as the number of data points increases. Rounding the AUC value to fewer decimal places does reduce the adversary's success, but comes at a cost to precision.

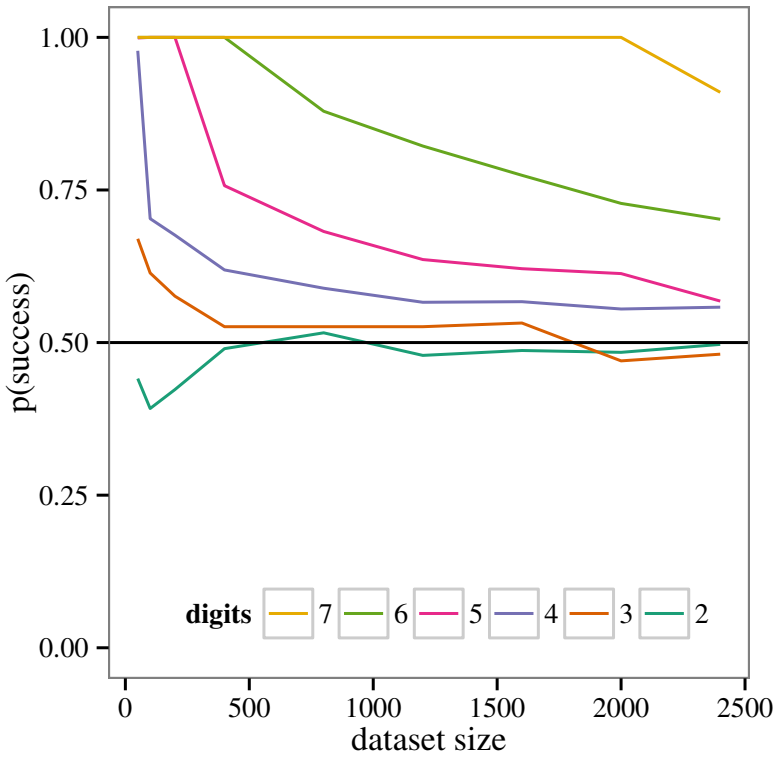


Figure 7.2: Adversary's success rate in identifying the class of the missing example given AUC of a dataset containing half positives and half negatives with specified significant digits. The horizontal black line at 0.5 denotes performance of randomly guessing the class.

7.3.2 AUC

When calculating sensitivity of AUC, each example can contribute to the sum multiple times. The sensitivity of AUC is further complicated because the factor $\frac{1}{nm}$ differs between neighboring datasets when a positive example changes to a negative or vice versa. Fortunately, we can bound the maximum change in AUC between neighboring datasets to find the local sensitivity.

Theorem 7.5. *Local sensitivity of area under the ROC curve (AUC) is*

$$\mathrm{LS}_{AUC}(n, m) = \begin{cases} \frac{1}{\min(n,m)} & \text{if } n > 0 \text{ and } m > 0 \\ 1 & \text{otherwise} \end{cases} \tag{7.6}$$

where n *and* m *are the number of positive and negative examples in the test set, respectively.*

The full proof can be found in Section 7.7, but we present a short proof outline here. For the purposes of calculating AUC, it is sufficient to consider the ranking and class label for rows in the database. This is captured in the x_i and y_j variables and indicator function in (7.1). Since most rows do not change, the majority of the indicator functions in the double summation in (7.1) remain the same. So we can decompose (7.1) into components that change and those that do not for each of the four possible cases. We then bound the maximum difference possible between two databases in the components that do change to find the local sensitivity.

Local sensitivity itself is not suitable for creating differentially private algorithms since adding different amounts of noise for adjacent databases can leak information Nissim et al. (2007). Instead, we use β-smooth sensitivity which ensures the scale of noise for adjacent databases is within a factor of e^{β}.

Theorem 7.6. β*-smooth sensitivity of area under the ROC curve (AUC) is*

$$S^*_{\mathrm{AUC},\beta} = \max_{0 \leqslant i \leqslant n+m} \mathrm{LS}_{AUC}(i, n + m - i) e^{-\beta |i-n|} \tag{7.7}$$

The proof, given in Section 7.7, is a straight-forward application of the definition of β-smooth sensitivity. Figure 7.3 shows smooth sensitivity given by (7.7) for several values of β demonstrating that the advantages of lower sensitivity are more pronounced with higher β or balanced datasets.

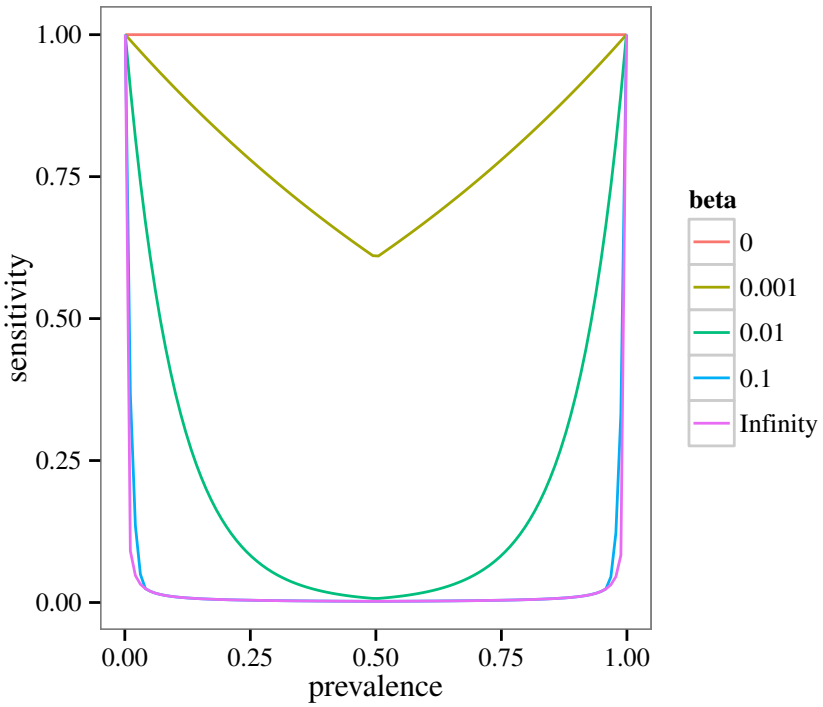


Figure 7.3: β-smooth sensitivity for AUC on a dataset with 1000 examples. The sensitivity changes depending on the prevalence of positive examples, shown on the x-axis. When β is 0, the smooth sensitivity cannot change so it is always 1, i.e., the global sensitivity. If β is infinity, there is no smoothness constraint and the smooth sensitivity is exactly the local sensitivity.

With the β-smooth sensitivity of AUC, appropriately scaled Cauchy noise can be used to obtain ϵ-differential privacy or Laplace noise can be used to obtain (ϵ, δ)-differential privacy as described in Theorem 7.4. Since the range of AUC is $[0, 1]$, we truncate the output. The truncation does not violate differential privacy Ghosh et al. (2009) because an adversary knows the range of the true function.

7.3.3 Average Precision

Once again we want to use smooth sensitivity to produce differentially private average precision, but first we need to find the local sensitivity of average precision (AP). Precision at low recall has high variance since changing just a single row for neighboring datasets can cause precision to go from 1 to $\frac{1}{2}$ simply by adding a high-scoring negative example. Though precision at low recalls can vary substantially between neighboring datasets, the impact on average precision is mitigated by the $\frac{1}{n}$ coefficient in (7.2) and the sensitivity is bounded in the following theorem.

Theorem 7.7. *Local sensitivity of average precision (AP) is*

$$\Delta_{AP} = \begin{cases} \max\left(\frac{\log(n+1)}{n}, \frac{9+\log(n-1)}{4(n-1)}\right) & \\ + \max\left(\frac{\log(n+1)}{n}, \frac{9+\log n}{4n}\right) & \text{if } n > 1 \\ 1 & \text{if } n \leqslant 1 \end{cases} \tag{7.8}$$

where n *is the number of positive examples and* $\log$ *is the natural logarithm.*

The proof approach for AP is similar to that for AUC in that the summation in (7.2) can also be decomposed into indicator functions that do and do not change. However, there are a few differences to the approach that are outlined here and the full proof can be found in Section 7.7. Most notably, instead of directly considering changing a row of the database, we derive bounds for adding or removing a positive or negative example

separately as this greatly simplified the math. Changing an example is equivalent to removing an example and then adding an example, so the local sensitivity is the sum of the bounds for both actions. Additionally, due to the fraction that is summed in (7.2), the bounds include harmonic terms. We use a simple upper bound of the harmonic numbers to obtain (7.8), but a tighter bound could be used to get slightly smaller local sensitivities.

Note that the local sensitivity of AP depends only on the number of positive examples, n, and not the number of negative examples. This aligns with the notion that AP (and PR curves) does not give credit for true negatives.

Theorem 7.8. *β-smooth sensitivity of average precision (AP) is*

$$S^*_{AP,\beta} = \max_{0 \leqslant i \leqslant n+m} LS_{AP}(i) e^{-\beta|i-n|} \tag{7.9}$$

The proof is identical to Theorem 7.6 with LS_{AP} instead of LS_{AUC}.

As in AUC, we can use Cauchy or Laplace noise to produce ϵ- or (ϵ, δ)-differentially private outputs. The range of AP is not [0,1] since the the minimum AP for any particular n and m is strictly greater than zero Boyd et al. (2012). Though the minimum AP can be sizable (about 0.3 when $n = m$), it depends on the non-public n and m, so we cannot truncate to the database specific minimum AP and just truncate to the overall range of [0,1].

7.4 Experiments

In this section we apply the algorithms from the previous section to two datasets. Since our mechanisms operate on the output of a classification model, they should not be influenced by the number of features in the original dataset. The first dataset is the `adult` dataset from the UCI repository Bache and Lichman (2013). It contains potentially private information

in both the class label (yearly income greater or less than $50,000) and other features (e.g., capital gain/loss and work status) that individuals might be hesitant to provide without privacy guarantees. The dataset has 14 features and 48,842 examples. The second dataset is `diabetes` – a medical dataset from a Kaggle competition[1] to predict diabetes from anonymized electronic health records. We processed the dataset to include age, gender, and binary features for the 50 most common drugs, diagnoses, and laboratory tests for a total of 152 features. The dataset contains 9,948 patients.

In our experiments, we trained a model on part of each dataset using logistic regression. We use differentially private evaluation on subsets of the rest of the dataset as a surrogate for a private database. We investigate the utility of the differentially private evaluations using mean absolute error (MAE). The primary error of interest is between the differentially private output and the answer calculated directly from the private data which we call the "DP error". For example, Figure 7.4 shows the distribution of private outputs for repeated queries of the same private dataset and model.

However, the DP error is not the only source of error or uncertainty in the process. The private data set is a sample from a (much) larger population of examples, and what we as machine learners are most interested in is the performance on the entire population. But we just have a small data set, so the metric we calculate from that private data set (before applying the perturbation for privacy) is itself just an estimate, with associated error, of the population value. We refer to the difference between the population value and the non-private metric calculated from the sampled data set as the "estimation error". The estimation error is a level of uncertainty that is unavoidable, given the size of the data set. If the additional noise due to making the output differentially private is much smaller then making the

[1] http://www.kaggle.com/c/pf2012-diabetes

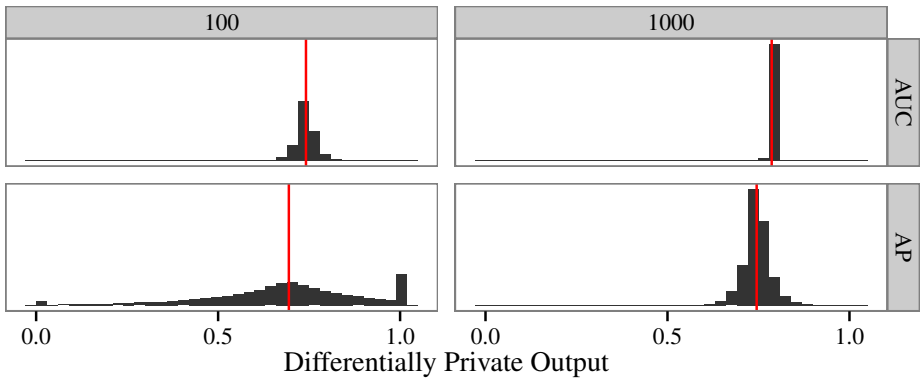


Figure 7.4: Histograms of (ϵ, δ)-differentially private AUC (top) and AP (bottom) from datasets of size $n = m = 100$ (left) and $n = m = 1000$ (right). The vertical red line denotes the non-private value and the histograms are from repeated draws of Laplace noise and correspond to the DP error. Data are from the `diabetes` dataset and the privacy budget is $\epsilon = 1$ and $\delta = 0.01$.

output private has minimal impact on utility. Even if the DP error is of similar magnitude to the estimation error, the private metric is still highly useful, adding a bit more noise to the result, but not changing the scale of uncertainty.

The top part of Figure 7.5 shows the DP and estimation error of AUC for several dataset sizes. While the Cauchy noise when $\delta = 0$ causes the estimation error to be considerably larger than the DP error, with (ϵ, δ)-differential privacy the DP error is similar or smaller than the estimation error. As dataset size increases, not only does the DP error decrease, but it decreases relative to the estimation error. Thus, the lost utility from providing privacy is decreasing more quickly than the uncertainty due to estimating from a fixed data set. For the larger dataset on the right side of Figure 7.4, the private AUC is clustered tightly around the non-private version, while the smaller dataset is more spread out due to the larger amount of noise required by differential privacy.

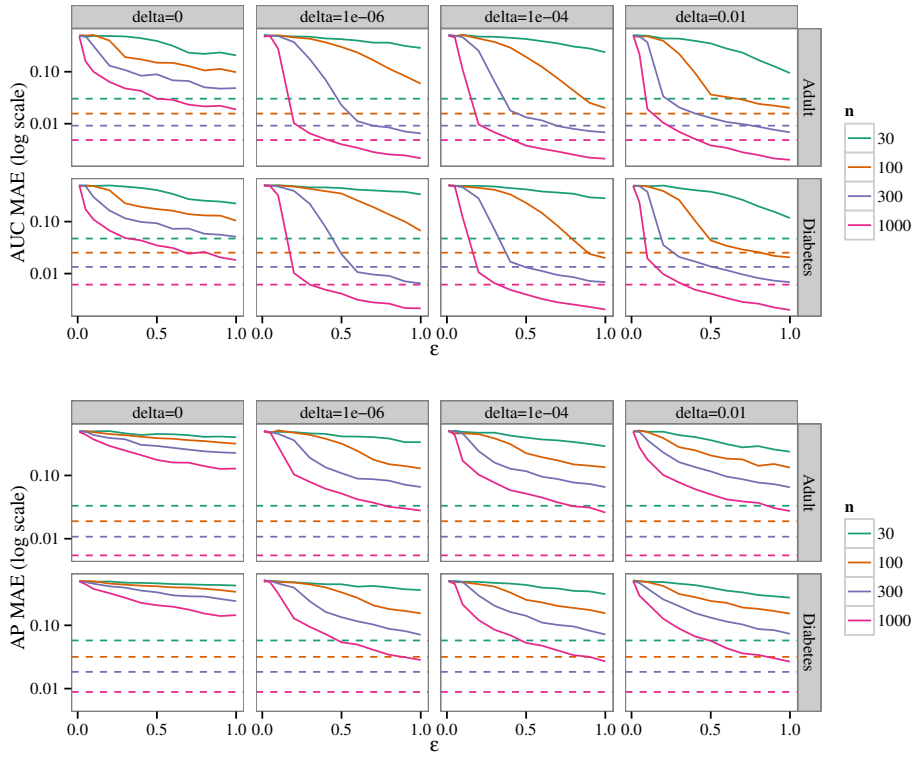


Figure 7.5: Mean average error (log scale) of AUC (top) and AP (bottom) for several dataset sizes ($n = m$). The solid lines show the DP error due to the perturbation required for (ϵ, δ)-differential privacy, while the horizontal dashed lines show the estimation error. The $\delta = 0$ plots use Cauchy noise to guarantee ϵ-differential privacy and the other plots use Laplace noise.

For AP, we use the same setup as for AUC, measuring MAE versus ϵ for several dataset sizes with $n = m$. The general trends for DP error of AP in the bottom part of Figure 7.5 are similar to those for AUC, but the DP error is much higher than the estimation error. This matches the sensitivity theorems where there is an additional $\log n$ factor in the AP sensitivity compared to AUC sensitivity. Thus, for DP error and estimation error to be equal for AP requires much larger datasets or ϵ than for AUC. The bottom part of Figure 7.4 shows the distribution of outputted private

AP values. When n is small, the Laplace noise often puts the value outside the valid range of AP. Since we truncate back to $[0, 1]$, the mode of the AP distribution for $n = 100$ is 1 instead of the correct AP of about 0.75.

7.5 Symmetric Binormal Curves

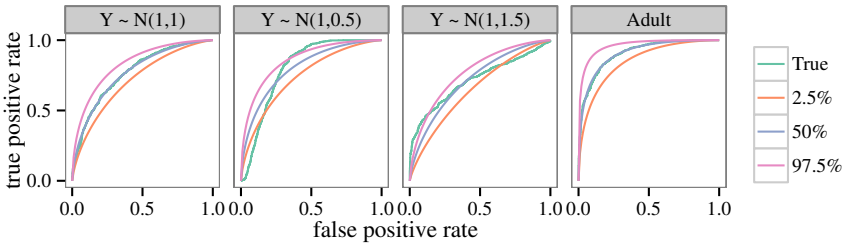


Figure 7.6: (ϵ, δ)-differentially private ROC curves generated using symmetric binormal method. The true ROC curve with $n = m = 1000$ and empirical 2.5%, 50% (median) and 97.5% quantiles of the differentially private curves are shown. $\epsilon = 0.1$ and $\delta = 0.01$.

Extending the work on AUC to ROC curves is challenging. One possibility is to add noise directly to each of the points that make up the curve. However, this method requires the privacy budget to be split among all points in the curve. The approach was used in Stoddard et al. (2014) to generate differentially private ROC curves, first selecting a number of interpolation points followed by adding noise to each of the points. With a differentially private AUC from Section 7.3.2 and a method to map from AUC back to an ROC curve, we can produce differentially private ROC curves. Unfortunately, numerous ROC curves have the same area, so we need a way to choose among the ROC curves with a specified area. One way of specifying unique curves for each value of AUC is to use a symmetric binormal ROC curve Pepe (2004), a 1-parameter ROC curve estimator.

Binormal ROC curves assume the scores for negative and positive examples are drawn from two normal distributions. The theoretical ROC curve is $y = \Phi(a + b\Phi^{-1}(x))$, where Φ is the cumulative distribution functions of the normal distribution, $a = \frac{\mu_Y - \mu_X}{\sigma_Y}$, and $b = \frac{\sigma_X}{\sigma_Y}$. To remove the second degree of freedom, we can set $b = 1$. The output curves will be symmetric binormal curves (symmetric over the line $y = 1 - x$), which assumes the standard deviation of the positive and negative examples is the same. Setting $a = \sqrt{2}\Phi^{-1}(\text{AUC})$ produces the desired curve.

Figure 7.6 shows differentially private ROC curves created in this manner as well as the true curves being approximated. The far right subplot is for the `adult` dataset and the others are for data simulated from various binormal distributions. The simulations all use $X \sim N(0, 1)$ and correspond to the assumed symmetric form in the far left subplot when $Y \sim N(1, 1)$ and two different non-symmetric misspecifications in the other two plots. If a classifier's predictions (or any monotonic transformation) are modeled reasonably by two normal distributions with the same standard deviation, the symmetric ROC curves are a good approximation, as in the `adult` dataset. Compared to Stoddard et al. (2014), this method produces smoother curves when the assumptions are met, but is not flexible to differing variances among the two classes.

7.6 Conclusion

Differentially private models allow organizations with sensitive data to provide guarantees about the effect of model release on the privacy of database entries. But for these models to be effectively evaluated, they must be run on new data, which may have similar privacy concerns. We presented methods for providing the same differential privacy guarantees for model evaluation, irrespective of the training setting. We provided high-utility mechanisms for AUC and AP. Future work includes creat-

ing mechanisms for other evaluation methods, general ROC curves, and investigating the effect of cross-validation. We hope the discussion of differential privacy for model evaluation motivates future work to enable differential privacy to be applied more broadly throughout machine learning.

7.7 Proofs

7.7.1 Proof of Theorem 7.5

Proof. Let D and D' be two neighboring databases that differ by exactly one row. Let n and m be the number of positive and negative examples in D, respectively.

We consider the four cases of moving a negative in the ranking, moving a positive, changing a positive to negative (and moving), and changing a negative to a positive. Our analysis of these four cases requires $n > 0$ and $m > 0$, so for completeness we say the local sensitivity of AUC is 1 if either $n = 0$ or $m = 0$.

Case 1) Move negative: D' has the same x_i and y_j as D except for some x_k that is now x^* in D'. The only changes in (7.1) occur when x_k is compared in the indicator functions. x_k appears n times and each time the indicator function can change by at most 1, so in this case sensitivity is $\frac{n}{nm} = \frac{1}{m}$.

Case 2) Move positive: Similar to Case 1, D' is the same as D except for some y_k that changes to y^*. This y_k appears in (7.1) m times so the sensitivity is $\frac{m}{nm} = \frac{1}{n}$.

Case 3) Change negative to positive: Here, D' has $n+1$ positive and $m-1$ negative examples (assume $m > 2$) with the same x_i and y_j except for some x_k that has been removed and a new positive example with score y^* has been added. Without loss of generality, assume that $k = m$. Using C to collect the unchanged terms, and noting that $0 \leqslant C \leqslant (m-1)n$, we

have

$$\mathrm{AUC}(D) = \frac{1}{nm}(C + \sum_{j=1}^{n} \mathbb{I}[x_m < y_j]) \tag{7.10}$$

$$\mathrm{AUC}(D') = \frac{1}{(n+1)(m-1)}(C + \sum_{i=1}^{m-1} \mathbb{I}[x_i < y^*]). \tag{7.11}$$

$$\begin{aligned}\mathrm{AUC}(D) - \mathrm{AUC}(D') = &\frac{m-n-1}{nm(n+1)(m-1)}C \\ &+ \frac{1}{nm}\sum_{j=1}^{n} \mathbb{I}[x_m < y_j] \\ &- \frac{1}{(n+1)(m-1)}\sum_{i=1}^{m-1} \mathbb{I}[x_i < y^*]\end{aligned} \tag{7.12}$$

(7.12) is maximized when each of the three terms is maximized. The first term is maximized when $m > n$ and $C = (m-1)n$,

$$\frac{m-n-1}{nm(n+1)(m-1)}C \leqslant \frac{m-n-1}{m(n+1)}. \tag{7.13}$$

The second and third terms are bounded above by $\frac{n}{nm} = \frac{1}{m}$ and 0, respectively. Putting it all together we have an upper bound of

$$\mathrm{AUC}(D) - \mathrm{AUC}(D') \leqslant \frac{m-n-1}{m(n+1)} + \frac{n}{nm} \leqslant \frac{m}{nm} = \frac{1}{n}. \tag{7.14}$$

Similarly, the lower bound for (7.12) occurs when $n > m$ and is

$$\mathrm{AUC}(D) - \mathrm{AUC}(D') \geqslant \frac{m-n-1}{m(n+1)} - \frac{1}{n+1} = -\frac{1}{m}. \tag{7.15}$$

Case 4) Change positive to negative: Symmetric with Case 3. □ □

7.7.2 Proof of Theorem 7.6

Proof. Let n_x and n_y be the number of positive examples in databases x and y, respectively, and similarly m_x and m_y be the number of negatives. The smallest row difference between x and y occurs if we just need to change the positive or negative labels on the minimal number of examples to ensure the n_i and m_i counts are correct, hence $d(x, y) \geqslant |n_x - n_y|$. Starting from Definition 2.2 of Nissim et al. Nissim et al. (2007), we have,

$$S^*_{AUC,\beta} = \max_{y \in D^{n+m}} LS_{AUC}(n_y, m_y) e^{-\beta |n_x - n_y|} \tag{7.16}$$

$$= \max_{0 \leqslant i \leqslant n+m} LS_{AUC}(i, n+m-i) e^{-\beta |n_x - i|} \tag{7.17}$$

since there always exists some y for which $d(x, y) = |n_x - n_y|$. □ □

7.7.3 Proof of Theorem 7.7

Proof. Let $x_1, x_2, ..., x_m$ and $y_1, y_2, ..., y_n$ be the classifier scores on the m negative and n positive examples for a data set D. To bound the maximum change in AP between D and a neighboring database, we consider the four cases of adding or removing a positive example and adding or removing a negative example.

Case 1) Remove positive: Assume WLOG that $y_1 > y_2 > ... > y_n$. Consider making D' by removing a positive example y_z. Separating out the different parts of the AP sum to facilitate comparison between D and D', we have

$$AP(D) = \frac{1}{n} \left[\sum_{i=1}^{z-1} \frac{i}{i + s_i} + \frac{z}{z + s_z} + \sum_{i=z+1}^{n} \frac{i}{i + s_i} \right] \tag{7.18}$$

where $s_i = \sum_{j=1}^{m} \mathbb{I}[x_j > y_i]$. Removing the y_z example for D' yields

$$\mathrm{AP}(D') = \frac{1}{n-1}\left[\sum_{i=1}^{z-1} \frac{i}{i+s_i} + \sum_{i=z+1}^{n} \frac{i-1}{i-1+s_i}\right]. \tag{7.19}$$

After renormalizing and simplifying,

$$\begin{aligned}\mathrm{AP}(D) - \mathrm{AP}(D') = \frac{1}{n(n-1)}\Bigg[&\sum_{i=1}^{z-1} \frac{-i}{i+s_i} \\ &+ \frac{(n-1)z}{z+s_z} + \sum_{i=z+1}^{n} \frac{-i}{i+s_i} \\ &+ \sum_{i=z+1}^{n} \frac{ns_i}{(i+s_i)(i-1+s_i)}\Bigg]\end{aligned} \tag{7.20}$$

The two sums of $\frac{-i}{i+s_i}$ in (8.4) include all i's except $i = z$. So we can add and subtract $\frac{z}{z+s_z}$ to get,

$$\begin{aligned}\mathrm{AP}(D) - \mathrm{AP}(D') = \frac{1}{n(n-1)}\Bigg[&\frac{nz}{z+s_z} + \sum_{i=1}^{n} \frac{-i}{i+s_i} \\ &+ \sum_{i=z+1}^{n} \frac{ns_i}{(i+s_i)(i-1+s_i)}\Bigg].\end{aligned} \tag{7.21}$$

The absolute value of first two terms are maximized when $s_z = 0$ and $s_i = 0$, respectively. Both are bounded by $\frac{1}{n-1}$. The third term is relaxed to $\frac{1}{n-1}\sum_{i=z+1}^{n} \frac{s_i}{(i-1+s_i)^2}$. We need to maximize $\frac{s_i}{(i-1+s_i)^2}$ for each i where s_i is free to take any (integer) value between 0 and m. This function is maximized when $s_i = i - 1$, which is always a valid choice for s_i, and gives an upper bound of

$$\frac{1}{n-1}\sum_{i=z+1}^{n} \frac{i-1}{(i-1+i-1)^2} = \frac{1}{4(n-1)}\sum_{i=z+1}^{n} \frac{1}{i-1}. \tag{7.22}$$

Since all terms of the sum in (7.22) are positive ($z \geqslant 1$, so $i \geqslant 2$), it is maximized when there are as many terms as possible, i.e., when $z = 1$:

$$\frac{1}{4(n-1)} \sum_{i=z+1}^{n} \frac{1}{i-1} \leqslant \frac{1}{4(n-1)} \sum_{j=1}^{n-1} \frac{1}{j} = \frac{H_{n-1}}{4(n-1)}. \tag{7.23}$$

where H_{n-1} is the $(n-1)$st harmonic number. Combining the three terms to bound (7.21), we have

$$\Delta = \frac{2}{n-1} + \frac{H_{n-1}}{4(n-1)} = \frac{8 + H_{n-1}}{4(n-1)} \tag{7.24}$$

Case 2) Add positive: Equivalent to Case 1, but if D has n positive examples then D′ has $n+1$, so the sensitivity is

$$\Delta = \frac{8 + H_n}{4n}. \tag{7.25}$$

Case 3) Remove negative: Consider removing a negative example x_k.

$$\mathrm{AP}(D) = \frac{1}{n} \sum_{i=1}^{n} \frac{i}{i + s_i} \tag{7.26}$$

$$\mathrm{AP}(D') = \frac{1}{n} \sum_{i=1}^{n} \frac{i}{i + s_i + \delta_i} \tag{7.27}$$

where $s_i = \sum_{j=1}^{m} \mathbb{I}[x_j > y_i]$ and $\delta_i = -\mathbb{I}[x_k > y_i]$ is the change in false positive counts between D and D′. The difference in AP is

$$\mathrm{AP}(D) - \mathrm{AP}(D') = \frac{1}{n} \sum_{i=1}^{n} \frac{i\delta_i}{(i + s_i)(i + s_i + \delta_i)}. \tag{7.28}$$

$\delta_i \in \{0, -1\}$, so the absolute value of (7.28) is maximized when $\delta_i = -1$ and $s_i = 1 \; \forall \; i$ ($s_i \neq 0$ because there must be an existing false positive to

remove).

$$|\mathrm{AP}(D) - \mathrm{AP}(D')| \leqslant \frac{1}{n}\sum_{i=1}^{n}\frac{i}{(i+1)i} \tag{7.29}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{i+1} = \frac{H_{n+1}-1}{n} \tag{7.30}$$

Case 4) Add negative: If we add a negative example instead of removing it, we again get to (7.28), but now $\delta_i \in \{0,1\}$ and the absolute value is maximized when $\delta_i = 1$ and $s_i = 0\ \forall\ i$. Using these values also gives (7.30).

Putting the cases together,

$$\begin{aligned}\Delta_{\mathrm{AP}} = \max&\left(\frac{H_{n+1}-1}{n}, \frac{8+H_{n-1}}{4(n-1)}\right)\\ +\max&\left(\frac{H_{n+1}-1}{n}, \frac{8+H_n}{4n}\right)\end{aligned} \tag{7.31}$$

is our tightest bound on the local sensitivity. Using the bound $H_n < 1 + \log n$ gives (7.8). □ □

8 SUBSAMPLED EXPONENTIAL MECHANISM: DIFFERENTIAL PRIVACY IN LARGE OUTPUT SPACES

In this chapter we address another limitation of existing methods for preserving differential privacy: the exponential mechanism (Theorem 2.6) requires the evaluation of every possible outcome. In this chapter we propose an alternative based on sampling, and evaluate against the k-median problem, which is problematic due to the combinatorial solution set. This chapter was published as (Lantz et al., 2015a)

8.1 Introduction

Differential privacy (see Section 2.4) has been explored as a framework for data privacy in many scenarios, including data release (Xiao et al., 2011; Hardt et al., 2012; Blum et al., 2013), machine learning (Rubinstein et al., 2009; Friedman and Schuster, 2010; Zhang et al., 2013), auctions (McSherry and Talwar, 2007), and graph analysis (Karwa et al., 2011; Blocki et al., 2013; Kasiviswanathan et al., 2013a). For any two databases differing in one record, the ability of an adversary to determine the effect of the differing record is tightly bounded.

In this chapter, we examine an extension to the standard exponential mechanism for enforcing differential privacy. Roughly speaking, in the exponential mechanism the possible outcomes are scored by their utility, and instead of choosing the one with the highest score, a noisy choice is made in such a way that all solutions have non-zero probability of being selected. There are many cases in which the number of possible outcomes to a query are exponential, factorial, or even infinite. The exponential mechanism would typically require us to evaluate the quality of every possible outcome in order to calculate an appropriately-weighted distribution from which to select an outcome. This limits differential privacy to

cases where the outcome is numeric (falling under a special case called the Laplace mechanism) or is a member of a relatively small set. Nonetheless, it may be quite easy to sample an outcome from a very large distribution. We propose to sample from the space of possible outcomes, and choose an outcome from among this sample of the outcome space. We prove that the mechanism still preserves differential privacy, since the action is independent of the database. In addition, following a result from McSherry and Talwar (2007), we show that sampling the output space results in only a moderately weaker accuracy bound.

While the proposed mechanism is quite simple, it can prove surprisingly effective. We examine the private k-median problem, in which a set of cluster centers must be selected taking into account a private subset of data points. There are a factorial number of k-subsets, so approximate solutions are typically used even in a non-private setting. An existing differentially private solution has been proposed (Gupta et al., 2010), which uses a local search algorithm to find possible cluster centers. We show that our algorithm performs significantly better than theirs, and with a much faster runtime.

In addition, we explore an aspect of differential privacy that is often overlooked. The exponential mechanism is defined using a base measure over the outcome space. This is typically assumed to be uniform, but it need not be. It need only be independent of the data itself. We show how utilizing a non-uniform base distribution improves the effectiveness of our algorithm when applied to the private k-median task.

8.2 Subsampled Exponential Mechanism

The subsampled exponential mechanism is an extension to the exponential mechanism (Theorem 2.6 in which all possible outcomes do not need to be evaluated. Instead we require some way to sample outcomes from

the outcome space, and run the exponential mechanism among a limited sample. Note that the sampling is done over the output space, not the records in the database, as has been discussed in previous work (Chaudhuri and Mishra, 2006; Dwork et al., 2010; Wasserman and Zhou, 2010; Chaudhuri and Hsu, 2012; Li et al., 2012; Ji and Elkan, 2013). Since the exponential mechanism itself could be considered a weighted sampling over the output space, we call our mechanism the subsampled exponential mechanism.

Definition 8.1 (Subsampled Exponential Mechanism)**.** *Given a quality function* $q : D \times \mathcal{Z} \to \mathbb{R}$ *that assigns a score to each outcome* $z \in \mathcal{Z}$ *and a database-independent base distribution on outputs* $\mu(\mathcal{Z})$*, the subsampled exponential mechanism draws* m *independent samples* $z_1, z_2, ..., z_m \sim \mu(\mathcal{Z})$ *and uses the exponential mechanism (with equal weight on the* m *samples). That is, after selecting the* m *samples,*

$$\Pr(M(D) = z) = \frac{e^{\epsilon\, q(D,z)}}{\sum_{i=1}^{m} e^{\epsilon\, q(D,z_i)}} \tag{8.1}$$

The success of the subsampled exponential mechanism will be related to the distribution over outcomes of the quality function q as applied to a particular database. Figure 8.1 shows two theoretical distributions of q, where $q1$ has more probability mass shifted towards the maximum value of 1. The subsampled exponential mechanism will be more likely to sample an output with a high value of q if the distribution looks more like $q1$ than $q2$.

Before looking at the properties of the subsampled exponential mechanism, we will start by motivating an example of a problem for which it might be useful. When a patient gets cancer, a wide variety of mutations occur in the nuclear DNA of the tumor. Recent technologies have allowed scientists to perform genetic sequencing on tumors. If we collected tumor samples from patients, we could look at the difference between the

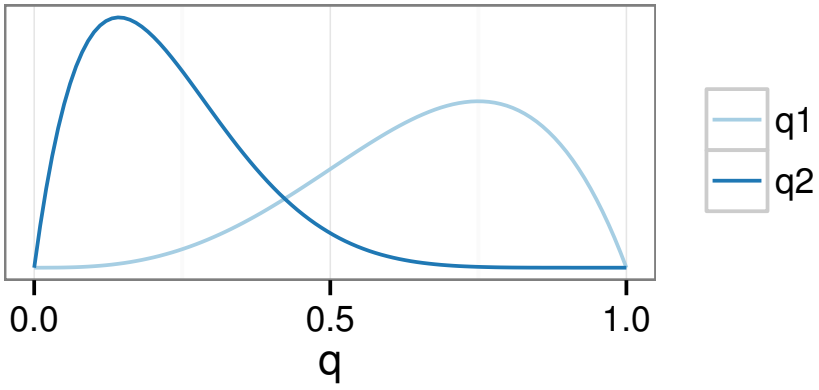


Figure 8.1: Two example distributions for the q function. Of these two distributions, the subsampled exponential mechanism will produce more useful (higher q value) outputs when applied to q1 than to q2 because of the long right tail on the q2 distribution.

sequence of a given gene (for example, the tumor suppressor p53) in our patients and a reference genome. We might want to publish a consensus sequence that approximates an average of the mutated sequences, but without releasing any of the patient's sequences. Can we do this in a differentially private manner?

If the sequences were all the same length, one approach could be to look individually at each position, and use the exponential mechanism and the proportion of each base pair at that position. This approach is straightforward, but has two important limitations. One, due the composition mechanism we must divide our ϵ by the number of positions in the sequence, increasing the resulting randomness at each base pair. Two, because of insertions and deletions, DNA sequences are rarely exactly the same length, and a variety of alignment algorithms and scoring functions are used.

Another approach would be to generate all possible sequences of length

n (or for a small range of n) and use the exponential mechanism to pick the best one. Our quality function could be the average pairwise distance to the patient sequences using any DNA sequence alignment function for which we could calculate a sensitivity. For the exponential mechanism, the number of possible sequences is $O(4^n)$, making it impractical for all but the smallest n.

However, we can leverage the fact that we have a reference sequence that is close to the patient sequences to produce randomized sequences that are close to the reference and choose among them with the subsampled exponential mechanism. For example, we could generate samples by performing a series of random mutations on the reference sequence. Note that we need not be able to explicitly calculate the probability distribution over sequences that this process generates, we only need be able to draw from it.

8.2.1 Proof of Differential Privacy

Theorem 8.2. *The subsampled exponential mechanism defined in definition 8.1 is $2\epsilon\Delta_q$-differentially private.*

Proof. Let $\hat{z} = z_1, ..., z_m$ be a vector-valued random variable for the m samples from base distribution μ. The sampled exponential mechanism is a two-step process. For a particular $z^* \in \mathcal{Z}$ to be selected by the mechanism, it must first be included in the sample, and then selected from that sample. This gives the following probability for returning a particular z^*:

$$\Pr(M(D) = z^*) = \mathbb{E}_{\hat{z}}\left[\frac{e^{\epsilon q(D,z^*)}}{\sum_{i=1}^{m} e^{\epsilon q(D,z_i)}}\mathbb{I}[z^* \in \hat{z}]\right] \tag{8.2}$$

where $\mathbb{I}[\text{expr}]$ is the indicator function that evaluates to 1 if expr is true and 0 otherwise. To reduce the mathematical clutter, we represent the normalization term in eq. (8.2) by $\phi(D, \hat{z}) = \frac{1}{\sum_{i=1}^{m} e^{\epsilon q(D,z_i)}}$.

$$\Pr(M(D) = z^*) = \mathbb{E}_{\hat{z}}\left[\phi(D, \hat{z})\, e^{\epsilon\, q(D, z^*)}\, \mathbb{I}[z^* \in \hat{z}]\right] \tag{8.3}$$

$$= e^{\epsilon\, q(D, z^*)}\, \mathbb{E}_{\hat{z}}\left[\phi(D, \hat{z})\, \mathbb{I}[z^* \in \hat{z}]\right] \tag{8.4}$$

$$= e^{\epsilon\, q(D, z^*)} \Pr(z^* \in \hat{z})\, \mathbb{E}_{\hat{z}}\left[\phi(D, \hat{z}) | z^* \in \hat{z}\right] \tag{8.5}$$

Provided the expectation in eq. (8.5) is finite, we can now consider neighboring databases D and D'. Since Δ_q bounds the change in q between D and D',

$$\mathbb{E}_{\hat{z}}\left[\phi(D', \hat{z}) | z^* \in \hat{z}\right] = \sum_{\hat{z}} \phi(D', \hat{z}) \Pr(\hat{z} | z^* \in \hat{z}) \tag{8.6}$$

$$\geqslant \sum_{\hat{z}} \frac{\phi(D, \hat{z}) \Pr(\hat{z} | z^* \in \hat{z})}{e^{\epsilon \Delta_q}} \tag{8.7}$$

Bounding each of the three terms in eq. (8.5) independently: the first by the definition of sensitivity, the second by independence, and the third by eq. (8.7),

$$\frac{\Pr(M(D) = z^*)}{\Pr(M(D') = z^*)} = \frac{e^{\epsilon\, q(D, z^*)} \Pr(z^* \in \hat{z})\, \mathbb{E}_{\hat{z}}\left[\phi(D, \hat{z}) | z^* \in \hat{z}\right]}{e^{\epsilon\, q(D', z^*)} \Pr(z^* \in \hat{z})\, \mathbb{E}_{\hat{z}}\left[\phi(D', \hat{z}) | z^* \in \hat{z}\right]} \tag{8.8}$$

$$\leqslant e^{\epsilon \Delta_q} \cdot 1 \cdot e^{\epsilon \Delta_q} = e^{2\epsilon \Delta_q} \tag{8.9}$$

which is precisely the definition for $2\epsilon\Delta_q$-differentially privacy. □ □

8.2.2 Proof of Accuracy

The paper that introduced the exponential mechanism McSherry and Talwar (2007) also established a bound on its accuracy. The theorem depends on a lemma relating S_t, the set of outcomes within t of the optimal q value, and $\bar{S}_{2t}$, the set of outcomes that are more than $2t$ away from the optimal value.

Lemma 8.3 (From McSherry and Talwar (2007))**.** *Assuming the sensitivity of* q *is* $\Delta = 1$*, for the exponential mechanism, where* $S_t = \{z \in \mathcal{Z} : q(D, z) > OPT - t\}$ *and* $OPT = \max_z q(D, z)$ *is the true optimal value among the set of outcomes,*

$$Pr(M(D) \in \bar{S}_{2t}) \leqslant \frac{e^{-\epsilon t}}{\mu(S_t)}. \tag{8.10}$$

where $\mu(S_t) = \int_{S_t} \mu(z)dz$*.*

Theorem 8.4 (From McSherry and Talwar (2007))**.** *For the exponential mechanism, where* t *satisfies* $t \geqslant \ln(\frac{OPT}{t\mu(S_t)})/\epsilon$*, we have*

$$\mathbb{E}[q(D, M(D))] \geqslant OPT - 3t \tag{8.11}$$

We will use a corollary to lemma 8.3 that provides a similar bound, but gives more flexibility.

Lemma 8.5. *For the exponential mechanism and* $w > 0$*,*

$$Pr(M(D) \in \bar{S}_{(1+w)t}) \leqslant \frac{e^{-\epsilon t w}}{\mu(S_t)}. \tag{8.12}$$

The proof is essentially the same as for lemma 8.3.

In the subsampled exponential mechanism, OPT might not be sampled, so we must modify the lemma as follows.

Lemma 8.6. *Assuming the sensitivity of* q *is 1, for the subsampled exponential mechanism we have*

$$Pr(M(D) \in \bar{S}_{2t}) \leqslant \frac{2e^{-\epsilon t}}{\mu(S_t)}. \tag{8.13}$$

provided the number of samples $m \geqslant \frac{1}{2}e^{\epsilon t}$*.*

Proof. Assuming points are drawn according to base distribution μ, we need to bound the probability that the selected $z^* \in \bar{S}_{2t}$. However, for the subsampled exponential mechanism, we have the additional step of

taking the sample $\hat{z}$, and only items in $\hat{z}$ can actually be selected. Let $a = \frac{\mu(S_t)}{\mu(\mathcal{Z})}$ be the probability mass in μ of the outputs in S_t. There are two possibilities for the sample $\hat{z}$, either it contains at least one output that is in S_t or it contains none. In the former case, which occurs with probability $1 - (1-a)^m$, we will bound the probability of selecting a $z^* \in \bar{S}_{2t}$. If $\hat{z}$ contains no elements of S_t, we make no assumptions on what the mechanism does. Since this occurs with probability $(1-a)^m$, we will need to add this term to our overall bound on $\Pr(\bar{S}_{2t})$.

Let $\widehat{OPT} = \max_{z_i \in \hat{z}} q(D, z_i)$ be the score of the best output that is selected in $\hat{z}$. Assuming the case that $\hat{z} \cap S_t \neq \emptyset$, $OPT \geqslant \widehat{OPT} \geqslant OPT - t$. Let $\hat{t} = \widehat{OPT} - (OPT - t)$. So $\hat{t}$ is the amount that $\widehat{OPT}$ is above the threshold $OPT - t$.

The subsampled exponential mechanism performs an exponential mechanism selection, but over just the sample $\hat{z}$. So we are going to use the bound in lemma 8.3 that applies to $\hat{z}$ and then show how that leads to the desired bound in $\mathcal{Z}$. In applying the exponential mechanism to $\hat{z}$, we have analogous version of S_t, but we will use

$$\hat{S}_{\hat{t}} = \{z_i \in \hat{z} : q(D, z_i) \geqslant \widehat{OPT} - \hat{t}\} \tag{8.14}$$

since it means

$$\widehat{OPT} - \hat{t} = OPT - t \tag{8.15}$$

However, we cannot directly apply lemma 8.3 because it only gives a bound for choosing an output where $q(D, z^*) \leqslant OPT - t - \hat{t}$, but $\hat{t} \leqslant t$ so this is not sufficient to bound the probability of $\bar{S}_{2t}$ which requires all outputs have score less than $OPT - 2t$. Instead we use the corollary in lemma 8.5 with $w = t/\hat{t}$. Using $\hat{\mu}$ for the uniform distribution on $\hat{z}$, we have

$$\Pr(\hat{\bar{S}}_{(1+w)\hat{t}}) \leqslant \frac{e^{-\epsilon \hat{t} w}}{\hat{\mu}(\hat{S}_{\hat{t}})} = \frac{e^{-\epsilon t}}{\hat{\mu}(\hat{S}_{\hat{t}})}. \tag{8.16}$$

The bound in eq. (8.16) is for the probability of selecting an output with score less than $\widehat{\mathrm{OPT}} - (1+w)\hat{t}$. Substituting for w and using eq. (8.15) means

$$\widehat{\mathrm{OPT}} - (1+w)\hat{t} = \widehat{\mathrm{OPT}} - \hat{t} - t \tag{8.17}$$

$$= \mathrm{OPT} - t - t = \mathrm{OPT} - 2t \tag{8.18}$$

So all outputs in $\hat{\bar{S}}_{(1+w)\hat{t}}$ have scores less than $\mathrm{OPT}-2t$, and thus $\hat{\bar{S}}_{(1+w)\hat{t}} \subseteq S_{2t}$. For a particular $\hat{z}$, no other elements from S_{2t} can possibly be chosen, so eq. (8.16) implies that when $\hat{z} \cap S_t \neq \emptyset$,

$$\Pr(\bar{S}_{2t}) \leqslant \frac{e^{-\epsilon t}}{\hat{\mu}(\hat{S}_{\hat{t}})}. \tag{8.19}$$

If we let $k = |\hat{z} \cap S_t|$, then $\hat{\mu}(\hat{S}_{\hat{t}}) = \frac{k}{m}$. Separating out each possible value of k using a binomial expansion when $\hat{z} \cap S_t \neq \emptyset$, we have

$$\Pr(\bar{S}_{2t}) \leqslant e^{-\epsilon t} m \sum_{k=1}^{m} \frac{a^k (1-a)^{m-k} \binom{m}{k}}{k} \tag{8.20}$$

$$\leqslant \frac{2e^{-\epsilon t} m}{a(m+1)} \sum_{k=1}^{m} a^{k+1} (1-a)^{m-k} \binom{m+1}{k+1} \tag{8.21}$$

eq. (8.21) results from multiplying by $\frac{2ka(m+1)}{(k+1)a(m+1)}$ which is $\geqslant 1$ when $k \geqslant 1$.

Let $i = k+1$ and $n = m+1$. The binomial theorem for $0 \leqslant a \leqslant 1$ can be written as $c_i = a^i (1-a)^{n-i} \binom{n}{i}$ and $\sum_{i=0}^{n} c_i = 1$. The sum in eq. (8.21) contains all but two of the terms in sum, so we subtract off those missing

terms: c_0 and c_1.

$$\Pr(\bar{S}_{2t}) \leqslant \frac{2e^{-\epsilon t}m}{a(m+1)}\left[\left(\sum_{i=0}^{n} c_i\right) - c_0 - c_1\right] \tag{8.22}$$

$$= \frac{2e^{-\epsilon t}m}{a(m+1)}\left[1-(1-a)^{m+1} - a(m+1)(1-a)^m\right] \tag{8.23}$$

$$\leqslant \frac{2e^{-\epsilon t}}{a}\left[1-(1-a)^{m+1} - a(m+1)(1-a)^m\right] \tag{8.24}$$

$$\leqslant \frac{2e^{-\epsilon t}}{a} - 2e^{-\epsilon t}(m+1)(1-a)^m \tag{8.25}$$

We drop the $-(1-a)^{m+1}$ term from eq. (8.24) to eq. (8.25) in order to get a looser, but simpler, bound at the end. eq. (8.25) is the bound for when $\hat{z} \cap S_t \neq \emptyset$. If that intersection is $\emptyset$, then we cannot make any claims about the probability of choosing a poor output. So we additionally have an additive term of $(1-a)^m$ from when the intersection is $\emptyset$.

$$\Pr(\bar{S}_{2t}) \leqslant \frac{2e^{-\epsilon t}}{a} - 2e^{-\epsilon t}(m+1)(1-a)^m + (1-a)^m \tag{8.26}$$

If $m \geqslant \frac{1}{2}e^{\epsilon t}$, then the second term in eq. (8.26) is larger than the additive third term, and we get final bound of

$$\Pr(\bar{S}_{2t}) \leqslant \frac{2e^{-\epsilon t}}{a} = \frac{2e^{-\epsilon t}}{\mu(S_t)} \tag{8.27}$$

□ □

Using lemma 8.6 leads to a modified version of the accuracy theorem from McSherry and Talwar (2007).

Theorem 8.7. *For the subsampled exponential mechanism, where* t *satisfies*

$t \geqslant \ln(\frac{OPT}{t\mu(S_t)})/\epsilon$ *and* $m \geqslant \frac{OPT}{2t\mu(S_t)}$, *we have*

$$\mathbb{E}[q(D, M(D))] \geqslant OPT - 5t \tag{8.28}$$

The proof is identical to theorem 8.4 as proven in McSherry and Talwar (2007), except using lemma 8.6 rather than lemma 8.3.

8.3 Experiments

To investigate the practical aspects of using the exponential and subsampled exponential mechanisms, we perform several experiments on a clustering task. Our primary investigation is the utility of the outputs produced by the various methods, in comparison to other private and non-private methods. Additionally, we explore the empirical behavior of the subsampled exponential mechanism as the number of samples varies and with alternative base distributions.

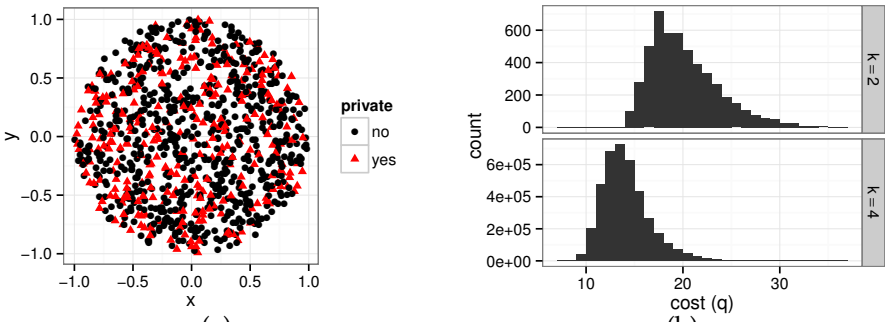


Figure 8.2: (a) Public data points (V) for $n = 1000$ with the $s = 300$ private data points in D denoted by red triangles. (b) Histogram of costs, the negative of the q function distribution, for our k-median task with $n = 100$ and $k = 2, 4$. Note that since the q function is actually the negative cost, the best outputs are on the left and the long right tail contains very poor clusterings.

8.3.1 K-median Task

We use the k-median (or, perhaps more accurately, the k-medoid) problem setting described in Gupta et al. (2010). This is a clustering task similar to k-means where the cluster centers must be chosen from among the data points, hence the name median. The centers are to be chosen among the public set of data points, V, with a distance metric, $d : V \times V \to \mathbb{R}$, over those points. The private data may be a subset, $D \subseteq V$, or may be a separate set of points in the same space (with d also defined between public and private points), the exact identity of which we want to keep private. The task then is to find the set of centers $F \subseteq V$ with $|F| = k$ that minimizes the distances to the private subset. That is, we want to minimize $cost(F) = \sum_{v \in D} \min_{f \in F} d(v, f)$.

While the brute-force approach of evaluating the cost of all $\binom{n}{k}$ subsets of size k will find the optimal solution, it is not practical for any but the smallest problems. A variety of non-private, heuristic algorithms exist for this problem, typically based around some sort of (greedy) local search. The prototypical approach is the partitioning around medoids (PAM) algorithm, an expectation-maximization-like algorithm made popular by Kaufman and Rousseeuw (1990). Another method that includes theoretical guarantees about the quality of the solutions in relation to the optimal is the local search with 1-swaps investigated in Arya et al. (2004). Private approaches seek to output a reasonably low-cost solution while ensuring the presence or absence of any particular point in D is sufficiently obfuscated. Gupta et al. discuss using the exponential mechanism on all subsets as well as their modified 1-swap search that sequentially composes a number of private steps to create a differentially private version of local search Gupta et al. (2010). After dividing up their budget ϵ to make a series of local search steps, the algorithm then uses the exponential mechanism once more to select the final set of centers.

The exponential mechanism and subsampled exponential mechanism

require a q function that indicates the quality of a particular solution F for a particular dataset D. As in Gupta et al. (2010), we use the negative of the cost function. The global sensitivity of this q function is the maximum change in the cost when removing, adding, or moving a single point in D. Changing a point in D can change the cost function by at most the maximum distance between two points in V, also known as the diameter. i.e., $\Delta = \max_{v,w \in V} d(v, w)$.

8.3.1.1 Algorithms

Results are presented for a variety of algorithms. These methods can be separated into three categories: data agnostic, differentially private, or non-private.

Data agnostic methods do not actually look at the private dataset, and instead only use properties of the public data to choose the centers. This means the algorithms are differentially private by default (for any ϵ), and can be used as input to other differentially private methods. The two data agnostic methods that we investigate are `random`, which simply chooses a random subset of size k from the n public points, and `random k++`, which uses the initialization algorithm from k-means++ Arthur and Vassilvitskii (2007). The initialization algorithm selects the first cluster center uniformly at random, then chooses each subsequent point weighted by the inverse square of the distance to the closest existing center. The result is that the centers tend to be relatively spread out, particularly for larger values of k.

We provide a baseline of the best solution found by non-private algorithms for each problem setting. For the smaller problems we can evaluates all $\binom{n}{k}$ subsets of the public points to identify the optimal solution with minimum cost. For the larger problems, we report the smallest solution found after 10 runs of the local 1-swap algorithm from Arya et al. (2004). Though this is not guaranteed to find the optimal solution and the theoretical bounds are quite loose, in practice it performs very well. In our

experiments, the local search always finds an optimal solution when we are able to calculate the optimal cost.

We evaluate several versions of the differentially private algorithms. First is the differentially private iterative method described in Gupta et al. (2010), which we refer to as `gupta`. Then, we have the full exponential mechanism over all $\binom{n}{k}$ subsets; this `em` method is only feasible on the smaller parameter settings. Finally, we have the subsampled exponential mechanism (`ssem`) with its additional parameter, m, for how many samples to take. As a relatively direct comparison to the iterative method, we set the number of samples to be the same as the number of iterations in `gupta` ($6k\ln(n)$) for the `ssemauto` method. These two approaches can be seen as the two extremes between using the privacy budget for the final choice with random samples (`ssemauto`) or using the majority of the privacy budget to obtain better samples via an iterative Monte Carlo method (`gupta`).

An often overlooked aspect of the exponential mechanism is the base measure. By changing the base measure to bias towards more typically useful outputs instead of a uniform distribution on outputs, it is possible to improve utility while preserving differential privacy. The critical aspect, though, is that the base distribution must be independent of the private data.

In the k-median task, for example, one expects good solutions to have centers that are reasonably spaced, and not bunched up together. But that intuition is difficult to encode in a probability distribution function. With the subsampled exponential mechanism, however, it is not necessary to formalize this intuition as a fully specified probability distribution function over the outputs. Instead, a sampling algorithm can be used that implies some probability distribution function over the output space. Thus, we can use the `random k++` algorithm mentioned above to probabilistically choose cluster centers for the public points (without reference to which points are in the private dataset) and use that as the base distribution

for `ssem` and `ssemauto`. These versions are referred to as `ssem k++` and `ssemauto k++`, respectively.

8.3.1.2 Synthetic Data

We compare and contrast the aforementioned methods in a simple, synthetic dataset modeled on the problem description from Gupta et al. (2010). $n = |V|$ points are uniformly chosen in the unit circle and a random subset of size s is used as the private data. See fig. 8.2a for an example of the points being clustered. We use $n = 100, 1000, 10000$ with associated $s = 30, 300, 3000$, respectively. The number of centers ranges over $k = 2, 4, 8, 16$. The utility of each method is assessed by the cost of the cluster centers produced. Median cost and quantiles are shown for 1,000 replicates of the randomized algorithms.

As discussed in section 8.2, the effectiveness of the subsampled exponential mechanism depends on the distribution of $q(\mathcal{Z})$. If the best output (the smallest cost for the k-median task) is at the end of a long tail of the distribution of the q function, then it will be difficult for the subsampling process to find good solutions. On the other hand, if the best output is near a good percentage of the density of the q function, then the subsampled exponential mechanism should perform well. Figure 8.2b contains the histograms of costs, i.e., the q function distribution, for two of the smaller k-median settings we consider. Since the goal is a minimum cost solution, the fact that both distributions are left-shifted and have long tails to the right, but not to the left, means the subsampled exponential mechanism should be effective at getting samples that are close to the optimal cost. If, for some unknown reason, the goal was to maximize the cost of the cluster centers, our method would not perform as well because it would be unlikely to sample any outputs in the long right tail.

8.3.2 Results

The first question we address is how the basic differentially private methods compare to each other and to the best solutions found via non-private means. In fig. 8.3a, we show results for the `em`, `gupta`, and `ssemauto` methods when only choosing 2 cluster centers ($k = 2$). With only $O(n^2)$ possible centers, we are able to run the `em` algorithm and can also find the optimal cost solution (denoted by the black horizontal line). The median cost of a randomly chosen subset is given by the dashed, gray horizontal line. The box plots show the 25th and 75th percentiles with middle line showing the median. The whiskers extending above and below to the minimum and maximum cost for any solution in the 1,000 replicates. If the notches for two algorithms do not overlap, this suggests their medians are statistically significantly differentChambers (1983).

When $\epsilon = 0.1$ and $n = 100$, none of the private algorithms do much better than random. Though they occasionally produce good solutions, they also can produce terrible solutions as evidence by the wide range of the whiskers, and the median is only slightly less than the median for a `random` solution. As ϵ or n increases, however, it is possible to protect privacy while providing lower cost solutions. We show results for ϵ of 10 and 100 to see if the algorithms improve with additional budget despite most discussion of differentially privacy focusing on ϵ of around 1 or lower. Indeed, as n increases, the utility improves considerably, and for $n = 10000$ and $\epsilon = 1.0$, the `em` method consistently produces solutions with nearly optimal cost. Figure 8.3a shows that the iterative `gupta` method requires large ϵ's to effectively select a good set of cluster centers. Except for a few cases when $\epsilon = 100$, `gupta` is worse than both the `em` and `ssemauto` methods. Despite using a subsample of the output space, our more computationally efficient `ssemauto` method (see discussion of runtimes in section 8.3.3 and fig. 8.5) performs similarly to the full exponential mechanism, and is typically much better than the `gupta` method. The few

times `gupta` is producing lower cost solutions than `ssemauto`, the `ssem` method can obtain comparable results by using more samples, while still running more quickly.

Next, we look at what happens as the number of cluster centers is varied. Figure 8.3b provides results for $k = 2, 4, 8, 16$ and has a similar structure to fig. 8.3a discussed above. Instead of the boxplots, we plot the median cost of solutions and the error bars show the 2.5% and 97.5% percentiles.

With $k > 2$, the full exponential mechanism is not feasible and the `gupta` method becomes prohibitively expensive for the largest parameter settings as well. These results are all for the same set of public/private points for each value of n, so the total cost for a particular n gets smaller as k increases and each private data point is closer to a center. We continue to see our `ssemauto` method performing about as well as `em` and similar to or better than `gupta`.

A critical parameter for our subsampled exponential mechanism is m: the number of samples to take. To explore the effect of m on performance in the k-medians task, fig. 8.3c shows the median cost of the solution produced by `ssem` as m varies. Additionally, we show the results for the `ssem k++` method which uses non-uniform sampling to bias the solutions.

The performance of the subsampled exponential mechanism improves or remains the same as m increases. When the median cost plateaus for `ssem` as m increases, it is most likely because it has attained as much utility as the `em` method. In the settings where we can run `em`, we demonstrate this in fig. 8.3d.

In the previous experiments, we assume that the private data points are drawn from the same distribution as the public points. What if this is not the case? In fig. 8.4, we show an experiment where the public points remain uniform but the private points are drawn from a gaussian distribution centered at (0.5,0.5). With the public points drawn from a

unit circle centered at (0,0), this provides both a different variance and mean for the private points. When the variance parameter of the gaussian is small, all private points are tightly packed, and the optimal solution has low cost. As the variance increases, the optimal solution has higher cost, though the cost of a random solution remains constant. In all cases, `ssemauto` is affected by the changing private density similarly to `local`.

8.3.3 Runtimes

In the previous section, we frequently alluded to where it was feasible to run certain algorithms and the relative computational efficiency of `ssem`. A rough presentation of selected algorithm runtimes is presented in fig. 8.5. We ran our experiments via a high throughput computing system[1] on a heterogeneous set of hardware, so precise time comparisons are not possible. However, by plotting the minimum time it took for any single replicate of a particular algorithm and parameter setting, we can approximately see what the performance is on the fastest machines in the pool. On the log-log scale in fig. 8.5, we see that the algorithms are generally scaling linearly in `n` (when `k` is held constant). Our subsampled exponential method is consistently faster than `gupta` ans scales better than `em`. Similar runtime results hold for the variations on the subsampled exponential method and as `k` is increased.

8.4 Conclusion

The subsampled exponential mechanism allows for differentially private queries to be made efficiently over large output spaces. In addition to the proof of privacy, we provided a theorem related to the accuracy of the subsampled exponential mechanism similar to the existing accuracy bounds

[1] HTCondor: `http://research.cs.wisc.edu/htcondor/`

for the exponential mechanism. This expands the range of problems that can be evaluated with the single application of a private mechanism. The `gupta` method is an iterative, private algorithm that involves many uses of the exponential mechanism, while our `ssem` approach consists of one private selection. As demonstrated in our experiments on the k-median task, the single private selection required in the subsampled exponential mechanism obtains better or similar utility solutions than the `gupta` method and runs more quickly. This is a vastly different approach than is typically used in non-private learning, but in private learning has the advantage of not needing to divide the privacy budget while allowing operation in extremely large output spaces.

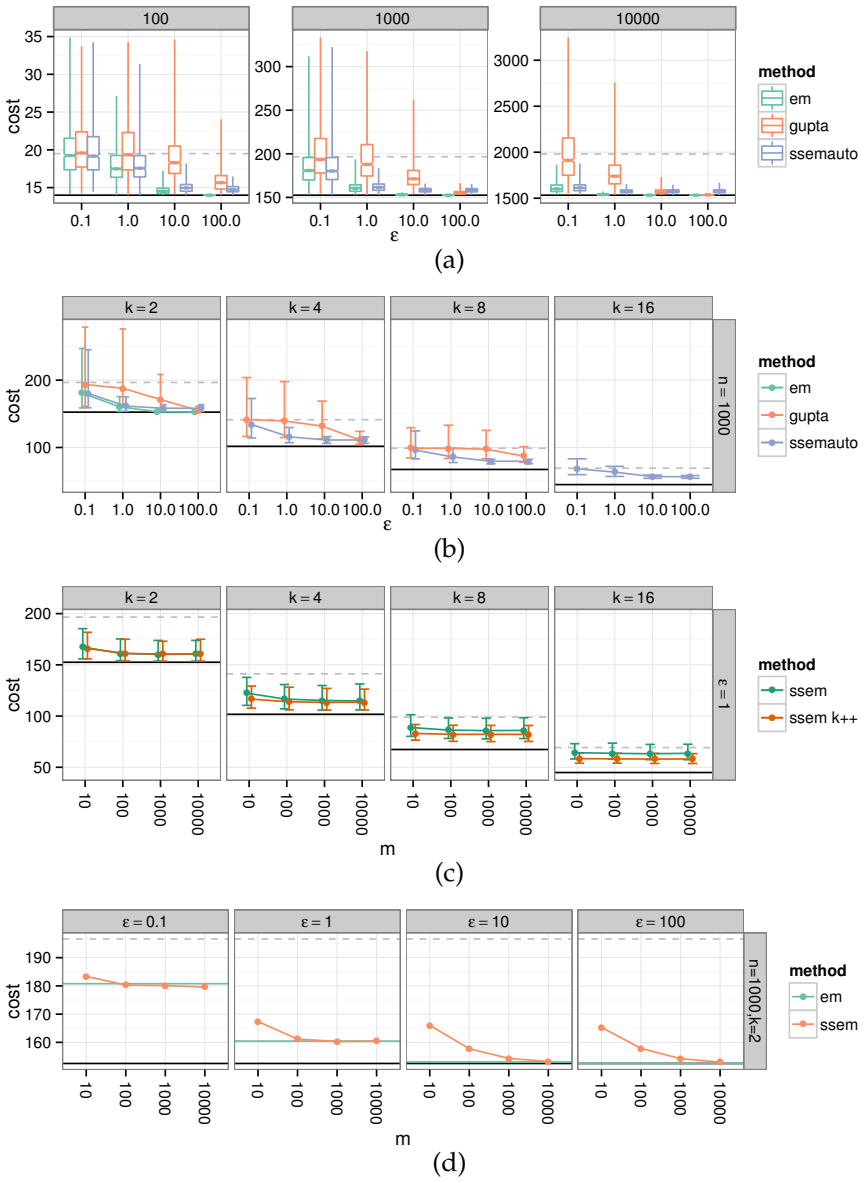


Figure 8.3: Median cost of differentially private solutions. Error bars are 2.5% and 97.5% quantiles. Missing points and error bars for `em` and `gupta` occur when runtimes or memory requirements were excessive. Cost of best solution found using non-private local search is shown by solid black line and the median cost of random solutions by the gray, dashed line. (a) Box and whisker plots comparing `em`, `gupta`, and `ssemauto` versus the privacy budget, ϵ. (b) Comparison of `em`, `gupta`, and `ssemauto` versus the privacy budget, ϵ for $n = 1000$. (c) Comparison of `ssem` and `ssem k++` versus the number of samples, m, used in `ssem` for $\epsilon = 1$. (d) Comparison of `ssem` with different number of samples, m, versus `em` shown as green horizontal line. $n = 1000$ and $k = 2$.

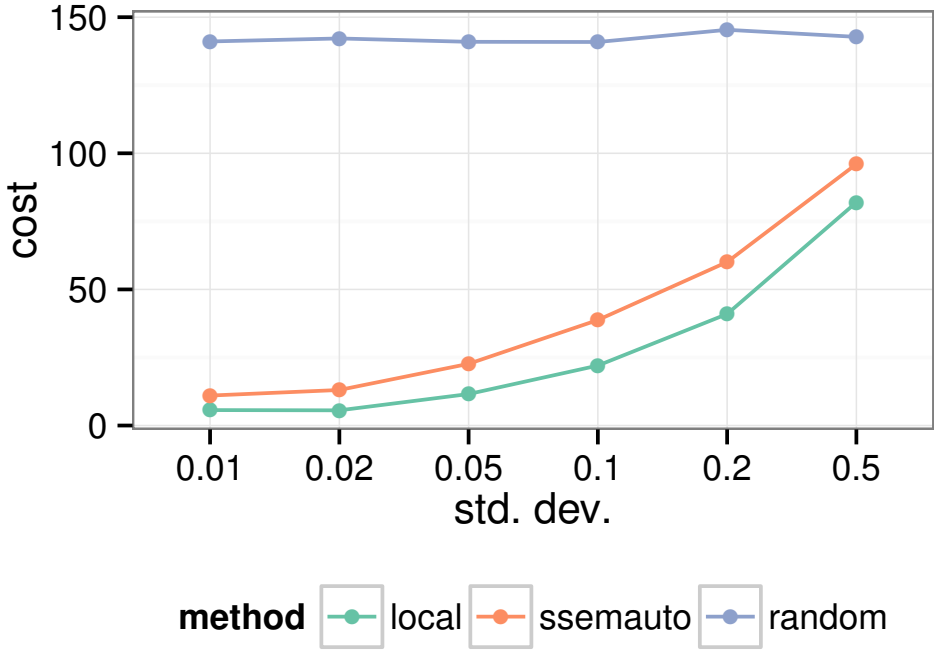


Figure 8.4: Median cost when the private distribution is drawn from a gaussian centered at (0.5,0.5) as the standard deviation of the gaussian changes. $n = 1000$, $k = 4$, and `ssemauto` run with $\epsilon = 1$.

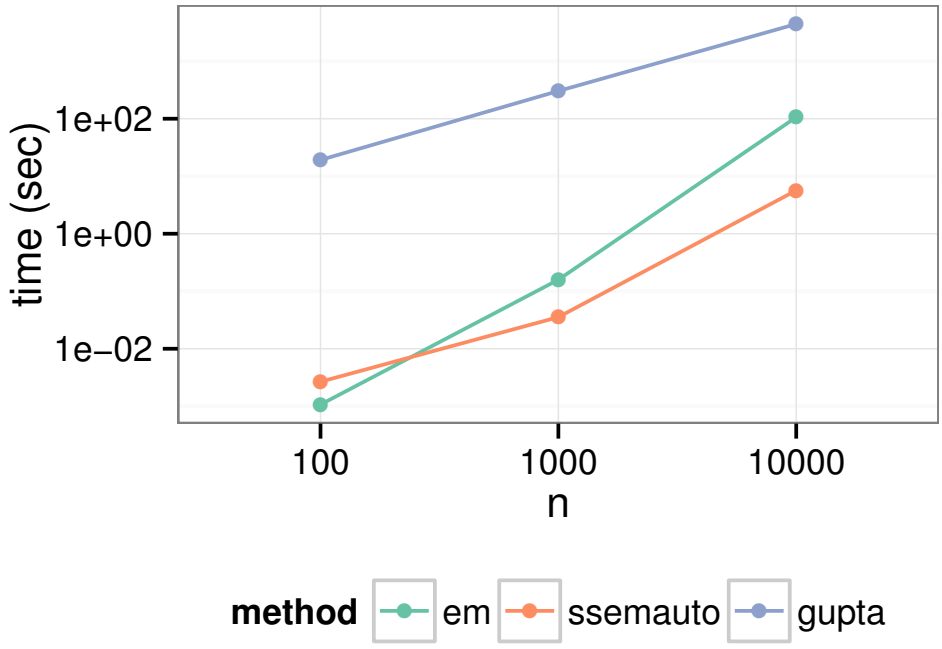


Figure 8.5: Log-scale runtime (in seconds) versus n for $k = 2$. The minimum time for a single replicate of an algorithm and parameter setting is plotted due to the heterogeneous set of machines upon which our experiments were executed.

9 ADDITIONAL EXPLORATIONS: BAYESIAN NETWORK STRUCTURE LEARNING FOR CORRELATION IMMUNE FUNCTIONS

This chapter contains an additional work that deviates from the central focus of this dissertation, but was performed as part of the doctoral work. We developed a method for learning Bayesian network structure in the presence of relationships that typically confound greedy learners. A version of this chapter was previously published as (Lantz et al., 2007).

9.1 Introduction

Bayesian networks (BNs) are an elegant representation of dependency relationships present over a set of random variables. The structure of the network defines a factored probability distribution over the variables and allows many inference questions over the variables to be answered efficiently. However, there are a super-exponential number of possible network structures that can be defined over n variables, and the process of finding the optimal structure consistent with a given data set is NP-complete (Chickering et al., 1994), so an exhaustive search to find the one that best matches the data is generally not possible. Techniques to learn BN structure from data must choose a way to restrict the search space of possible networks in order to gain tractability.

The most computationally efficient search technique traditionally employed to discover BN structure is a greedy search over candidate networks. Given a current network, a greedy search scores structures derived using local refinement operators, such as adding and deleting arcs, according to a score such as penalized likelihood. The search keeps only the best such structural modification to refine in the next iteration. However, as

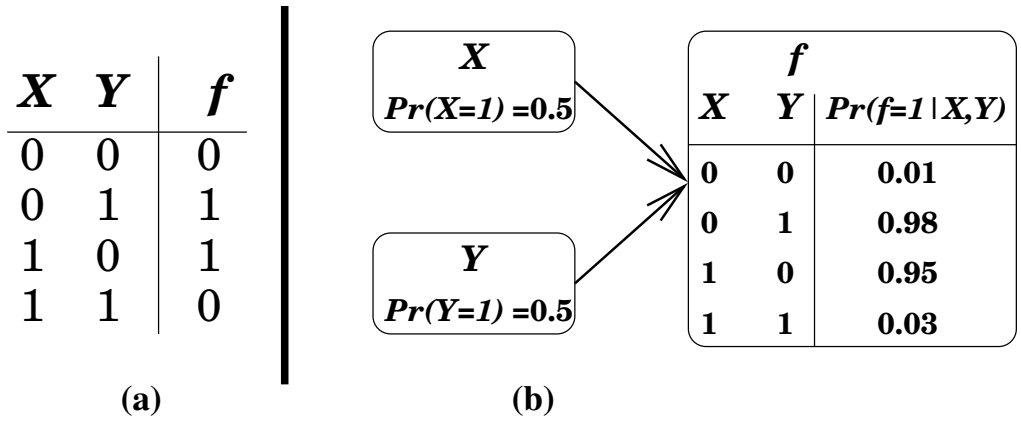


Figure 9.1: (**a**) Example of a correlation-immune function. (**b**) Example of an approximate correlation-immune relationship.

with greedy tree learning algorithms, this search is myopic. If the function between a node and its parents in a target Bayesian network is a type of function called correlation-immune (CI), or a relationship that is a stochastic variant of such a function, then adding one of the correct parents may not improve network score in the absence of the other parents. In such cases, the search algorithm may not be able to distinguish relevant refinements from irrelevant refinements and may be led astray. Hence CI relationships present a problem for greedy approaches to BN structure learning.

Correlation-immune (CI) functions (Camion et al., 1992) exhibit the property that when all possible function inputs and outputs are listed (for example in a truth table), there is zero correlation between the outputs and all subsets of the inputs of size at most c. Examples include exclusive-OR (Figure 9.1), parity, and consensus, among others. In BN terminology, a child node's probability of taking any particular setting is unchanged when conditioned on at most c of its parents.

In this work, we extend the commonly used Sparse Candidate (SC) al-

gorithm for BN structure learning to use a technique called skewing (Page and Ray, 2003). We empirically evaluate our algorithm on synthetic data sets generated by different network topologies, both with and without CI relationships. Our results show that, in most cases, our algorithm recovers the generating network topology more accurately than the standard Sparse Candidate algorithm. Further, the networks learned by our approach are significantly more accurate when CI relationships are present in the data.

9.1.1 Correlation Immunity

Consider a Boolean function f over n Boolean variables, $x_1, \ldots, x_n$. We say that f is *correlation-immune of order* c (Camion et al., 1992; Dawson and Wu, 1997) if f is statistically independent of any subset S_c of variables of size at most c: $\Pr(f = 1|S_c) = \Pr(f = 1)$. An example of such a function is 2-variable odd parity, shown in Figure 9.1(a). This function is correlation-immune of order one (or equivalently, a first order CI function), because for any subset of variables of size zero or one, the distribution over the values of f does not change. CI functions have been studied extensively in cryptography, where they are used as non-linear ciphers that are not subject to correlation attacks, in which the attacker measures statistics of the output as a method of gaining information about the key. In our work, we are interested in learning approximate CI relationships from data. Figure 9.1(b) shows a fragment of a BN, where the conditional probability tables (CPTs) of X, Y and f encode an approximate CI relationship between these variables.

CI relationships appear in real-world scenarios. For example in *Drosophila* (fruit fly), whether the fly survives is known to be an exclusive-OR function of the fly's gender and the expression of *SxL* gene (Cline, 1979). Similarly, during brain development in quail chicks, the *Fgf8* gene, which is responsible for organizing the midbrain, is expressed only in regions where neither or both of the genes *Gbx2* and *Otx2* are expressed (Joyner et al., 2000). This

behavior is an instance of *antagonistic repressors* – *Gbx2* and *Otx2* are repressors of *Fgf8*; however, they are also antagonistic – when they are both expressed, they repress each other. Such functions also arise in problems outside of genetics. For example, consider the task of predicting whether two proteins bind to each other. An important predictor of binding is the presence of regions in the proteins that are oppositely charged. Such a function is an exclusive-OR of features representing the charge on regions of the proteins: like charges repel, and thus hinder binding, while opposite charges attract, and thus facilitate binding.

The presence of (approximate) CI relationships in the data presents a challenge for machine learning algorithms that rely on greedy search to gain computational efficiency, such as the SC algorithm described below. This is because at some point in the search, no single feature appears to be relevant to the learning problem. To discover such relationships, *depth-c lookahead* can be used (Norton, 1989). This approach constructs all subsets of $c + 1$ features, and will find any target relationship that is correlation-immune of order at most c. However, the computational cost is exponential in c ($O(n^{2^{c+1}-1})$ where n is the number of variables), thus this approach can only be used to find small CI functions. Further, this procedure can result in overfitting to the training data, even when only lookahead of depth 1 is considered, because it examines so many alternatives during search (Quinlan and Cameron-Jones, 1995).

9.1.2 Sparse Candidate Algorithm

In our work, we are interested in learning probabilistic relationships between the attributes describing the data. To do this, we use the well-known Sparse Candidate algorithm (Friedman et al., 1999), which we review here. The algorithm controls the structure search by limiting the number of parents that will be considered for each variable. The algorithm begins with an initial structure, typically one with no edges. It proceeds by al-

ternating between two phases: a ***restrict*** phase to decide which variables will be considered potential parents (candidates) of each variable, and a ***search*** phase in which greedy structure modifications are made using the candidates and existing structure. The entire algorithm terminates when a search phase fails to make any changes to the structure.

The restrict phase performs a simple test on all pairs of variables in order to reduce the number of actions that need to be considered in the next phase. It limits each variable to a maximum of k candidate parents. For example, if node Y is a candidate parent of node X, the next phase of the algorithm will consider adding the directed arc $Y \rightarrow X$. The measure of the strength of the correlation between the two variables is the information theoretic measure conditional mutual information $I(X;Y|\mathbf{Z})$ as estimated from the data.

$$I(X;Y|\mathbf{Z}) = \sum_{x}^{X} \sum_{y}^{Y} \sum_{\mathbf{z}}^{\mathbf{Z}} \hat{p}(x,y,\mathbf{z}) \log \frac{\hat{p}(x,y|\mathbf{z})}{\hat{p}(x|\mathbf{z})\hat{p}(y|\mathbf{z})} \tag{9.1}$$

$\mathbf{Z}$ is the set of parents of X. If X has no existing parents, $\mathbf{Z} = \phi$ and the equation becomes mutual information. $\hat{p}(x,y|\mathbf{z})$ is the observed joint probability of x and y given the settings of $\mathbf{z}$.

Mutual information (or its conditional variant) is calculated for each pair of variables. For each variable, the current parents are added to the candidate set. Then the candidates with the highest (conditional) mutual information are added until the candidate set the contains k variables. The restrict phase outputs the list of k candidates for each variable.

The search phase consists of a loop to greedily build the best network given the current candidate sets. There are three search operators: add an arc to a variable from one of its candidate parents, remove an existing arc, or reverse the direction of an arc. Each addition or reversal is checked to ensure that directed cycles are not created in the network. All remaining actions are scored, and the best action is taken. Common scoring metrics,

including Bayesian-Dirichlet metric (BD) (Heckerman et al., 1995) and Bayesian Information Criterion (BIC) (Schwarz, 1978) include some way of trading off data likelihood with model simplicity. The important criterion of the metric for computational efficiency is that it be decomposable – the contribution of a variable to the score is dependent only on itself and its parents. When an action is considered, the score needs to be recalculated only for the variables whose parents have changed.

The search phase continues until the score is not improved by any available action. If changes have been made to the network during this phase, the algorithm then returns to the restrict phase and chooses new candidate sets based on the current network dependencies. If no changes were made, the algorithm terminates.

9.1.3 Combining Skewing and the Sparse Candidate Algorithm

The SC algorithm has two greedy components. The restrict phase looks only at pairwise relationships between variables when choosing candidates, and is greedy because only direct dependence is considered. The search phase chooses actions based on their local effect on the score. Both of these are limiting factors in learning approximate CI relationships. If we have data generated by such a relationship, the restrict phase of Sparse Candidate is unlikely to select the correct variables as candidate parents (unless there are no other variables in the model) because the mutual information score will be close to zero. Even if the correct variables are chosen, the search phase will not add a single such variable as a parent, because doing so will not improve the score of the structure under any of the previously mentioned scoring functions unless the other correct variables have already been added as parents.

To integrate skewing into the restrict phase, a skew is created as follows. For each variable x_i, $1 \leqslant i \leqslant n$, we randomly, uniformly (independently

for each variable) select a "favored setting" v_i of either 0 or 1. We also select a skew factor $\frac{1}{2} < s < 1$. For all variables, the weight of each example is multiplied by s if x_i takes the value v_i, and $(1 - s)$ otherwise.

$$w_e = \prod_{i=1}^{n} \begin{cases} s & : D(e, i) = v_i \\ 1 - s & : D(e, i) \neq v_i \end{cases} \tag{9.2}$$

At the end of this process, each example has a weight between 0 and 1. It is likely that each variable has a significantly different weighted frequency distribution than previously, as desired. We can then define the probability of a variable X taking on a certain value x.

$$\hat{p}_{skew}(X = x) = \frac{\sum_i w_i | D_{iX} = x}{\sum_i w_i} \tag{9.3}$$

This reduces to standard frequency counts when all weights are set to 1. We score the correlation between two variables by averaging the skewed conditional mutual information (Equation 9.1) over $T_1 - 1$ skews plus the original distribution, for a total of T_1 distributions.

$$I_{skew}(X; Y|\mathbf{Z}) = \frac{\sum_t^{T_1} I(X; Y|\mathbf{Z}, \vec{w_t})}{T_1} \tag{9.4}$$

where $I(X; Y|\mathbf{Z}, \vec{w_t})$ is computed by substituting the $\hat{p}_{skew}$ in Equation 9.3 into $\hat{p}$ in equation 9.1. Similarly, the search step evaluates each of its possible actions (adding an arc from a variable to one of its candidate parents, removing an arc, or reversing an arc) and chooses the best one according to a decomposable scoring function. Even if a relevant parent is chosen as a candidate in the restrict step, the scoring function – which looks at statistics of the original distribution – will still score the action poorly. So skewing is also needed when evaluating each action. We generate $T_2 - 1$ additional skewed distributions and apply a modified scoring function that takes into account $\vec{w}$. The BD metric calculates the number of times each pair of variables occurs with each combination of their values. This is

adjusted so that the counts are equal to the weight of the example in which they occur. As in the restrict step, we take the average of the structure scores over all skews before choosing the next action.

The phases of the SC algorithm are shown in Algorithms 6 and 7, with changes due to skewing shown in bold. The first "skewed" distribution in both phases is the original distribution, represented by using a vector of ones for the weight. In both phases, the calculations are affected by the vector $\vec{w}$ produced in creating the skewed distribution. Taking the average result over all skewed distributions serves to preserve the signal from strong relationships, but mitigate the effect of spurious relationships which achieve high scores as the result of a particular skew.

Since we are using multiple distributions, it is not clear how to determine the end condition of the search phase. If we score the modified structure against the original distribution within the search phase (as in normal SC), the search may terminate prematurely because the modification may result in a worse scoring structure if it was part of a correlation immune relationship. Continuing as long as the score improves on the skewed distributions is also problematic, as skewing may cause arcs to be added to the network that are irrelevant to the original distribution. We chose to terminate the search phase when the best move has less than half of the improvement of the first move. This puts bounds on the search and requires strong signals for network modification.

The restrict and search phases alternate, just as they do in normal SC, until the score of the network on the original distribution does not improve with a search phase. Throughout this process, the skewing procedure has used a variety of distributions in order to identify relevant parents. Nevertheless, we want to model the true distribution, not the skewed distributions. Therefore the algorithm closes by running normal SC on the original distribution, but using the structure built from skewing as the initial structure. This step greatly improves precision by removing arcs

Algorithm 6: Sparse Candidate Restrict Phase. Adapted from Figure 2 in Friedman et al. (1999). Changes due to skewing shown in bold.

Require: A matrix D of m data points over n variables, number of candidates k, initial network B_τ
Ensure: For each variable x_i a set of candidate parents c_i of size k
1: $\tilde{\mathbf{w}}_1 \leftarrow \tilde{\mathbf{1}}$
2: **for t** $\leftarrow$ **2** to $\mathbf{T_1}$ **do**
3: $\quad \tilde{\mathbf{w}}_t \leftarrow \mathbf{Skew}(\mathbf{D})$
4: **end for**
5: **for** $i \leftarrow 1$ to n **do**
6: $\quad$ **for t** $\leftarrow$ **1** to $\mathbf{T_1}$ **do**
7: $\quad\quad$ Calculate $I(x_i, x_j | Pa(x_i), \tilde{\mathbf{w}}_t)$ for all $x_j \neq x_i$ and $x_j \notin Pa(x_i)$
8: $\quad$ **end for**
9: $\quad$ Choose the $k - l$ variables with the highest $\mathbf{I}_{skew}$ **over all skews**, where $l = |Pa(x_i)|$
10: $\quad$ Set $c_i = Pa(x_i) \cup \{k - l \text{ chosen variables } \}$
11: **end for**
12: **return** $\{c_i\}$

induced by idiosyncrasies of particular skewed distributions.

It is difficult to compute computational complexity of the SC algorithm or its skewed variant, due to the unknown number of iterations. However, we can say something about the effect of skewing on the complexity of each phase. The restrict phase of SC is $O(n^2)$, where n is the number of variables, due to the calculation of pairwise (conditional) mutual information scores. With skewing, it becomes $O(T_1 n^2)$. The search phase will undergo an unknown number of iterations, but the process of choosing the action is $O(kn)$. Skewing raises that to $O(T_2(kn))$.

9.1.4 Experiments

In this section, we discuss the evaluation of the effectiveness of skewing with or without CI relationships. We expect skewing to have a strong

Algorithm 7: Sparse Candidate Search Phase. Changes due to skewing shown in bold.

Require: A matrix D of m data points over n variables, initial network B_τ, candidate parents $\{c_i\}$
Ensure: Network $B_{\tau+1}$
1: $\mathbf{\tilde{w}_1} \leftarrow \mathbf{\tilde{1}}$
2: **for t** $\leftarrow$ **2** to $\mathbf{T_2}$ **do**
3: $\quad \mathbf{\tilde{w}_t} \leftarrow \mathbf{Skew}(\mathbf{D})$
4: **end for**
5: **repeat**
6: $\quad B_{\tau+1} \leftarrow B_\tau$
7: $\quad$ **for t** $\leftarrow$ **1** to $\mathbf{T_2}$ **do**
8: $\quad\quad$ Calculate $Score(B_\tau, action|D, \mathbf{\tilde{w}_t})$ for all possible actions
9: $\quad$ **end for**
10: $\quad$ Apply action with highest **average score over all skews** to B_τ
11: **until** Score improvement threshold not met (see text)
12: **return** $B_{\tau+1}$

advantage over normal SC when CI relationships are present in the generating network, and that advantage will also be present with approximate CI relationships. Additionally, we expect skewing to not decrease the effectiveness of SC in networks which do not contain CI relationships.

For all experiments, we constructed Bayesian networks of Boolean variables. Training data and test data were sampled uniformly from the network. We set $T_1 = 30$, $T_2 = 30$, $k = 6$, and the test sets contained 1000 samples. The skewing weight factor, s, was randomly chosen in each skew. To account for the randomness implicit in the algorithm, skewed SC was run 5 times on each network. The scoring metric used in the search was K2 (Cooper and Herskovits, 1992), a version the BD metric, with a structure term that penalized based on the number of parameters in the network and the size of the training set ($\sum_i 2^{|pa(i)|} \log|D|/2$).

We used two measures to evaluate the effectiveness of our algorithm on synthetic data. The first is the log likelihood of the model given the test

data, which describes how well the data appears to have been generated by the model. We also wanted to look at whether the correct arcs of the generating structure were being discovered by the algorithm. Unfortunately, most CI functions are statistically invariant as to which variable is the "output". For example, Figure 9.1(a) could represent $f = X \oplus Y$, $X = Y \oplus f$, or $Y = f \oplus X$, and the difference is impossible to determine solely from data. So instead of looking for the exact directed arcs, we compare the Markov blankets of the generating structure and learned structure. The Markov blanket of a variable X consists of X's parents, X's children, and the other parents of X's children. The Markov blankets for all variables will be the same in all output variations of correlation immune functions. In order to penalize both missing and superfluous arcs, we calculate the F1 score of the Markov blanket of all variables. Precision is the fraction of Markov blanket variables returned by the algorithm that are present in the generating structure. Recall is the fraction of Markov blanket variables in the generating structure that are returned by the algorithm.

The first synthetic network type consisted of 30 variables, with one variable having 5 parents and all others having no parents. For all parents P(X=1) = 0.5, whereas the probabilities of other variables were randomly assigned. The CPT of the child variable represented either a CI function or a function constructed by randomly selecting the output for each row of the truth table. The function representation could be exact (probability of 1 for the function value and 0 otherwise) or approximate (function value having a probability of 0.9 or 0.8). Approximate CI relationships are equivalent to noise in the value of the child variable.

Figure 9.2 shows learning curves for these experiments as a function of the number of examples in the training set. In terms of both likelihood and Markov blanket F1 score, skewing greatly outperforms normal SC on CI data sets. The difference between the two algorithms on the exact functions is statistically significant at the 99.9% confidence level by a

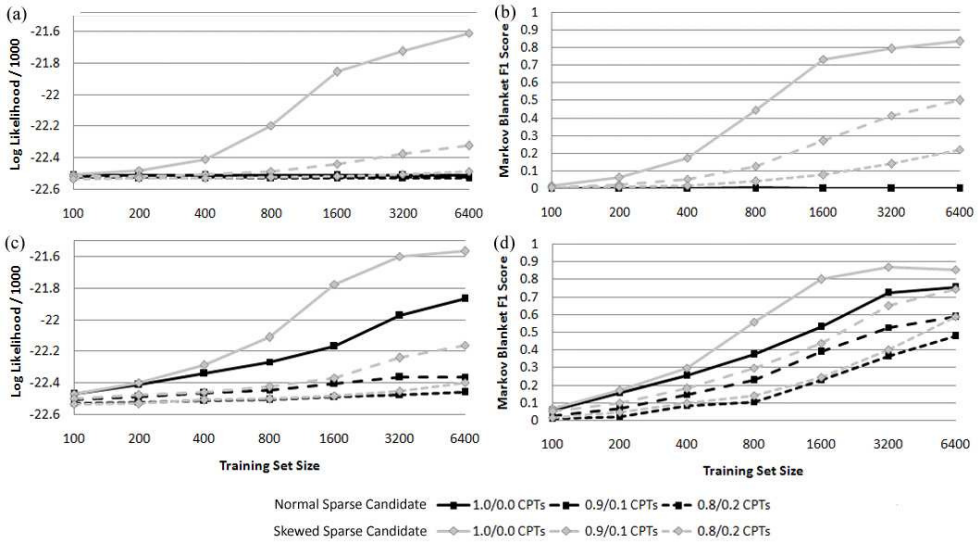


Figure 9.2: Learning curves on 30-variable data sets. Each data point is the average of 100 generated data sets. (top) Performance of normal and skewed SC on CI 5-variable functions as measured by log likelihood (left) and Markov blanket F1 (right). (bottom) Performance on random 5-variable functions by log likelihood (left) and Markov blanket F1 (right).

two-tailed t-test under both measures when the training set size $\geqslant 400$. Normal SC fails to improve despite more training data being available. Interestingly, skewed SC also outperforms normal SC for randomly generated functions. This can be explained by noting that randomly generated functions could contain CI subproblems (of 2, 3, or 4 variables) or be CI functions themselves. The difference between the two algorithms on the exact functions is significant at 95% confidence for training set size $\geqslant 1600$. Skewing shows some robustness to approximate CI relationships, particularly when measured by Markov blanket F1 score. However, for every 10% reduction in the probability of the CPT returning the function value, scores fall by more than half as compared to the baseline in all cases.

Another synthetic network type was inspired by the Quick Medical

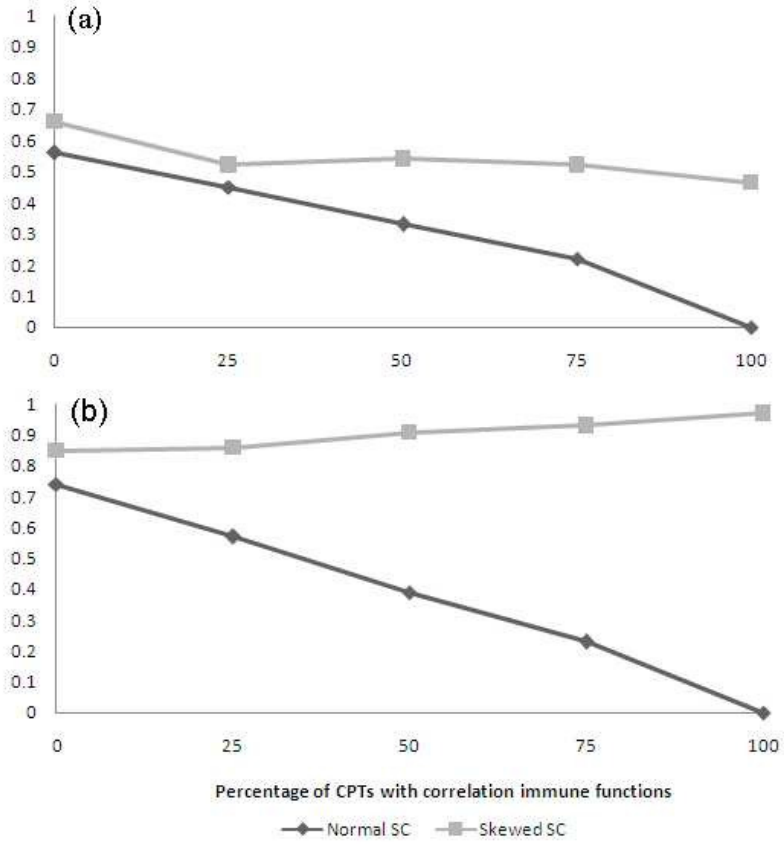


Figure 9.3: Markov blanket F1 scores on synthetic QMR-like data as function of percentage of CPTs with CI functions. (left) Unconstrained learning. (right) Constrained to only allow arcs from top layer to bottom layer.

Reference (QMR) network structure (Shwe et al., 1991) as a representation of disease diagnosis. The structure consists of a layered bipartite graph, with directed arcs only from the top layer to the bottom layer. The bottom layer nodes represent symptoms. The nodes in the top layer represent diseases or conditions, with arcs towards the symptoms they influence. It is possible to imagine conditional probability tables for the nodes in the lower level that are correlation immune functions like exclusive-OR or 3

variable parity. The generating networks contained 20 top layer variables and 20 bottom layer variables, with bottom layer nodes having 2 or 3 parents. We examined how well normal and skewed SC could reconstruct the structures with varying probability that a given bottom layer node would have a CPT that represented a correlation immune function. Figure 9.3(a) shows that skewing outperforms normal SC as measured by Markov blanket F1 score, and while both algorithms suffer as more correlation immune CPTs are present in the generating structure, skewing continues to be more accurate. When all bottom layer nodes have correlation immune function CPTs, normal SC is unable to discover any true arcs.

Additionally, we considered the effect of adding prior knowledge to the structure learner in the form of labeling the nodes as belonging in the top or bottom layer, and allowing arcs only from top layer to the bottom layer. Since the algorithms are prevented from making certain types of errors, we would expect this to improve scores. Figure 9.3(b) shows that the Markov blanket F1 scores are indeed improved for both versions of SC, but the performance of skewing now improves as more of the nodes are correlation immune functions, and the Markov blanket F1 (which is very close to the F1 of the returned structure due to the limitations on allowed arcs) reaches 0.975. In both graphs, skewed SC outperforms normal SC even when there are no correlation immune relationships present.

Another algorithm that has had some success learning parity-like targets is Optimal Reinsertion (Moore and Wong, 2003), which avoids the pitfalls of greedy search by using a multi-edge search operator. A node in the graph is chosen, and all edges in and out of that node are removed. Then a constrained optimization is done over possible edge combinations for the node, using AD-trees to make the search more tractable. Since this operation can change more than one edge, it has the potential to learn CI functions like parity. The authors evaluate their algorithm on three artificial datasets drawn from 36-node boolean networks 9.4, in which the

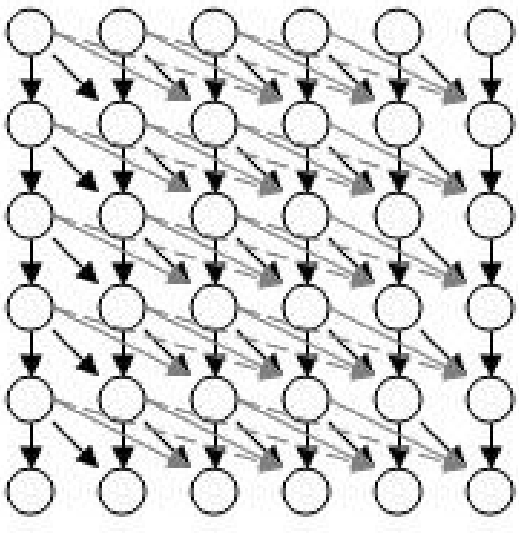


Figure 9.4: The synthetic networks from Moore and Wong (2003). Synth2 contains edges in black. Synth3 also contains edges in gray with solid lines. Synth4 contains all edges.

a node takes the value of the parity of its parents with probability 0.9. In synth2 each node has at most 2 parents from the previous level, synth3 and synth4 have at most 3 and 4, respectively.

We compared the performance of Optimal Reinsertion (OR) against SC and SC with skewing on these three datasets. The results are shown in table 9.1. OR outperforms SC by all metrics (except Markov blanket precision) by a large margin. SC with skewing significantly outperforms OR in all metrics, demonstrating a more complete reconstruction of the generating network.

Table 9.1: SC and OR are deterministic in their current implementations. SC with skewing was averaged over 12 runs with standard deviations below in parentheses. Avg L means average likelihood. Numbers in bold denote that the difference between SC with skewing and the other two methods is significant as measured by a two-tailed t-test at 95% confidence. For OR, the *max_secs* parameter, from which it derives other parameters, was set to 100000. All datasets had 25000 examples.

	Synth2			Synth3			Synth4		
	Prec	Rec	Avg L	Prec	Rec	Avg L	Prec	Rec	Avg L
SC	0.333	0.063	-32.49	0.333	0.036	-32.50	0.333	0.025	-32.50
OR	0.182	0.863	-22.71	0.203	0.729	-24.69	0.267	0.615	-26.06
SkSC	**0.589**	**0.996**	**-20.34**	**0.445**	**0.888**	**-21.62**	**0.416**	0.673	**-22.96**
	(0.11)	(0.01)	(0.27)	(0.05)	(0.12)	(1.53)	(0.03)	(0.12)	(1.33)

9.1.5 Applicability to medical data

While the QMR structure does have an intuition behind it, there is the question of how often CI functions would appear in the relationships between diseases and symptoms. QMR-like structures are more commonly modeled with a noisy-OR relationship, as it is generally thought that only one of the diseases must be present for the symptom to present itself. We would require a circumstance in which two conditions cause the same symptom, but having both of them causes no symptoms. There may be cases of this, like when one disease raises blood pressure and another lowers it, resulting in blood pressure appearing unchanged when the patient had both conditions. Nonetheless, while skewing likely is applicable to data from sources such as genetics, it is not clear if it can help elucidate relationships in medical domains.

10.1 Future Work

10.1.1 Timeline Forests

The models for Sections 3.3 and 3.4 used random forests as the machine learning model. Random forests perform well on many tasks, including those we explore in prediction from medical records. However, they don't fully take into account the temporal nature of EHR data. Also, while individual trees are relatively human-interpretable, the weighted combination of dozens or hundreds of trees is not. The problem is exacerbated by the fact that the trees in a random forest are typically not pruned, often resulting in trees with larger depth than those learned by decision tree algorithms like ID3 and C4.5.

To address these issues, we can create forests that operate on the timeline itself, selecting tree splits based on intervals in the past. By modifying standard variable importance techniques for random forests, we are also able to present a heat map of how the importance of a given variable varies over time, improving interpretability of the model.

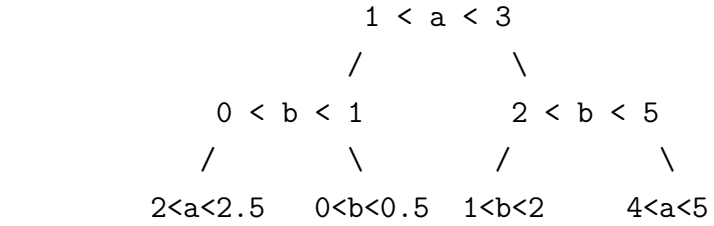
We use the term "timeline" to distinguish from "time-series." Unlike the latter, the timestamps do not come at regular intervals. In addition, we allow each example to have a different number of events, and multiple events can occur at a single timestamp. These aspects of the problem make it difficult to transform into fixed-length models.

We propose the timeline random forest, that acts directly on the timeline structure of the data. When selecting a split variable, the algorithm evaluates the possible time windows for that variable, directing an example down the left branch if the example contains the symbol within the interval, and down the right branch if it does not.

Since an example can have multiple instances of a symbol, the impurity

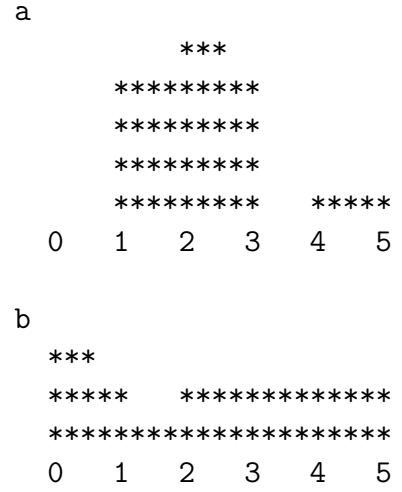
metrics apply to the frequency of the symbols themselves, rather than the frequency of the examples with each symbol. However, impurity is still weighted by the number of examples in each branch.

The use of a window, rather than a single time split, provides two advantages. First, it provides a better formalism for the example that does not contain the symbol at all. It is unclear whether an example without x should go down the $x < t$ branch or the $x \geqslant t$ branch. Second, it produces an additional dimension to variable importance scoring. When calculating impurity-based importance, we implicitly produce a set of weighted split points for each variable. After summing up the weights, we can find the normalized importance of each time period based on the weights of the windows that overlap at that point.



As an example, imagine a tree consisting of two variables, shown above. Assume that each node splits the remaining examples exactly in half. In addition to knowing the relative importance of variables a and b, we can determine the periods in which each is important. Considering the number of examples present at each node, we can build a histogram (or heatmap) of each variable over time.

We see that the bulk of the weight for variable a is located between the values of 1 and 3. The importance of variable b is more dispersed between 0 and 5, despite the fact that no node splits at this interval. In other words, while both a and b are important, it is more important that a occurs within a particular time interval, while b can occur at any time (i.e. the presence of b is more important than the time of occurrence).



10.1.2 Differentially Private Decision Tree Ensembles

Another extension to random forests relevant to this dissertation is to look at differentially private random forests and other tree ensembles.

There are many ways to produce an ensemble classifier from multiple decision trees. The standard random forest algorithm makes the following choices:

- Bagging each tree is trained independently on a bagged sample of training data (i.e. a dataset of size |D| is used with examples drawn with replacement).
- Feature subselection At each split, only a subset of the features may be chosen.
- No pruning
- Predictions from all trees are averaged.

This is not the only way to produce ensembles. Other tree ensembles make different choices. Some models use decision stumps, where all trees

are of depth 1. In boosted trees, the trees of the ensemble are not independent, but are instead learned sequentially with the examples reweighted based on the errors of the previous model. Oblique trees change the assumption that a single variable is used to split at each node in the tree, and instead use a linear combination of features.

Two previous papers (Friedman and Schuster, 2010; Jagannathan et al., 2012) have examined the issue of learning differentially private decision trees. Each utilizes a somewhat different method to achieve differential privacy.

In Friedman and Schuster (2010), a few variants of differentially private decision trees are presented. The first is based on ID3, and uses Laplace noise to calculate noisy counts for the number of examples at a node containing a given attribute value. This method requires a large number of noisy count calculations, and divides the privacy budget many times, leading to poor performance. They next present a variant of C4.5 that uses the exponential mechanism to choose the variable to split on based on the sensitivity of the node purity quality function. Several purity metrics were analyzed, including information gain, gini, gain ratio, and maximum class. They also introduce methods for pruning and operating on continuous attributes.

A very different approach was used in Jagannathan et al. (2012). They look at producing decision trees with random variables chosen at each split point, with a predetermined depth. Training examples are then traced down the tree, resulting in counts for each class at the leaves. Laplace noise is added to these counts to make the resulting tree differentially private.

A major limitation to ensemble learning in differential privacy comes from the composition theorem (2.4). The privacy budget must be divided among each of the trees, as all trees potentially examine the entire dataset. It is often true that adding an additional tree does not improve the accuracy

enough to counteract the reduction in the budget from $\frac{\epsilon}{k}$ per tree to $\frac{\epsilon}{k+1}$ per tree.

There are several methods to deal with this that might improve performance. The first is to assume the dataset size is large enough to partition the dataset into k independent parts, and learn one tree for each partition. Since each example will be found in at most one partition, it is not necessary to divide up ϵ. A second method related to the first is to take advantage of the fact that bagging doesn't select all examples in each sample. In fact, on average the percentage selected is $1 - \frac{1}{e}$ or about 63.2%. If bagging is done pseudo-randomly, the remaining 36.8% of examples can be reallocated to other trees without a penalty on ϵ. Lastly, there are tighter bounds on the composition theorem than simply dividing the budget by k. In the context of k-fold composition, Kairouz et al. (2015) show an improvement by a logarithmic factor.

10.1.3 Deep Private Recurrent Neural Networks

In Chapter 4, I presented EHR embeddings, which produce vector representations for the various codes used in the EHR. These embeddings provide a measure of the relationships between different codes, both within and between coding systems. However, the model presented in the chapter utilized the embeddings only via clusters, not the representations themselves. We desire a model that could take into account the exact vector representations into the model, as well as time between them.

Continuing the inspiration from word embeddings, similar tasks in natural language processing have been modeled with recurrent neural networks (Mikolov et al., 2010). In a recurrent model, inputs to the hidden layer at time step t are both the current embedding (e.g. word or code) and the hidden layer at time step $t-1$. At the end, the final hidden layer is used to predict the class of the example. To prevent the signal from being dominated by the last input, a memory-like structure can be used in the

hidden layer, such as long short-term memory (LSTM). LSTM has been successfully used in speech and time-series prediction tasks (Graves et al., 2013). Additionally, various deep RNN structures can be used, where one hidden layer feeds into another.

Like other temporal models, such as HMMs, the RNN still assumes that the data is a time series with relatively regular intervals. As mentioned previously, the EHR data is not regular, and may cooccur. One way to deal with this is to add two additional variables in the input representation along with the code embedding. One is the time between the current code and the next code, and the other is the time between the current code and prediction time. This could allow the model to weight the effect of a given code relative to the density of codes at a given time and the time until prediction.

It is possible to allow the model to change the embeddings of the codes themselves, but it is unclear if this would be useful or not. For a given prediction task, there are may only be a few thousand patients in the positive class. However, the embeddings can be learned from the entire patient population of potentially millions, allowing data for the entire population to be shared among the prediction tasks.

While the RNN approach is promising in its own right for prediction tasks like those in Chapter 3, it also opens up new possibilities for the application of differential privacy. There has been recent work looking at the issue of learning deep neural networks under differential privacy (Shokri and Shmatikov, 2015; Phan et al., 2016). While the analysis of such models is difficult, there are some promising early results. In addition, there is indication that dropout, a regularization technique where half the nodes are randomly shut off for each example during training, may provide differential privacy benefits on its own (Jain et al., 2015).

10.2 Summary

This dissertation presents investigations into the connected fields of machine learning and data privacy, particularly as related to medical prediction in EHRs. In support of the thesis statement, Chapter 3 presented examples of the promise machine learning has in medical prediction. Myocardial infarction, warfarin dosing, and atrial fibrillation can all be predicted using coded data from within the electronic health record. These are but three examples of the potential diagnoses and treatment decisions within medical practice. In addition, we presented an alternate way to represent the coded data within the EHR, allowing semantic relationships to be captured between events.

As EHRs contain a wealth of sensitive information, we study the use of differential privacy as a potential technique to mitigate the risks. We show that in a simulated clinical trial, we are unable to find an operating point in which there is a reasonable privacy/utility tradeoff. Next, we show that differentially private ILP is possible, but is quite limited in scaling with the number of clauses. These two results demonstrate the work that needs to be done to make differentially private methods practical.

As a first step towards improving the scope of differential privacy, we extend the capabilities of private methods. In Chapter 7, we show how classifier evaluations are themselves vulnerable, and show how two common methods can be made to satisfy differential privacy. Lastly, we extend the exponential mechanism, an important tool for building differentially private systems, to cases where the number of possibilities is large.

The lessons from these chapters motivate the future work, to further bridge the gap between the ideal of protecting patient privacy and the goal of improving patient care. As the amount of data stored in EHRs grows, there will be simultaneous pressures to increase the utility of this data and to protect the public from its misuse. This dissertation presents some technical methods to help strike the right balance of care.

REFERENCES

Anderson, Jeffrey L., Benjamin D. Horne, Scott M. Stevens, Amanda S. Grove, Stephanie Barton, Zachery P. Nicholas, Samera F.S. Kahn, Heidi T. May, Kent M. Samuelson, Joseph B. Muhlestein, John F. Carlquist, and for the Couma-Gen Investigators. 2007. Randomized trial of genotype-guided versus standard warfarin dosing in patients initiating oral anticoagulation. *Circulation* 116(22):2563–2570.

Arthur, David, and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete Algorithms*, 1027–1035. Society for Industrial and Applied Mathematics.

Arya, Vijay, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. 2004. Local search heuristics for K-median and facility location problems. *SIAM Journal on Computing* 33(3): 544–562.

Bache, K., and M. Lichman. 2013. UCI machine learning repository.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*, 238–247.

Barrett, Tyler W, Amy R Martin, Alan B Storrow, Cathy A Jenkins, Frank E Harrell, Stephan Russ, Dan M Roden, and Dawood Darbar. 2011. A clinical prediction model to estimate risk for 30-day adverse events in emergency department patients with symptomatic atrial fibrillation. *Annals of Emergency Medicine* 57(1):1–12.

Blocki, Jeremiah, Avrim Blum, Anupam Datta, and Or Sheffet. 2013. Differentially private data analysis of social networks via restricted sen-

sitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 87–96.

Blum, Avrim, Katrina Ligett, and Aaron Roth. 2013. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)* 60(2):12.

Bonate, Peter L. 2000. Clinical trial simulation in drug development. *Pharmaceutical Research* 17(3):252–256.

Boyd, Kendrick. 2014. Mitigating the risks of thresholdless metrics in machine learning evaluation. Ph.D. thesis, University of Wisconsin-Madison.

Boyd, Kendrick, Vítor Santos Costa, Jesse Davis, and David Page. 2012. Unachievable region in precision-recall space and its effect on empirical evaluation. In *Proceedings of International Conference of Machine Learning*, 639–646.

Boyd, Kendrick, Eric Lantz, and David Page. 2015. Differential privacy for classifier evaluation. In *Proceedings of the 8th ACM workshop on Artificial Intelligence and Security*, 15–23. ACM.

Brace, Larry D. 2001. Current status of the international normalized ratio. *Lab Medicine* 32(7):390–392.

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

Brickell, Justin, and Vitaly Shmatikov. 2008. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceedings of Knowledge and Discovery in Databases*.

Caldwell, Michael D, Tarif Awad, Julie A Johnson, Brian F Gage, Mat Falkowski, Paul Gardina, Jason Hubbard, Yaron Turpaz, Taimour Y Lan-

gaee, Charles Eby, et al. 2008. Cyp4f2 genetic variant alters required warfarin dose. *Blood* 111(8):4106–4112.

Camion, Paul, Claude Carlet, Pascale Charpin, and Nicolas Sendrier. 1992. On correlation-immune functions. In *Proceedings of the eleventh annual International Cryptology Conference on Advances in Cryptology*, 86–100.

Carlquist, John F., Benjamin D. Horne, Joseph B. Muhlestein, Donald L. Lappé, Bryant M. Whiting, Matthew J. Kolek, Jessica L. Clarke, Brent C. James, and Jeffrey L. Anderson. 2006. Genotypes of the Cytochrome P450 Isoform, CYP2C9, and the Vitamin K Epoxide Reductase Complex Subunit 1 conjointly determine stable warfarin dose: a prospective study. *Journal of Thrombosis and Thrombolysis* 22(3).

Chambers, J.M. 1983. *Graphical methods for data analysis*. Chapman & Hall statistics series, Wadsworth International Group.

Chaudhuri, Kamalika, and Daniel J Hsu. 2012. Convergence rates for differentially private statistical estimation. In *Proceedings of the 29th International Conference on Machine Learning*, 1327–1334.

Chaudhuri, Kamalika, and Nina Mishra. 2006. When random sampling preserves privacy. In *Advances in Cryptology*, 198–213. Springer.

Chaudhuri, Kamalika, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *The Journal of Machine Learning Research* 12:1069–1109.

Chaudhuri, Kamalika, and Staal A Vinterbo. 2013. A stability-based validation procedure for differentially private machine learning. In *Proceedings of Neural Information Processing Systems*, 2652–2660.

Chickering, David M., Dan Geiger, and David Heckerman. 1994. Learning Bayesian networks is NP-Hard. Tech. Rep. MSR-TR-94-17, Microsoft Research.

Cline, Thomas W. 1979. A male-specific lethal mutation in Drosophila melanogaster that transforms sex. *Developmental Biology* 72(2):266–275.

Colilla, Susan, Ann Crow, William Petkun, Daniel E Singer, Teresa Simon, and Xianchen Liu. 2013. Estimates of current and future incidence and prevalence of atrial fibrillation in the us adult population. *The American Journal of Cardiology* 112(8):1142–1147.

Consortium, The International Warfarin Pharmacogenetics. 2009. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine* 360:753–764.

Cooper, Gregory F., and Edward Herskovits. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9(4):309–347.

Cormode, Graham. 2011. Personal privacy vs population privacy: learning to attack anonymization. In *Proceedings of Knowledge Discovery in Databases*.

Dalenius, T. 1977. Towards a methodology for statistical disclosure control. *Statistik Tidskrift* 15(429-444):2–1.

Dankar, Fida Kamal, and Khaled El Emam. 2012. The application of differential privacy to health data. In *Proceedings of International Conference on Database Theory*.

Datta, Anupam, Divya Sharma, and Arunesh Sinha. 2012. Provable de-anonymization of large datasets with sparse dimensions. In *Principles of Security and Trust*, 229–248. Springer.

Davis, Jesse, Elizabeth Burnside, Inês Dutra, David Page, and Vïtor Santos Costa. 2005. An integrated approach to learning Bayesian networks of rules. In *Proceedings of the 16th European Conference on Machine Learning*, 84–95.

Davis, Jesse, Elizabeth Burnside, Inês C. Dutra, David Page, Raghu Ramakrishnan, Vítor Santos Costa, and Jude Shavlik. 2007a. Learning a new view of a database: With an application to mammography. In *An Introduction to Statistical Relational Learning*, ed. Lise Getoor and Ben Taskar. MIT Press.

Davis, Jesse, Vïtor Santos Costa, Soumya Ray, and David Page. 2007b. An integrated approached to feature construction and model building for drug activity prediction. In *Proceedings of the 24th International Conference on Machine Learning*.

Davis, Jesse, Eric Lantz, David Page, Jan Struyf, Peggy Peissig, Humberto Vidaillet, and Michael Caldwell. 2008. Machine learning for personalized medicine: Will this drug give me a heart attack? In *Proceedings of Machine Learning in Health Care Applications Workshop. in conjunction with ICML*.

Dawson, Ed, and Chuan-Kun Wu. 1997. Construction of correlation immune Boolean functions. In *Proceedings of the first International Conference on Information and Communication Security*, 170–180.

De Raedt, Luc. 2005. Statistical relational learning: an inductive logic programming perspective. In *Proceedings of the 9th European conference on Principles and Practice of Knowledge Discovery in Databases*.

Dehaspe, Luc, and Luc De Raedt. 1997. Mining association rules in multiple relations. In *In proceedings of the 7th International Workshop on Inductive Logic Programming*.

Dewland, Thomas A, Eric Vittinghoff, Mala C Mandyam, Susan R Heckbert, David S Siscovick, Phyllis K Stein, Bruce M Psaty, Nona Sotoodehnia, John S Gottdiener, and Gregory M Marcus. 2013. Atrial ectopy as a predictor of incident atrial fibrillation: a cohort study. *Annals of Internal Medicine* 159(11):721–728.

Dwork, C., and J. Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the 41st annual ACM Symposium on Theory of Computing*.

Dwork, Cynthia. 2006. Differential privacy. In *Proceedings of the International Colloquium on Automata, Languages and Programming*. Springer.

———. 2011. The promise of differential privacy: A tutorial on algorithmic techniques. In *Focs*.

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, 265–284. Springer.

Dwork, Cynthia, Guy N Rothblum, and Salil Vadhan. 2010. Boosting and differential privacy. In *Proceedings of 51st annual symposium on Foundations of Computer Science*, 51–60. IEEE.

Frankowski, Dan, Dan Cosley, Shilad Sen, Loren Terveen, and John Riedl. 2006. You are what you say: Privacy risks of public mentions. In *Conference on Research and Development in Information Retrieval*.

Fredrikson, Matthew. 2015. Understanding privacy in the era of "Privacy is dead": Inference attacks and new defenses. Ph.D. thesis, University of Wisconsin-Madison.

Fredrikson, Matthew, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *Proceedings of USENIX Security*.

Friedman, Arik, and Assaf Schuster. 2010. Data mining with differential privacy. In *Proceedings of Knowledge Discovery in Databases*, 493–502. ACM.

Friedman, Nir, Dan Geiger, and Moises Goldszmidt. 1997. Bayesian networks classifiers. *Machine Learning* 29:131–163.

Friedman, Nir, Iftach Nachman, and Dana Peér. 1999. Learning Bayesian network structure from massive datasets: The "Sparse Candidate" algorithm. In *Proceedings of the fifteenth international conference on Uncertainty in Artificial Intelligence*, 206–215.

Fusaro, Vincent A., Prasad Patil, Chih-Lin Chi, Charles F. Contant, and Peter J. Tonellato. 2013. A systems approach to designing effective clinical trials using simulations. *Circulation* 127(4):517–526.

Gage, BF, C Eby, JA Johnson, E Deych, MJ Rieder, PM Ridker, PE Milligan, G Grice, P Lenzini, AE Rettie, et al. 2008. Use of pharmacogenetic and clinical factors to predict the therapeutic dose of warfarin. *Clinical Pharmacology & Therapeutics* 84(3):326–331.

Gail, Mitchell H. 2009. Value of adding single-nucleotide polymorphism genotypes to a breast cancer risk model. *Journal of the National Cancer Institute* 101(13):959–963.

Ghosh, Arpita, Tim Roughgarden, and Mukund Sundararajan. 2009. Universally utility-maximizing privacy mechanisms. In *Proceedings of Symposium on Theory of Computing*.

Glurich, Ingrid, James K Burmester, and Michael D Caldwell. 2010. Understanding the pharmacogenetic approach to warfarin dosing. *Heart Failure Reviews* 15(3):239–248.

Graves, Alan, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 6645–6649. IEEE.

Greenlee, Robert T, and Humberto Vidaillet. 2005. Recent progress in the epidemiology of atrial fibrillation. *Current Opinion in Cardiology* 20(1): 7–14.

Gupta, Anupam, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. 2010. Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM Symposium on Discrete Algorithms*, 1106–1125. SIAM.

Hall, Rob, Alessandro Rinaldo, and Larry Wasserman. 2013. Differential privacy for functions and functional data. *The Journal of Machine Learning Research* 14(1):703–727.

Hamberg, A. K., Dahl, M. L., M. Barban, M. G. Srordo, M. Wadelius, V. Pengo, R. Padrini, and E.N. Jonsson. 2007. A PK-PD model for predicting the impact of age, CYP2C9, and VKORC1 genotype on individualization of warfarin therapy. *Clinical Pharmacology Theory* 81(4):529–538.

Hand, DavidJ., and RobertJ. Till. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning* 45(2):171–186.

Hardt, Moritz, Katrina Ligett, and Frank McSherry. 2012. A simple and practical algorithm for differentially private data release. In *Proceedings of Neural Information Processing Systems*, 2339–2347.

Heckerman, David, Dan Geiger, and David M. Chickering. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20(3):197–243.

Holford, N., S. C. Ma, and B. A. Ploeger. 2010. Clinical trial simulation: A review. *Clinical Pharmacology Theory* 88(2):166–182.

Holford, N. H. G., H. C. Kimko, J. P. R. Monteleone, and C. C. Peck. 2000. Simulation of clinical trials. *Annual Review of Pharmacology and Toxicology* 40(1):209–234.

Homer, Nils, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F.

Nelson, and David W. Craig. 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics* 4(8).

Hripcsak, George, and David J Albers. 2013. Next-generation phenotyping of electronic health records. *Journal of the American Medical Informatics Association* 20(1):117–121.

Ingall, Timothy. 2004. Stroke-incidence, mortality, morbidity and risk. *Journal of Internal Medicine* 36:143–152.

Jagannathan, Geetha, Krishnan Pillaipakkamnatt, and Rebecca N. Wright. 2012. A practical differentially private random decision tree classifier. *Transactions on Data Privacy* 5(1):273–295.

Jain, Prateek, Vivek Kulkarni, Abhradeep Thakurta, and Oliver Williams. 2015. To drop or not to drop: Robustness, consistency and differential privacy properties of dropout. *CoRR* abs/1503.02031.

Ji, Zhanglong, and Charles Elkan. 2013. Differential privacy based on importance weighting. *Machine Learning* 93(1):163–183.

Joyner, A. L., A. Liu, and S. Millet. 2000. Otx2, Gbx2 and Fgf8 interact to position and maintain a mid-hindbrain organizer. *Current Opinion in Cell Biology* 12(6):736–741.

Kairouz, Peter, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on Machine Learning*, 1376–1385.

Kamali, Farhad, and Hilary Wynne. 2010. Pharmacogenetics of warfarin. *Annual Review of Medicine* 61(1):63–75.

Karnik, Shreyas, Sin Lam Tan, Bess Berg, Ingrid Glurich, Jinfeng Zhang, Humberto J Vidaillet, C David Page, and Rajesh Chowdhary. 2012. Pre-

dicting atrial fibrillation and flutter using electronic health records. In *Proceedings of Engineering in Medicine and Biology Society*, 5562–5565. IEEE.

Karwa, Vishesh, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. 2011. Private analysis of graph structure. *Proceedings of the VLDB Endowment* 4(11):1146–1157.

Kasiviswanathan, Shiva Prasad, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2013a. Analyzing graphs with node differential privacy. In *Theory of Cryptography*, 457–476. Springer.

Kasiviswanathan, Shiva Prasad, Mark Rudelson, and Adam Smith. 2013b. The power of linear reconstruction attacks. In *Proceedings of the twenty-fourth annual ACM-SIAM Symposium on Discrete Algorithms*, 1415–1433. SIAM.

Kaufman, Leonard, and Peter J Rousseeuw. 1990. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Kearney, Patricia M, Colin Baigent, Jon Godwin, Heather Halls, Jonathan R Emberson, and Carlo Patrono. 2006. Do selective cyclo-oxygenase-2 inhibitors and traditional non-steroidal anti-inflammatory drugs increase the risk of atherothrombosis? Meta-analysis of randomised trials. *BMJ* 332:1302–1308.

Kifer, Daniel, and Ashwin Machanavajjhala. 2011. No free lunch in data privacy. In *Proceedings of the SIGMOD International Conference on Management of Data*.

———. 2014. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions Database Systems* 39(1):3:1–3:36.

Kim, M. J., S. M. Huang, U. A. Meyer, A. Rahman, and L. J. Lesko. 2009. A regulatory science perspective on warfarin therapy: a pharmacogenetic opportunity. *Journal of Clinical Pharmacology* 49:138–146.

Kimmel, Stephen E, Benjamin French, Scott E Kasner, Julie A Johnson, Jeffrey L Anderson, Brian F Gage, Yves D Rosenberg, Charles S Eby, Rosemary A Madigan, Robert B McBane, et al. 2013. A pharmacogenetic versus a clinical algorithm for warfarin dosing. *New England Journal of Medicine* 369(24):2283–2293.

Kovacs, Michael J., Marc Rodger, David R. Anderson, Beverly Morrow, Gertrude Kells, Judy Kovacs, Eleanor Boyle, and Philip S. Wells. 2003. Comparison of 10-mg and 5-mg warfarin initiation nomograms together with low-molecular-weight heparin for outpatient treatment of acute venous thromboembolism. *Annals of Internal Medicine* 138(9):714–719.

Kramer, Daniel B, and Peter J Zimetbaum. 2013. Ectopy and expectations: can we predict atrial fibrillation, and should we try? *Annals of Internal Medicine* 159(11):787–788.

Kuruvilla, Mariamma, and Cheryle Gurk-Turner. 2001. A review of warfarin dosing and monitoring. *Proceedings of the Baylor University Medical Center* 14(3):305–306.

Kuusisto, F., I. Dutra, H. Nassif, Yirong Wu, M.E. Klein, H.B. Neuman, J. Shavlik, and E.S. Burnside. 2013. Using machine learning to identify benign cases with non-definitive biopsy. In *Proceedings of e-Health Networking, Applications Services (Healthcom)*, 283–285.

Landwehr, Niels, Kristian Kersting, and Luc De Raedt. 2005. nFOIL: Integrating Naive Bayes and FOIL. In *Proceeding of the 20th National Conference on Artificial Intelligence*, 795–800.

Landwehr, Niels, Andrea Passerini, Luc De Raedt, and Paolo Frasconi. 2006. kFOIL: Learning simple relational kernels. In *Proceedings of the 21st National Conference on Artificial Intelligence*.

Lantz, Eric, Kendrick Boyd, and David Page. 2015a. Subsampled exponential mechanism: Differential privacy in large output spaces. In *Proceedings of the 8th ACM workshop on Artificial Intelligence and Security*, 25–33. ACM.

Lantz, Eric, Soumya Ray, and David Page. 2007. Learning bayesian network structure from correlation-immune data. In *Proceedings of the twenty-third international conference on Uncertainty in Artificial Intelligence*.

Lantz, Eric, Jeremy Weiss, David Page, John Schmelzer, Richard Berg, Steven Yale, Aaron Miller, and James Burmester. 2015b. Using electronic health records to predict therapeutic warfarin dose. In *AMIA Joint Summits on Translational Science*.

Lavrač, N., and S. Džeroski, eds. 2001. *Relational Data Mining*. Springer.

Lavrač, Nada, and Saso Džeroski. 1992. Inductive learning of relations from noisy examples. In *Inductive Logic Programming*, 495–516.

Lee, Jaewoo, and Chris Clifton. 2011. How much is enough? Choosing ϵ for differential privacy. In *ISC*.

———. 2012. Differential identifiability. In *Proceedings of Knowledge Discovery in Databases*.

Lei, Jing. 2011. Differentially private m-estimators. In *Proceedings of Neural Iinformation Processing Systems*.

Li, Ninghui, Wahbeh Qardaji, and Dong Su. 2012. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, 32–33. ACM.

Lloyd-Jones, Donald M, Thomas J Wang, Eric P Leip, Martin G Larson, Daniel Levy, Ramachandran S Vasan, Ralph B DâŁ™Agostino, Joseph M

Massaro, Alexa Beiser, Philip A Wolf, et al. 2004. Lifetime risk for development of atrial fibrillation the framingham heart study. *Circulation* 110(9):1042–1046.

Loukides, Grigorios, Joshua C. Denny, and Bradley Malin. 2010a. The disclosure of diagnosis codes can breach research participants' privacy. *Journal of the American Medical Informatics Association* 17(3):322–327.

Loukides, Grigorios, Aris Gkoulalas-Divanis, and Bradley Malin. 2010b. Anonymization of electronic medical records for validating genome-wide association studies. *Proceedings of the National Academy of Sciences* 107(17): 7898–7903.

Manning, Christopher D, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Matthews, Gregory J., and Ofer Harel. 2013. An examination of data confidentiality and disclosure issues related to publication of empirical {ROC} curves. *Academic Radiology* 20(7):889 – 896.

McSherry, Frank, and Kunal Talwar. 2007. Mechanism design via differential privacy. In *Proceedings of Foundations of Computer Science*, 94–103. IEEE.

Meads, Catherine, Ikhlaaq Ahmed, and Richard D Riley. 2012. A systematic review of breast cancer incidence risk prediction models with meta-analysis of their performance. *Breast Cancer Research and Treatment* 132(2):365–377.

Menze, Bjoern H, B Michael Kelm, Daniel N Splitthoff, Ullrich Koethe, and Fred A Hamprecht. 2011. On oblique random forests. In *Machine Learning and Knowledge Discovery in Databases*, 453–469. Springer.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, vol. 2, 3.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of Neural Information Processing Systems*, 3111–3119.

Moore, Andrew, and Weng-Keen Wong. 2003. Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning. In *Proceedings of the 20th International Conference on Machine Learning*, 552–559. AAAI Press.

Muggleton, Stephen. 1995. Inverse entailment and progol. *New Generation Computing* 13(3-4):245–286.

Muggleton, Stephen, and Luc de Raedt. 1994. Inductive logic programming: Theory and methods. *Journal of Logic Programming*.

Naccarelli, Gerald V, Helen Varker, Jay Lin, and Kathy L Schulman. 2009. Increasing prevalence of atrial fibrillation and flutter in the united states. *American Journal of Cardiology* 104(11):1534–1539.

Narayanan, Arvind, and Vitaly Shmatikov. 2008. Robust deanonymization of large sparse datasets. In *Proceedings of IEEE Symposium on Security and Privacy*.

———. 2010. Myths and fallacies of personally identifiable information. *Communications of the ACM* 53(6).

National Institutes of Health. 2013. Draft NIH genomic data sharing policy (request for public comments). `https://federalregister.gov/a/2013-22941`.

Neelakantan, Arvind, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient nonparametric estimation of multiple embeddings per word in vector space. In *Proceedings of Empirical Methods in Natural Language Processing*.

Nissim, Kobbi, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of IEEE Symposium on Theory of Computing*, 75. ACM Press.

Norton, S. 1989. Generating better decision trees. In *Proceedings of the eleventh International Joint Conference on Artificial Intelligence*, 800–805.

Page, David, and Soumya Ray. 2003. Skewing: An efficient alternative to lookahead for decision tree induction. In *Proceedings of the seventeenth International Joint Conference on Artificial Intelligence*, 601–607.

Pepe, Margaret Sullivan. 2004. *The statistical evaluation of medical tests for classification and prediction*. Oxford University Press, USA.

Phan, NhatHai, Yue Wang, Xintao Wu, and Dejing Dou. 2016. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Pirmohamed, Munir, Girvan Burnside, Niclas Eriksson, Andrea L Jorgensen, Cheng Hock Toh, Toby Nicholson, Patrick Kesteven, Christina Christersson, Bengt Wahlström, Christina Stafberg, et al. 2013. A randomized trial of genotype-guided dosing of warfarin. *New England Journal of Medicine* 369(24):2294–2303.

Pompe, Uros, and Igor Kononenko. 1995. Naïve Bayesian classifier within ILP-R. In *Proceeding of the 4th international workshop on Inductive Logic Programming*, 417–436.

Provost, Foster J, Tom Fawcett, and Ron Kohavi. 1998. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of International Conference on Machine Learning*, vol. 98, 445–453.

Quan, Hude, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L. Duncan Saunders, Cynthia A. Beck, Thomas E. Feasby, and William A. Ghali. 2005. Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. *Medical Care* 43(11):pp. 1130–1139.

Quinlan, J. Ross, and R. Mike Cameron-Jones. 1995. Oversearching and layered search in empirical learning. In *Proceedings of the fourteenth International Joint Conference on Artificial Intelligence*, 1019–1024.

Quinlan, J.R. 1987. Induction of decision trees. *Machine Learning* 1:81–106.

Ramirez, Andrea H, Yaping Shi, Jonathan S Schildcrout, Jessica T Delaney, Hua Xu, Matthew T Oetjens, Rebecca L Zuvich, Melissa A Basford, Erica Bowton, Min Jiang, et al. 2012. Predicting warfarin dosage in european-americans and african-americans using dna samples linked to an electronic health record. *Pharmacogenomics* 13(4):407–418.

Rubinstein, Benjamin IP, Peter L Bartlett, Ling Huang, and Nina Taft. 2009. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *arXiv:0911.5708*.

Saar-Tsechansky, Maytal, and Foster Provost. 2007. Handling missing values when applying classification models. *Journal of Machine Learning Research* 8:1623–1657.

Sankararaman, Sriram, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. 2009. Genomic privacy and limits of individual detection in a pool. *Nature Genetics* 41(9):965–967.

Schnabel, Renate B, Thor Aspelund, Guo Li, Lisa M Sullivan, Astrid Suchy-Dicey, Tamara B Harris, Michael J Pencina, Ralph B DâŁ™Agostino, Daniel Levy, William B Kannel, et al. 2010. Validation of an atrial fibrillation risk algorithm in whites and african americans. *Archives of Internal Medicine* 170(21):1909–1917.

Schnabel, Renate B, Lisa M Sullivan, Daniel Levy, Michael J Pencina, Joseph M Massaro, Ralph B D'Agostino, Christopher Newton-Cheh, Jennifer F Yamamoto, Jared W Magnani, Thomas M Tadros, et al. 2009. Development of a risk score for atrial fibrillation (Framingham heart study): a community-based cohort study. *The Lancet* 373(9665):739–745.

Schneeweiss, Ronald, Roger A Rosenblatt, Daniel C Cherkin, C Richard Kirkwood, and Gary Hart. 1983. Diagnosis clusters: a new tool for analyzing the content of ambulatory medical care. *Medical Care* 105–122.

Schwarz, Gideon. 1978. Estimating the dimension of a model. *Annals of Statistics* 6(2):461–464.

Sconce, Elizabeth A., Tayyaba I. Khan, Hilary A. Wynne, Peter Avery, Louise Monkhouse, Barry P. King, Peter Wood, Patrick Kesteven, Ann K. Daly, and Farhad Kamali. 2005. The impact of CYP2C9 and VKORC1 genetic polymorphism and patient characteristics upon warfarin dose requirements: proposal for a new dosing regimen. *Blood* 106(7):2329–2333.

Scott, Stuart A, and Steven A Lubitz. 2014. Warfarin pharmacogenetic trials: is there a future for pharmacogenetic-guided dosing? *Pharmacogenomics* 15(6):719.

Shokri, Reza, and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on Computer and Communications Security*, 1310–1321. ACM.

Shwe, M. A., B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, and H. P. Lehmann. 1991. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Methods of Information in Medicine* 30(4):241–255.

Simmons, Daniel, Regina Botting, and Timothy HLA. 2004. Cyclooxygenase isozymes: The biology of prostaglandin synthesis and inhibition. *Pharmacological Reviews* 56:387–437.

Singh, Anima, Girish Nadkarni, John Guttag, and Erwin Bottinger. 2014. Leveraging hierarchy in medical codes for predictive modeling. In *Proceedings of the 5th ACM conference on Bioinformatics, Computational Biology, and Health Informatics*, 96–103. ACM.

Sorensen, Sonja V., Sarah Dewilde, Daniel E. Singer, Samuel Z. Goldhaber, Brigitta U. Monz, and Jonathan M. Plumb. 2009. Cost-effectiveness of warfarin: Trial versus real-world stroke prevention in atrial fibrillation. *American Heart Journal* 157(6):1064 – 1073.

Srinivasan, Ashwin. 2001. *The Aleph Manual*.

Stoddard, Ben, Yan Chen, and Ashwin Machanavajjhala. 2014. Differentially private algorithms for empirical machine learning. *arXiv:1411.5428*.

Sweeney, Latanya. 2000. Simple demographics often identify people uniquely. Working paper.

———. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05):557–570.

Takeuchi, Fumihiko, Ralph McGinnis, Stephane Bourgeois, Chris Barnes, Niclas Eriksson, Nicole Soranzo, Pamela Whittaker, Venkatesh Ranganath, Vasudev Kumanduri, William McLaren, Lennart Holm, Jonatan Lindh, Anders Rane, Mia Wadelius, and Panos Deloukas. 2009. A genome-wide association study confirms VKORC1, CYP2C9, and CYP4F2 as principal genetic determinants of warfarin dose. *PLoS Genetics* 5(3).

Tran, Truyen, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. 2015. Learning vector representation of medical objects via emr-driven nonnegative restricted boltzmann machines (enrbm). *Journal of Biomedical Informatics* 54(0):96 – 105.

Ullman, Jonathan. 2012. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. *CoRR* abs/1207.6945.

Valiant, L. G. 1984. A theory of the learnable. *Communications of the ACM*.

Vidaillet, Humberto, Juan F Granada, Po-Huang Chyou, Karen Maassen, Mario Ortiz, Juan N Pulido, Param Sharma, Peter N Smith, and John Hayes. 2002. A population-based study of mortality among patients with atrial fibrillation or flutter. *American Journal of Medicine* 113(5):365–370.

Vinterbo, Staal. 2012. Differentially private projected histograms: Construction and use for prediction. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery*.

Vu, Duy, and Aleksandra Slavkovic. 2009. Differential privacy for clinical trial data: Preliminary evaluations. In *Proceedings of International Conference on Data Mining workshops*.

Wang, Rui, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. 2009. Learning your identity and disease from research papers: information leaks in genome wide association studies. In *Proceedings of the conference on Computer and Communications Security*.

Wasserman, Larry, and Shuheng Zhou. 2010. A statistical framework for differential privacy. *Journal of the American Statistical Association* 105(489): 375–389.

Weiss, Jeremy C, Sriraam Natarajan, Peggy L Peissig, Catherine A McCarty, and David Page. 2012. Statistical relational learning to predict primary myocardial infarction from electronic health records. In *Proceedings of Innovative Applications of Artificial Intelligence*.

Williams, Oliver, and Frank McSherry. 2010. Probabilistic Inference and Differential Privacy. In *Proceedings of Neural Information Processing Systems*.

Witten, Ian H., and Eibe Frank. 2005. *Data mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

Wu, Jionglin, Jason Roy, and Walter F Stewart. 2010. Prediction modeling using ehr data: challenges, strategies, and a comparison of machine learning approaches. *Medical Care* 48(6):S106–S113.

Xiao, Xiaokui, Guozhang Wang, and Johannes Gehrke. 2011. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering* 23(8):1200–1214.

Zeng, Chen. 2013. On differentially private mechanisms for count-range queries and their applications. Ph.D. thesis, University of Wisconsin-Madison.

Zeng, Chen, Eric Lantz, Jeffrey F. Naughton, and David Page. 2014. On differentially private inductive logic programming. In *Proceedings of international conference on Inductive Logic Programming*, 18–30. Springer Berlin Heidelberg.

Zeng, Chen, Jeffrey F. Naughton, and Jin-Yi Cai. 2012. On differentially private frequent itemset mining. *Proc. VLDB Endow.*

Zhang, Jun, Xiaokui Xiao, Yin Yang, Zhenjie Zhang, and Marianne Winslett. 2013. Privgene: differentially private model fitting using genetic algorithms. In *Proceedings of international conference on Management of Data*, 665–676. ACM.

Zhang, Jun, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. 2012. Functional mechanism: regression analysis under differential privacy. In *Proceedings of international conference on Very Large Data Bases*.