Interpretable Machine Learning Methods for Stroke Prediction

by

Rebecca Zhang

B.S.E. Operations Research and Financial Engineering, Princeton University (2015)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2019

© Massachusetts Institute of Technology 2019. All rights reserved.

Author . . . .
Sloan School of Management
August 8, 2019

Certified by . . . .
Prof. Dimitris Bertsimas
Boeing Professor of Operations Research
Thesis Supervisor

Accepted by . . . .
Prof. Georgia Perakis
William F. Pounds Professor of Operations Research
Codirector, Operations Research Center

THIS PAGE INTENTIONALLY LEFT BLANK

Interpretable Machine Learning Methods for Stroke Prediction

by

Rebecca Zhang

Submitted to the Sloan School of Management
on August 8th, 2019 in partial fulfillment of the
requirements for the degree of
Master of Science in Operations Research

Abstract

Machine learning has long been touted as the next big tool, revolutionizing scientific endeavors as well as impacting industries like retail and finance. Naturally, there is much interest in the potential of next improving healthcare. However, using traditional machine learning approaches in this domain has many difficulties, chief among which is the issue of interpretability. We focus on the medical condition of stroke, a particularly desirable problem to target because it is one of the most prevalent and yet preventable conditions affecting Americans today.

In this thesis, we apply novel interpretable prediction techniques to the problem of predicting stroke presence, location, acuity, and mortality risk for patient populations at two different hospital systems. We show that using an interpretable, optimal tree-based approach is roughly as effective if not better than black-box approaches. Using the clinical learnings from these studies, we explore new interpretable methodologies designed with medical applications and their unique challenges in mind. We present a novel regression algorithm to predict outcomes when the population is comprised of notably different subpopulations, and demonstrate that this gives comparable performance with improved interpretability. Finally, we explore new natural language processing techniques for machine learning from text. We propose an alternate end-to-end framework for going from unprocessed textual data to predictions, with an interpretable linguistics-based approach to model words. Altogether, this work demonstrates the promise that new parsimonious, interpretable algorithms have in the domain of stroke and broader healthcare problems.

Thesis Supervisor: Prof. Dimitris Bertsimas
Boeing Professor of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

I would like to thank all the people in my life who have helped make this experience possible.

First and foremost, my greatest thanks go to my advisor, Professor Dimitris Bertsimas. I chose to return to school to conduct research with the singular goal of improving the way decision-making is done in healthcare and medicine. From our very first meeting, it was clear that I had met a future mentor whose interests, ambitions, and spirit were a perfect match to shape me into the researcher I'd wanted to become. Thank you for your constant unwavering support, for the generous academic and life advice, and for always having the confidence and belief in me to aim a little higher every day.

I would like to thank my many wonderful collaborators in these various projects whose collaboration improved both the quality and enjoyment of my work. These include ORC affiliates, Jack Dunn, Lea Kapelevich, Agni Orfanoudaki, Francois Caprasse, and Katerina Giannoutsou, who showed me that research is truly a team process. I am also immensely grateful to my medical collaborators, whose expertise in the field formed the foundation of these projects. Charlene, Meghan, Amre: your knowledge of the data and input on how to define the tasks were invaluable. Thank you for your guidance and for being a constant reminder of the work involved in putting the patient first.

I am incredibly grateful to the National Science Foundation's Graduate Research Fellowships Program for providing support, both in the form of financial grants and an active community, that made this work possible. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1122374. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

Finally, I want to thank my family and friends without whom I would not be here and this work could not have been done. To my parents and brother, thank you for all your constant support throughout all of my years in school and during this latest return. To Sub, I will be forever grateful to have had you by my side during these years. Thank you for being my rock. And lastly, to my wonderful fellow graduate students at the ORC, I could not have asked for a better group of people to have experienced MIT with. From our retreats to viewing parties to project deadlines, I will keep many treasured memories with me even as I move to the next phase in my life and career.

Contents

Chapter 1

Introduction

The cost of healthcare in the United States is constantly rising, by latest estimates it is today over three trillion dollars a year. Even though as a country we spend more each year, medical errors are widespread and conditions go undiagnosed or mismanaged. The use of machine learning in this context is hardly new - the use of computing technologies for healthcare settings extends at least as far back as the 1970s, when a group from Stanford developed a system identifying infections caused by bacteria [66]. Ever since then, new approaches have constantly been published in the academic literature, and yet real-world success stories of using machine learning in the clinical setting to diagnose or treat patients has been few and far between. This is likely due to the unique challenges healthcare problems pose to machine learning algorithms. Sufficient amounts of training data could be difficult to obtain, causing algorithms that worked well in a theoretical or limited setting to generalize poorly or preventing their development in the first place. Even strong algorithms developed on immense amounts of data were not adopted as they are difficult to fit well into the workflow, not least because these algorithms were often black-box techniques that clinicians could not easily interpret. Without easily interpretable algorithms, using machine learning to predict outcomes for patients is risky, since test populations differ from trial or training populations and change over time, and models can overfit to noise rather than real medical factors.

In the last decade or so, there has been a resurgence in using machine learning in the medical community. Some factors contributing to this include the widespread use of electronic health records in the United States, which increased from 20 percent in 2004 to nearly 90 percent in 2017 [59], as well as publicly available de-identified data released to non-clinical researchers. There have also been major breakthroughs in machine learning and revived interest from the medical community. The one constant from before this period that is still true now is the desperate need for interpretability in deployed algorithms. Indeed, there has been increased focus given to the subject of interpretability in machine learning, with conferences and publications dedicated to the subject. In this thesis, we demonstrate the need and the potential for interpretable predictive algorithms in the specific context of stroke. As one of the country's leading causes of death as well as drivers of healthcare costs [26], stroke is also one of the more preventable and manageable medical conditions in the population.

1.1 Contributions

In Chapter 2, we describe an application of an interpretable machine learning technique, Optimal Classification Trees (OCT), on predicting stroke outcomes, both mortality and recurrence, of patients at Hartford Healthcare using available structured data. Collaborating with chief neurologists at the institution, we present decision trees for calculating stroke risk trained on the hospital system's own patient population and verified by practicing clinicians. This was joint work done with Katerina Giannoutsou.

In Chapter 3, motivated by patterns observed in our actual patient populations, we introduce a novel method for regression problems in the context of medical outcomes. Called Sparse Regression over Clusters (SparClur), the model trains a set of coordinated linear regressors that share the same support. The coordination is done over leaves generated by an optimal tree on a given patient population. We show this technique recovers the true support on synthetic data, and performs well on empirical data with a low cost to accuracy. This was joint work done with Lea Kapelevich.

In Chapter 4, we shift from using structured phenotypic data to analyzing text data. We predict the presence, location, and acuity of ischemic stroke from radiology reports at Partners Healthcare. We again employ OCTs, as well as a suite of other classification techniques including neural networks, and demonstrate that simpler, more interpretable methods can perform on par with black-box approaches. We also contribute a set of GloVe word embeddings trained for the specific context of parsing radiology report texts. This was joint work done with Agni Orfanoudaki and Francois Caprasse.

Finally, in Chapter 5, we introduce a set of optimization-based approaches to natural language processing problems. These techniques are designed with interpretability in mind. We demonstrate that these new word embeddings and document representations result in higher predictive performance in downstream machine learning tasks.

1.2 Summary

In this thesis, we present a collection of works using interpretable machine learning approaches to predict characteristics or outcomes of stroke at various hospitals. At Hartford HealthCare, we illustrate the ability of Optimal Classification Trees to predict in-hospital mortality, mortality within a year from discharge, and recurrence of stroke. We show that our method not only has predictive performance similar to or better than other machine learning techniques and risk scores, but that the resulting trees are highly interpretable and verify neurologists' clinical understanding of stroke. Motivated by this experience, we devise a similar approach for regression problems, where a real-valued variable of interest depends on the same set of features, but the degree of dependence can differ for various subpopulations. We present SparClur as a method that achieves this generalizability on the feature set while still ensuring state of the art accuracy and retrieval of the correct support. We next shift our focus to unstructured text data, extremely prevalent in healthcare, and give a comprehensive overview of how a combination of popular machine learning classifiers combined with natural language processing techniques perform for predicting presence, location, and acuity of ischemic stroke in patients at Partners

HealthCare from the raw text of radiology reports. Our findings suggest that in this domain, simple, interpretable methods perform comparably and may be preferable to black-box methods. Finally, we conclude by presenting initial work in developing interpretable natural language processing techniques. This includes language-based approaches for word sense disambiguation, word representation, and classification. We introduce WordNet and show promising preliminary results indicating that optimization-based approaches to these problems may be more powerful than traditional approaches on both regular and medical text. As a whole, this thesis begins grounded in real-world problems faced by actual clinicians concerned with understandable yet sophisticated methods that can capture the complexities and non-linearities of stroke in the real world. From there, we continue by demonstrating the importance and the potential of interpretable machine learning approaches, and present new methodologies with the objective of improving accuracy and interpretability together.

Chapter 2

Predicting Stroke Outcomes at HHC

2.1 Introduction

Stroke is one of the leading causes of death and one of the most prominent drivers of health costs in the United States [26]. The readmission rates within 30 days or within one year are causing a significant increase in healthcare costs [50] making the reduction of hospital readmission rates after a stroke a top priority among the hospitals and the Centers for Medicare and Medicaid Services (CMS). In order to be able to counsel patients and families, allocate resources in an optimal way and ultimately, reduce the recurrence and mortality rates following a stroke, it is essential to understand and assess the mortality and readmission risk of the patients. Although some risk factors for stroke recurrence are known, the mortality risk remains unclear and studies have been conducted in order to find the etiologies and best predictors of mortality risk [58]. Since there are numerous factors that can cause a wide variety of complications to the patient, a better understanding of them, particularly those not yet well-established, are key to reducing readmission and mortality following a stroke. Over the past four decades, several risk scores have been introduced to identify individuals at high risk for cerebrovascular disease [71, 54, 19]. These approaches apply traditional statistical tools such as the Cox Proportional Hazards model [25], which assumes a linear relationship between the risk factors and the prevalence of stroke. While useful, they assume that the variables in their models interact in a linear and additive fashion. The mathematical and medical realities, however, suggest that the interaction of risk factors and markers of disease acuity are far from linear, and that some variables gain or lose significance due to the absence or presence of other variables [20, 47].

Take, for example, three variables which have been repeatedly found to be independent predictors of stroke: age more than 75 years, high cholesterol, and atrial fibrillation. In existing, linear predictive models, each of these variables is treated as “present” or “absent”, and often assigned the same weight irrespective of the presence or absence of the other two risk factors. However, it is theoretically possible that, for patients older than 75 years, high cholesterol plays a role but atrial fibrillation does not; whereas in patients younger than 75 years, high cholesterol does not play a significant role but atrial fibrillation does. Therefore, in a non-linear risk model, the age of the patient would determine whether high cholesterol or atrial fibrillation would be included in the prediction of outcomes. The inclusion of one of these two would then determine the next variable to be included, and this variable could be different for each of the two choices.

For example, if atrial fibrillation was chosen, then presence of cardiovascular disease could be the next variable added; if high cholesterol was chosen, then diabetes could be added. As a result, in a linear model the stroke risk of these two observations would be established based on the presence or absence of the same set of variables, while in a non-linear model, the risk could be determined by two very different sets of variables. The latter arguably better represents the complexity, interactivity, and non-linearity of real life.

2.1.1 Existing Methodologies

As described above, currently the most popular techniques used to evaluate mortality and recurrence risk operate on the assumption of linearity in the factors. [2] conducted a widely cited study predicting mortality in-hospital on the Lausanne Stroke Registry of 3362 patients. Using a multivariate logistic analysis, the authors found that impaired consciousness and weak limbs along with the presence of various past health events were predictors of mortality for brain infarction. Likewise, impaired consciousness and weak limbs were good predictors for mortality for brain hemorrhage. They noted that age did not appear as a predictor in their models, but the average age of patients who exhibited all of the risk factors in the predictive model increased as the number of risk factors increased, a clear indication of a non-linear interaction between these features.

Predicting mortality within a year has also been treated linearly in the past. [70] developed a technique based on the Cox proportional hazards model for the mortality within a year of 440 patients hospitalized for acute ischemic stroke. The final model used eight clinical predictors, each assigned varying integer weights between two and nine, and patients whose total score exceeded 10 were assigned to a high-risk group. The high-risk group had a mortality rate of 76% compared to 8% in the low risk group. Aside from the linear treatment of the predictors, the model has the additional shortcoming of assigning only two drastically differing rates to all patients.

2.1.2 Contributions

Considering the challenge of both identifying risk factors and understanding their non-linear impact on a patient's total risk of mortality and recurrence from stroke, we sought, in this paper, to create a non-linear, highly accurate, and user-friendly mortality and recurrence risk calculator for patients who experience a stroke. Our predictive algorithm is a tree-based method called Optimal Classification Trees (OCT) that allows the physician to explore the exact model and assess the interpretability of its results [6]. We implement multiple machine learning classifiers and use medical risk scores to predict the outcomes of in-hospital mortality, mortality within a year, and recurrence within a year on a large-scale patient population at Hartford Health Care (HHC). Our results demonstrate that the three tasks are of varying complexity, with recurrence in a year being a low-signal problem. Our algorithm outperforms others in the task of predicting mortality within a year, and is close to optimal in performance of in-hospital mortality while being far more interpretable than the most accurate model. The resulting trees are validated by neurologists and match clinical intuition and understanding of stroke in our patient population.

2.2 Methods

In this section, we describe in detail the formal methodology used to define, formulate, and solve the problem of predicting stroke mortality and recurrence at HHC.

2.2.1 Data

Three data tables, comprising basic patient information, medication information, and core data, were made available. These included all 11,665 visits of 10,543 unique patients hospitalized for an ischemic stroke, hemorrhagic stroke, or transient ischemic attack (TIA) between January 1st 2005 and June 29th 2016 at HHC. The medications table lists the medications prescribed, both at admission (home med) and at discharge. These include over 400 specific medications, which to incorporate into our model, we mapped to six medication types: anticoagulant, antihypertensive, antiplatelet, diabetes, statins, and other medications. Each visit then included 12 binary variables, marking whether a medication of that type was taken at admission as well as whether it was prescribed at discharge.

Feature Variables

The variables used to design our predictive models are collected on each visit of a patient to HHC and include basic demographic information and medications prescribed (Table 2.1), as well as Medical Data/Stroke Risk Factors. We also had available the Risk Scores that are assigned to each patient and the complications that might arise during the hospital stay, but these are usually known only close to discharge. As a result, we do not include them in the predictive model for mortality inside the hospital, but do include them in the mortality within one year and stroke recurrence within one year models.

Demographic Data	%	Medication	%
Female	51.26	Anticoagulant, HomeMed	12.92
Age, Mean (SE)	69.1 (0.15)	Antihypertensive, HomeMed	60.51
Height, Mean (SE)	87.64 (10.35)	Antiplatelet, HomeMed	40.94
Weight, Mean (SE)	177.52 (1.39)	Diabetes, HomeMed	15.19
Race/White	77.3	Other, HomeMed	6.11
Race/Black or African American	8.79	Anticoagulant, Discharge	26.84
Hispanic Ethnicity	9.46	Antihypertensive, Discharge	61.49
Marital Status/Married	45.68	Antiplatelet, Discharge	58.41
Admitted for Ischemic Stroke	62.55	Diabetes, Discharge	16.97
Admitted for Hemorrhagic Stroke	23.86	Other, Discharge	10.86

Table 2.1: Breakdown of basic patient and medication information.

Missing Data Imputation

Missing data are a prevailing problem in any type of data analysis. A participant variable is considered missing if the value of the variable for the participant is not observed. The most common type is missing response and/or covariate data for covariates with either discrete or continuous values [41]. In most analyses appearing in the medical literature, the most common way of dealing with missing data is to just omit those participants who have any missing data. Another very common approach in the literature is to just replace the missing value with the most common value (discrete case) of the covariate or the mean (continuous case).

Due to the relative rarity of observing mortality or stroke recurrence, it was a priority for us to retain as many observations as we could. Therefore, we chose not to omit observations that had any missing values. Instead, we dropped variables that were rarely observed - specifically, those that were missing in more than 90% of the entries. For the remaining variables, we imputed missing values using a recently developed and novel machine-learning method called OptImpute [12], which formulates the . Imputing the missing values in this way before building predictive models has been shown in multiple real-world datasets to lead to significant improvements in prediction accuracy compared to classical missing values imputation methods.

2.2.2 Task and Cohort Definition

In this section, we describe our problem definition. The task of predicting outcomes in an interpretable way is by no means straightforward, especially due to the problem of data censoring. That is, since many patients in HHC are at risk of mortality due to other comorbidities and most tend to be elderly, the data may show that they didn't have an additional stroke, even if they would have suffered a recurrent stroke or TIA had they survived. In addition, the usual problem in healthcare data of patients switching providers is present here. Those patients who go on to have a recurrent stroke outside of HHC will not be captured in the available data, which only has information surrounding the patient hospitalisation.

Given this, we separate our goal into three distinct tasks: predicting mortality of the patient during the hospital stay, predicting mortality within a year of patients discharged from the hospital, and predicting the recurrence of stroke within a year in patients who survive for at least a year from discharge. A patient will display different features (age, past medications, history of stroke, etc) at each distinct visit. Therefore, we treat each hospitalization as its own observation, regardless of whether that patient has presented in the data before.

We define our cohorts and their outcomes as follows:

1. Patients who died while in-hospital: defined as those visits with a discharge type labeled as "death", or where discharge type is missing but the date of death is before or the same as the date of discharge. Any visit that was missing a discharge date but has a date of death was treated as if the discharge date was the date of death. This comprises 1404 out of a total of 11665 visits (a 12% mortality rate).

2. 10261 visits then remained where the patient was discharged alive from the hospital. We find those patients whose mortality outcomes are known within a year. Since the last date of death available in our dataset is July 4th, 2016, any patient who was discharged after July 4th, 2015 will have an unknown outcome for mortality within the year. For example, some patients were admitted and discharged January of 2016, and since the data collection period ended only six months later, whether they survived the year is unknown and should not be marked as not having died. To keep our dataset clean, then, we consider only hospital visits with a discharge date of July 4th, 2015 or earlier. This leaves us a total

of 9066 observations. In addition, note that since we use as observations all patient visits rather than the individual patients themselves. Therefore, if a patient dies, it is possible for multiple of their previous visits to result in mortality within a year if they were close in time. For example, if a patient is hospitalized for stroke on day 1 and day 30 and later dies on day 300, both visits on day 1 and day 30 would be labeled with mortality within a year as true. This gives us a dataset of 555 out of 9066 visits resulting in mortality within a year (a rate of approximately 6%).

3. Finally, we remove all visits where the patients died in-hospital and all visits where patients died within a year of discharge. Once again, we remove visits where the outcome of recurrence in a year is unknown. Visit data was collected up to June 30th, 2016, so we therefore keep only visits where the patient was discharged on or before June 30th, 2015. This also takes into account those visits where mortality within a year is uncertain, since the cutoff date of June 30th is before the cutoff date of July 4th we used for the mortality outcome, and therefore we know these patients' mortality within a year status with certainty. Out of those remaining, we found those visits that resulted in patients returning to the hospital for ischemic or hemorrhagic stroke (not TIA) within a year. Those visits were marked as having a recurrence. This gave us a dataset of 379 recurrences out of 8504 visits (a rate of approximately 4.5%).

As an illustration of this process, we give an example in Table 2.2. Patient A is still alive at the end of the data collection period, so both mortality outcomes at the two visits are marked with a 0. Their second hospitalization is for a TIA, not ischemic or hemorrhagic stroke, so even though that admission date is within a year of the last discharge date, we do not count this as a recurrence, so the first visit's outcome for recurrence within a year is also 0. Finally, since they have not had a third hospitalization with CVA type stroke, the second visit's outcome for recurrence is again 0.

Patient B has a slightly more interesting history at HHC. Over a year passes between the discharge of their first visit and the admission of their second visit, and additionally the second hospitalization is for a TIA, so the recurrence outcome for the first visit is 0. Their third visit occurs soon after their second, and is of a stroke type, so the recurrence outcome for the second visit is 1. Finally, the third visits results in death in hospital, so that field in the third visit and mortality within a year for the second visit are both marked as 1.

Pt	CVA	Admit Date	Discharge Date	Death Date	Mort Inhosp?	Mort Yr?	Recur Yr?
A	Stroke	1	10	-	0	0	0
A	TIA	370	372	-	0	0	0
B	Stroke	1	3	418	0	0	0
B	TIA	410	415	418	0	1	1
B	Stroke	417	418	418	1	1	0

Table 2.2: An illustration of the cohort definition and outcome labeling process.

2.2.3 Optimal Classification Trees

Our predictive algorithm is a tree-based method called Optimal Classification Trees (OCT) that allows the physician to explore the exact model and assess the interpretability of its results.

Compared with other ML methods such as Neural Networks that are not interpretable [51], OCT is comprehensible and can be easily visualized in a tree form. The final model optimally estimates the probability of the event in each one of the leaves of the tree.

Classical decision tree methods typically cannot achieve the same level of accuracy as deep machine learning methods. Moreover, the early AI machine learning trees often suffered from limited interpretability. Our novel OCT methodology is a recent advancement in machine learning classification that trains a single decision tree, permitting high-accuracy predictions without sacrificing interpretability [6]. This high level of accuracy is achieved by leveraging modern optimization techniques to train decision trees from the perspective of global optimality rather than using greedy heuristics like the classical methods.

Through OCT, we produced a set of predictive models for inhospital mortality, mortality within one year and recurrence of stroke within one year. We trained a separate decision tree for each of the above outcomes. To illustrate how an OCT works, a path from a decision tree that estimates recurrence of stroke within one year is displayed in Figure 2.1. This tree is built on a random subset rather than the entirety of our data. The root node of the tree shows that the overall risk of stroke recurrence is 4.6%. The first decision tree split refers to history of stroke. If the patient has history of prior stroke events, then we proceed to the right branch where the updated risk of stroke recurrence is now higher at 6.84%. The tree then proceeds to split on the Weight variable, where we see that if the weight of the patient is less than 208.5 pounds, the risk of a recurring stroke is 6%. If, however, the patient weighs more than 208.5 pounds, the outcome depends on whether Antiplatelet medication was taken at home, before hospital admission. If the answer to this question is positive, the final risk estimation for that patient is approximately 17% while if it is negative, meaning that the patient was not on Antiplatelets before arriving at the hospital, then the algorithm predicts no risk for this patient. It is important to notice in this setting that after each new split the risk is re-calculated and the pre-operative variables used by the tree are not the same at each level; the questions asked change based on the responses at the prior node capturing, in this way, nonlinear interactions between the variables.

2.2.4 Model Performance

OCT has been shown in previous studies [6] to outperform other machine learning techniques, in that it provides the highest degree of interpretability at no or little cost to accuracy. In this study, to demonstrate the power of the OCT methods, we also implemented logistic regression with LASSO and gradient boosting as baselines for comparison. LASSO logistic regression [68] is a technique that models the log-odds of a binary outcome as a sum of a linear combination of a limited number of predictors. While it is somewhat interpretable, logistic regression assumes a linear relationship betweeen the features with the outcome, and will not capture nonlinear interactions that may exist. Gradient boosting is a technique that builds a large number of decision trees and then uses a voting process to arrive at a prediction [33]. While this enables it to capture nonlinear behavior, it is highly non-interpretable.

The OCT algorithm performance and its ability to predict mortality within the hospitalization, mortality within one year and stroke recurrence within one year was measured using

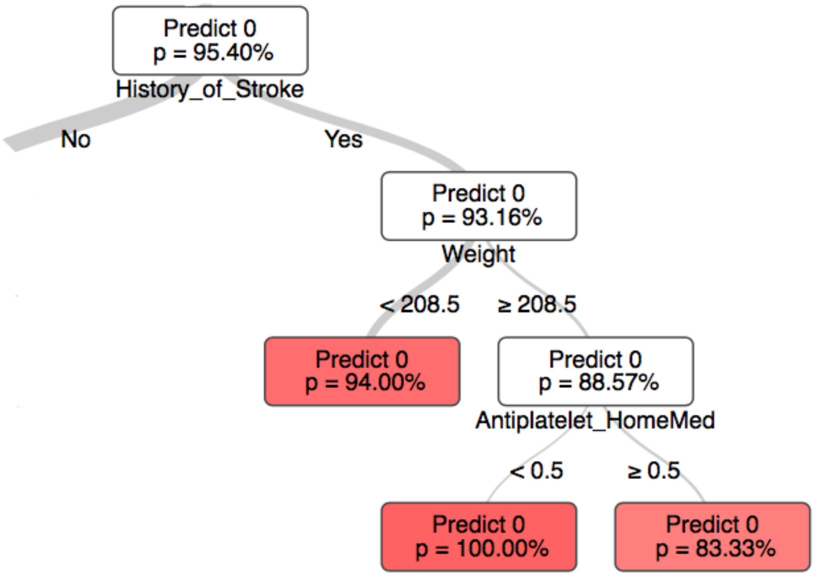


Figure 2.1: Selected path from an OCT for the prediction of recurrence of stroke within one year

Area Under the Curve (AUC), also known in the medical community as the c-statistic. The c-statistic measures the ability of a model to discriminate between the outcomes of interest and has been used as a measure of model success in multiple prior risk-scoring development efforts [15, 37, 65, 46]

The out-of-sample performance of our models was also measured against the performance of the Intracerebral Hemorrhage (ICH) Score [39], pre-morbid modified Rankin Scale (mRS) [31] taken prior to stroke, and mRS at hospital admission. The ICH score ranges from 0 to 6 and measures the severity of an intracerebral hemorrhage, while the mRS is used to measure the level of disability of a patient with a neurological condition and also has a range of 0 to 6. In order to calculate the c-statistic of the risk scores for this comparison, we used each risk score to group patients into five groups. In each group, we calculated the empirical probability of the outcome and used that as the predicted risk for anyone with that score. Although these scores were not necessarily developed to predict our exact outcomes of interest for any given patient, this approach provides a good baseline for the accuracy of the methods that are currently used to counsel patients who experience a stroke and has been used in previous literature comparing machine learning techniques with these types of risk scores [8].

2.3 Results

For each task and corresponding cohort, we developed an optimal tree algorithm to predict the outcome of interest. Table 2.3 compares the performance of our OCT models with those of logistic regression with a LASSO penalty and the XGBoost implementation of boosting [21] in predicting in-hospital mortality, mortality within a year from discharge, and recurrence within a year from discharge. The numbers shown are an average of five c-statistics of the model on out-of-sample test set. We can see that for each task, a different model has the highest performance. For the mortality inhospital task with 11665 visits, boosting has the highest c-statistic of 0.8428, outperforming optimal trees (0.8177) and logistic regression (0.8095). Optimal trees with hyperplanes had the best performance for predicting mortality within a year of discharge (0.7958), with XGBoost a few percent lower (0.7862) and logistic regression significantly lower (0.7309). Finally, for predicting which visits will result in a stroke readmission within a year from discharge, logistic regression had the highest c-statistic (0.5921), outperforming both boosting (0.5736) and optimal trees (0.5447) by a few percentage points.

Model	Mortality inhospital	Mortality in a year	Recurrence in a year
OCT	0.8177	0.7916	0.5384
OCT-H	0.8068	**0.7958**	0.5447
XGBoost	**0.8428**	0.7862	0.5736
Lasso Logistic Regression	0.7862	0.7309	**0.5921**
ICH	0.702	0.6836	0.5380
Pre-morbid mRS	0.604	0.6525	0.5401
Hospital Admission mRS	N/A	0.7368	0.5740

Table 2.3: The performance of Optimal Classification Trees (OCT) and OCTs with hyperplanes (OCT-H) in predicting each of the outcomes, as compared to other machine learning methods and known stroke risk-scores.

Table 2.3 also includes the c-statistics from predictions generated using ICH, pre-morbid mRS, and hospital admission mRS scores as a baseline. As expected, these scores generally perform significantly worse than the machine learning algorithms, given that they were not designed to predict these outcomes explicitly and instead were used as a proxy for risk in our data as described in the previous section.

2.4 Discussion

2.4.1 Contributions

We proposed a novel stroke risk calculator to assess the risk of patients for mortality and recurrence following a stroke, that considers non-linear relationships between the variables utilizing state-of-the-art machine learning methods. Our approach introduces tree-based decision rules where the number of parameters required determine the risk is not fixed. Other than its non-linear aspect, the calculator offers the advantages of being more accurate than the currently existing methods of measuring stroke risk, as well as significantly more interpretable. At the same time, our user-friendly interface renders it very actionable to both physicians and patients while being amenable to integration into existing Electronic Health Records (EHR).

We demonstrate the predictive performance of our OCT models on the tasks of predicting in-hospital mortality, mortality within a year from discharge, and recurrence within a year from discharge. We first note that overall, predicting inhospital mortality appears to be the least difficult task while predicting recurrence within a year is most difficult, with a c-statistic close to 0.5 indicated performance that is barely better than random chance. Given the way we structured our cohorts and the problem definition, this makes sense: while in-hospital, patient information and activity is restricted and the data available more readily captures any factors

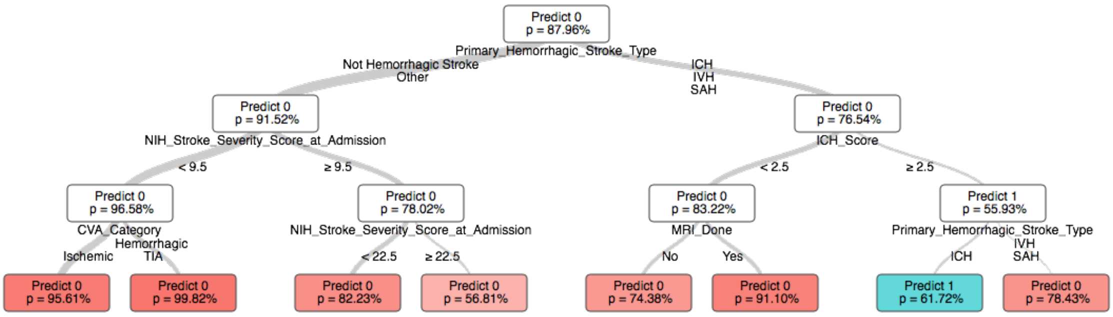


Figure 2.2: Tree to predict mortality risk within the hospital stay

pertaining to their treatment and state that could affect the outcome of interest. On the other hand, once a patient is discharged, the data collected at discharge does not accurately reflect their behavior over the next year. For example, the medications prescribed at discharge will not necessarily be adhered to, resulting in noise in the dataset. The outcomes of mortality within a year and recurrence in a year could also be affected by changes in patient health that occur after discharge, and for mortality in particular, could result from any number of unrelated conditions not captured in the admission data of the current visit. Finally, we will have a number of false negatives in the data (patients marked as not having died or been readmitted within the year) due to the fact individuals can leave the HHC system at any time and would thus be omitted from the data.

We also note that while the three techniques are generally on par with each other, for each task, a different technique appears to be strongest. For predicting inhospital mortality, boosting is the strongest, while for mortality and recurrence within the year, OCT and logistic regression are the best models respectively. This indicates that predicting mortality for a patient who has previously had a stroke is a mostly nonlinear problem, whereas predicting whether a patient will suffer a recurrent stroke is more linear. Given how noisy the data for stroke recurrence is, focus should be given to the task of predicting inhospital mortality and mortality within a year from discharge, in which performance is quite high. In these respects, OCT provides the highest interpretability at the lowest cost to accuracy, and should be the tool of choice as it achieves this balance.

The model we developed for the prediction of mortality within one year of the stroke event is displayed in Figure 2.3. The root node of the tree shows that the overall risk of mortality is around 6%, since the probability of survival is around 94%. The first decision split refers to anticoagulant medication prescribed at discharge. If the patient was put on anticoagulants at the time of discharge then this leads to the right branch of the tree where the tree now splits on the variable “History of Stroke”. If the answer to this question is positive and the patient has experienced a stroke in the past then the algorithm leads to the right with an updated risk of 11.21% . On that node the tree finally splits on the variable regarding the Modified Rankin Scale (MRS) score of the patient, measured at the time of hospital admission. If the score is 4.5 or below then the final risk estimation for the patient is around 8%. If, on the other side, the score is greater than 4.5 then the final risk of mortality is significantly higher, at almost 20%.

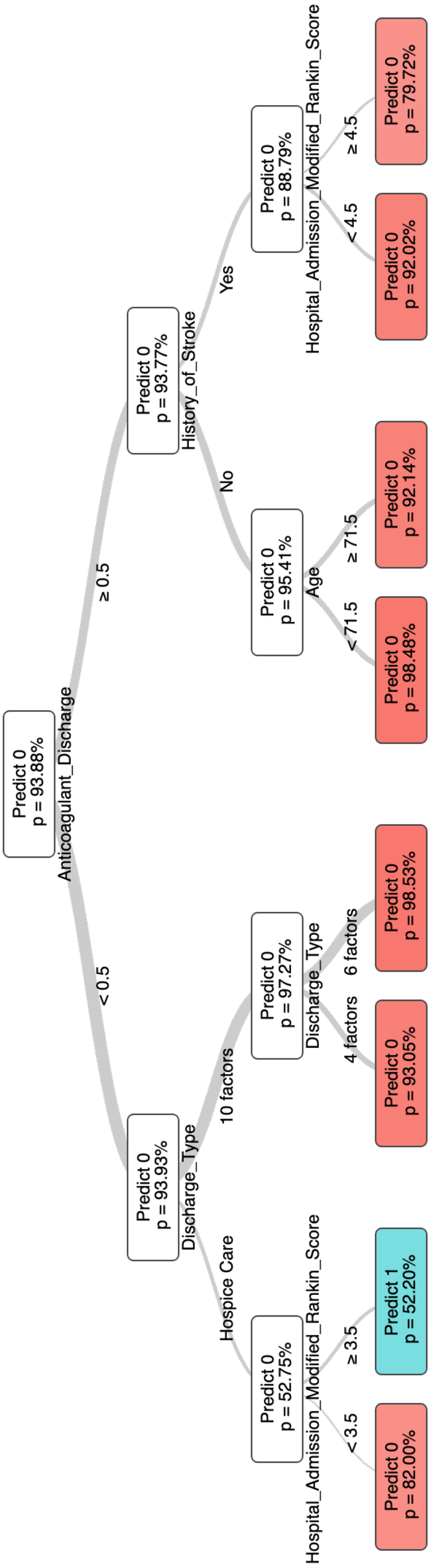
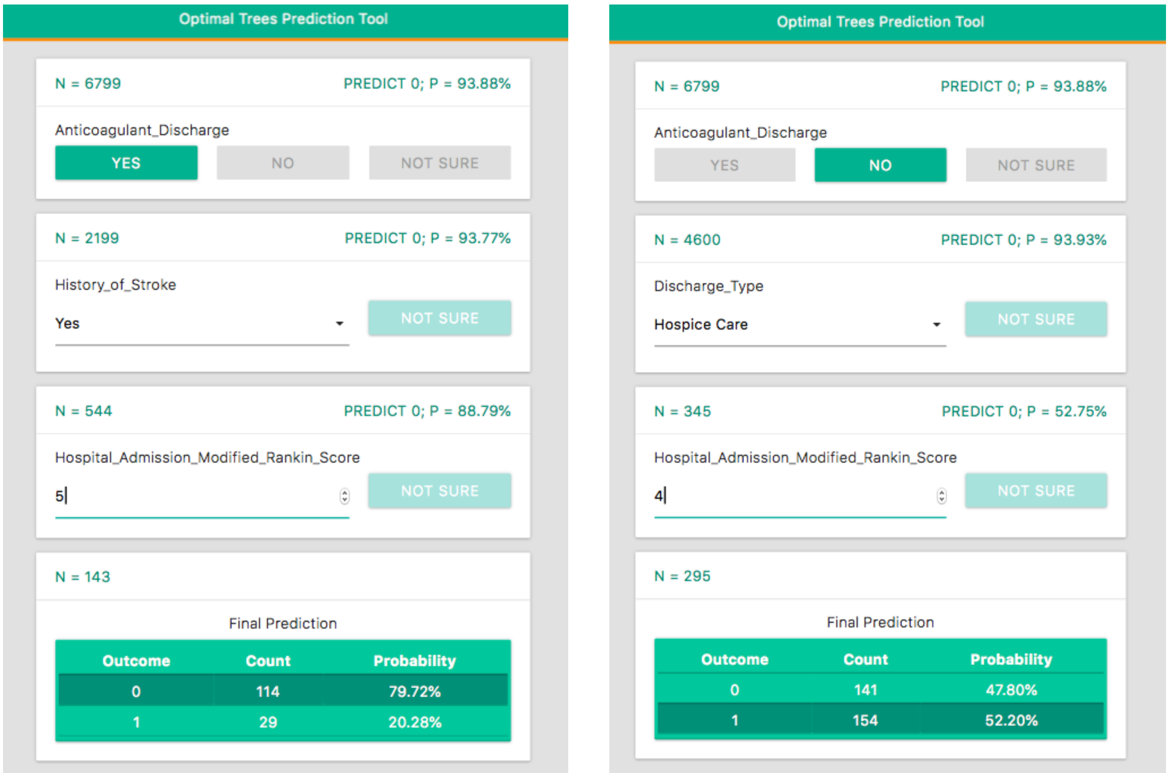


Figure 2.3: Tree to predict mortality risk within one year of stroke

We also see that if we had followed the left path at the node referring to “History of Stroke” then the tree would have next split on a different variable, the age of the patient. Patients aged less than 71.5 are given different risk estimates than those older than 71.5 but still considerably lower than the patients who have history of stroke and a high MRS score. What is important to note here is that the variables used by the tree are not the same at each level; the questions asked change based on the responses at the prior node. In this way, decision trees can capture nonlinear interactions between variables rather than mandate that the variables interact in a linear and additive fashion, as classical logistic regression does.

One of the most important goals of this study is to create a predictive tool that doctors can use on an everyday basis. It is therefore imperative to have an output that is interpretable and carries meaningful insight on the patient’s risk. Consequently, it is of great importance that the specific variables and thresholds on which our tree algorithm splits on make sense from a medical perspective. Based on our results, we believe that we have succeeded in creating a model that combines high accuracy with interpretability. Such a model can be easily deployed in the hospital setting for use by clinicians, using an app on their devices to arrive at an easily explainable and accurate outcome prediction, as illustrated in Figure 2.4.



Optimal Trees Prediction Tool

N = 6799 PREDICT 0; P = 93.88%

Anticoagulant_Discharge

N = 2199 PREDICT 0; P = 93.77%

History_of_Stroke: Yes

N = 544 PREDICT 0; P = 88.79%

Hospital_Admission_Modified_Rankin_Score: 5

N = 143

Final Prediction

Outcome	Count	Probability
0	114	79.72%
1	29	20.28%

Optimal Trees Prediction Tool

N = 6799 PREDICT 0; P = 93.88%

Anticoagulant_Discharge

N = 4600 PREDICT 0; P = 93.93%

Discharge_Type: Hospice Care

N = 345 PREDICT 0; P = 52.75%

Hospital_Admission_Modified_Rankin_Score: 4

N = 295

Final Prediction

Outcome	Count	Probability
0	141	47.80%
1	154	52.20%

Figure 2.4: Screenshots showing a questionnaire that can be used to predict risk of mortality within a year (outcome = 1) for HHC stroke patients. On the left, an example of a patient who has been prescribed anticoagulants at discharge, has a history of stroke, and a modified Rankin score of 5 is predicted to have a 20% risk; on the right, a different outcome for a patient who was not prescribed anticoagulants at discharge, was discharged to hospice, and has a modified Rankin score of 4 is predicted to have a 52% risk. Note the questions asked appear sequentially and depend on the answers of the previous questions. At each step, the sample size used to determine the risk is displayed.

2.4.2 Further Work

We recognize the limitations of this study which can affect the final output of our model. Central to the limitations of our study lies the fact that the power of machine-learning prediction depends on the accuracy and comprehensiveness of the data it uses [20], in this case the Hartford Hospital database. As such, systematic biases resulting from Hartford's data collection methodology, and its changes over the multiple years of data might exist. For the missing values interpretation for example, we had to closely work with with the data collection team of Hartford in order to be able to separate for which features we could safely assume that a missing value refers to a negative answer (e.g a certain complication did not occur) and for which features we could not make this assumption and impute them using our state-of-the-art method. The same goes for the understanding of the medication data that we were given from Hartford as well as the marking of recurrent visits; our body of work is tightly related to the way information is collected and validated in the Hartford system. Furthermore, the fact that our algorithm uses as input data solely from Hartford which is a mainly Caucasian population does not allow for much generalization to other ethnicities. Thus, in order for our results to be generalized, they may need to be refined by retraining our algorithm with data from other longitudinal studies.

Another limitation refers to causality between the variables and the outcomes, which is still not proven despite the high degree of connectivity between the two. Therefore, interpretability and actionability on the relevant variables remains controversial. For example, if the mortality decision tree of a specific patient included a high cholesterol level, correcting it might not necessarily improve the patient's chances of survival. The decision-tree might simply change in a different direction, and ultimately estimate the same mortality risk. Further studies are therefore needed to explore the ability of our calculator to identify actionable "break points" in patient care after a stroke event that can effectively lower their mortality and recurrence risks.

2.5 Conclusion

This study demonstrates the potential of Optimal Classification Tress in the field of medical prediction, and more specifically, in the difficult task of predicting mortality and stroke recurrence for patients who have already experienced a stroke. The models we have developed for each of our tasks have high c-statistics and good higher prediction performance than the already existing stroke risk classification methods. At the same time, they can prove to be valuable tools for more accurately and efficiently identifying individuals at high risk of mortality or stroke recurrence because of their interpretability and therefore ability to demonstrate to the physician and patient the relation and non-linearity of the stroke risk factors. The highly-accurate and user -friendly risk calculator we have developed can therefore appear useful as an evidence-based, adaptive, and interactive tool for stroke patients.

Chapter 3

SparClur: Sparse Regression Over Clusters

3.1 Introduction

In this chapter, we aim to develop machine learning models that combine state of the art accuracy and interpretability. Motivated in particular by applications of medicine, we turn our attention to methods that capture highly nonlinear relationships between features and continuous target variables. Currently, the most popular among these include decision trees (classification and regression trees), random forests, and boosted trees.

Deep learning and ensemble models (including random forests and boosted trees) achieve state of the art accuracy, but are not interpretable. This limits their applicability in areas where understanding the rationale of a model's prediction is important. This is particularly relevant in applications where human experts use machine learning predictions in conjunction with their own knowledge to make decisions.

Sparse regression models [32] and decision trees [18] are machine learning models that aspire to be interpretable and have strong out of sample accuracy. In this paper, we combine ideas from new developments in sparse regression [13] and classification and regression trees [6] to propose a new method that is interpretable, and also provides state of the art accuracy.

To motivate the problem we address, assume we have data $(\mathbf{x}_i, y_i), i = 1, \dots, n$ with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. Let $\mathbf{x}_i$ represent electronic medical records of patient i and y_i represent a medical outcome, for example, a measure of glucose levels of patient i.

Applying Optimal Regression Trees from [29] gives rise to trees similar to that depicted in Figure 3.1.

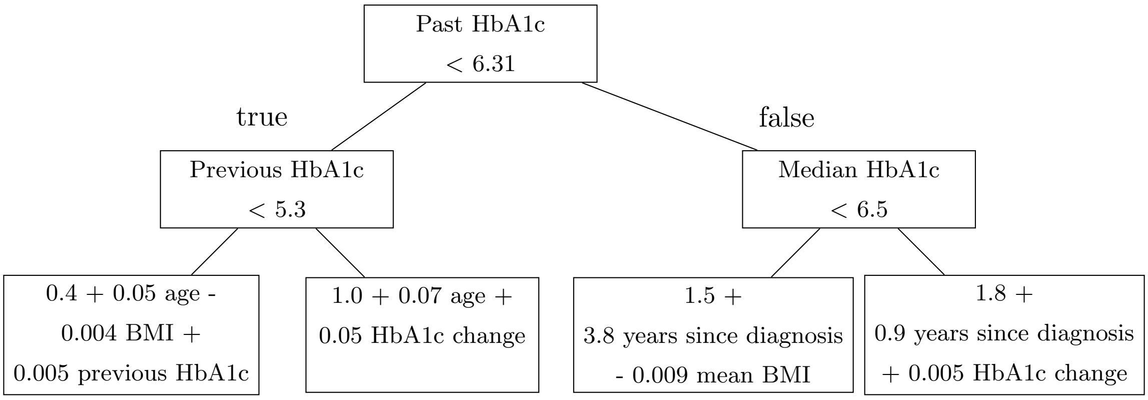


Figure 3.1: Example of an Optimal Regression Tree of depth two for predicting blood glucose levels based on electronic medical records.

In each of the leaves $L_1, \dots, L_4$ the outcome is predicted as a regression involving different variables:

$$\hat{y}_i = w_{0,j} + \mathbf{w}_j^\top \mathbf{x}_i \quad j = 1, \dots, 4. \tag{3.1}$$

In other words, there can potentially be different factors in each leaf affecting the prediction.

Suppose we impose the additional constraint that the support of each vector $\mathbf{w}_j$ is the same for all leaves and in addition the cardinality of this support is limited. That is, $|\text{supp}(\mathbf{w}_j)| \leq q$ for some positive integer q, and $\text{supp}(\mathbf{w}_j) = \text{supp}(\mathbf{w}_k)$, for $j, k = 1, \dots, 4$.

With this criterion, the regression in each leaf is sparse and coordinated among leaves to involve the same variables. This increases the interpretability of the model significantly. Specifically, in the uncoordinated case, it is possible within some leaves for glucose level to be affected by past HbA1c, while not in others, which is medically implausible. More generally, the problem we consider is as follows:

Given data $(\mathbf{x}_i, y_i), i = 1, \dots, n$ and a partition of the data within *clusters* L_j such that $(\mathbf{x}_i, y_i) \in L_j, j = 1, \dots, J$ we want to solve the sparse regression problem over J clusters:

$$\min \ \frac{1}{2\gamma} \sum_{j=1}^{J} ||\mathbf{w}_j||^2 + \sum_{j=1}^{J} \sum_{i \in L_j} (y_i - \mathbf{w}_j^\top \mathbf{x}_i)^2 \tag{3.2}$$

$$\text{s.t.} \quad ||\mathbf{w}_j||_0 \leq q \quad \forall j \tag{3.3}$$

$$\text{supp}(\mathbf{w}_1) = \ldots = \text{supp}(\mathbf{w}_J). \tag{3.4}$$

The term $\frac{1}{2\gamma} \sum_{j=1}^{J} ||\mathbf{w}_j||^2$ is a regularization term that makes the overall model more robust [5]. Note that the Problem given by (3.2) — (3.4) reduces to the sparse regression problem studied in [13] when $J = 1$.

3.1.1 Existing Methodologies

Recently introduced in [29], *Optimal Regression Trees* (ORTs) are a predictive tool similar to CART. ORTs are constructed in a fashion that is optimal for a loss function with respect to a local neighborhood. While granting a higher degree of interpretability than black-box methods such as random forest and boosted trees, ORTs attain comparable performance, in terms of predictive accuracy to these black-box methods, and notably higher predictive accuracy than CART [29].

Linear regression models in tree based methods, including ORTs are currently computed heuristically [17, 29]. These regression models are either point predictions, or in the case of ORTs, linear models built using Lasso. This has several shortcomings. As shown by [10], while Lasso generally performs well at the task of discovering relevant features, it has the property of also selecting a significant number of features that are not part of the true support. This hinders the interpretability of trees because often a large number of features are selected in the support of the linear model at the leaf nodes, and it is unclear which features are truly relevant. Secondly, the use of heuristics for sparse linear regression does not leverage the predictive power of optimal regression methods.

A powerful approach for achieving sparse regression models with an explicit constraint on the zero norm of the weights was recently proposed in [13]. This approach is more favourable in terms of interpretability, because it is able to explicitly limit the support in a regression model to a fixed number of features. Furthermore, the authors have shown that the method outperforms Lasso in terms of accuracy and especially in false recovery rate on a test set of problems. The paper also introduced the phenomenon of *phase transitions* for the exact sparse regression problem. That is, at a critical number of observations, the performance of the algorithm begins to *improve* in terms of accuracy, false detection, and computational speed. This is a notable empirical result, as it puts in question the commonly held belief that exact algorithms are not comparable in practice with heuristics for solving large scale regression problems.

3.1.2 Contributions

There is currently no approach that leverages the interpretability and predictive power of integer optimization approaches together with tree based methods. To that end, we propose an integer optimization approach for regression that can be naturally applied to prediction trees. The technique we propose, called *SparClur* (*spar*se *clu*ster *r*egression), computes a number of regression models simultaneously for different nodes in a tree, and enforces coordination between nodes by requiring for the support within all regression models to be the same.

We demonstrate the validity of SparClur in both synthetic and real world datasets. Specifically, we show that imposing the coordination constraint (3.4) is computationally inexpensive while the formulation results in similar accuracy to the uncoordinated problem. We also demonstrate that SparClur recovers the true support while ignoring irrelevant features, and can do so for large problems in seconds.

In order to solve Problem (3.2), we use a variant of the approaches presented in [13] and [11]. In [13], the authors formulate the sparse regression problem (J=1) as a mixed integer optimization problem and suggest an outer approximation algorithm for computing provably optimal solutions. In [11], the authors develop a subgradient descent algorithm for a relaxation of the problem, and show empirically that the algorithm produces solutions of high quality, similar to the exact algorithm on large datasets. In this paper, we extend the earlier work on sparse regression methodologies to Problem (3.2).

3.1.3 Outline

We present a formulation and describe an algorithm for solving Problem (3.2) in the next section. In Section 3.3, we test the performance of solutions generated by SparClur in synthetic datasets. In Section 3.4, we apply the same approach to two separate high dimensional datasets. The first addresses a prediction problem arising in personalized diabetes management. The second relates to the prediction of stroke in participants of a longitudinal study.

3.2 Problem Formulation

In this section, we review key results from previous literature and generalize the algorithm for the case where input data is divided among clusters. We formulate (3.2) as a mixed integer optimization problem with a Tikhonov regularization term [69]. Let $\mathbf{s} \in \{0,1\}^p$ denote the common support of all sparse weight vectors. Let $\mathbf{W}_j$ be a matrix with its diagonal entries the components of $\mathbf{w}_j$. The problem that SparClur seeks to solve can be written as

$$\begin{aligned} \min_{\mathbf{w},\mathbf{s}} \quad & \frac{1}{2\gamma}\sum_{j=1}^{J} ||\mathbf{w}_j||_2^2 + \frac{1}{2}\sum_{j=1}^{J}\sum_{i\in L_j}(y_i - \mathbf{x}_i^\top \mathbf{W}_j \mathbf{s})^2 \\ \text{s.t.} \quad & \mathbf{1}^\top \mathbf{s} \le q \\ & \mathbf{w} \in \mathbb{R}^p, \mathbf{s} \in \{0,1\}^p \end{aligned} \tag{3.5}$$

where $\gamma \in \mathbb{R}_+$.

Although (3.5) is NP-hard, the algorithm for the single cluster case was able to recover provably optimal solutions in practical times, that scaled well for problems with hundreds of thousands of observations n and tens of thousands of features p.

Theorem 3.2.1. Problem (3.5) is equivalent to solving:

$$\min_{\mathbf{s}\in\{0,1\}^p:\mathbf{1}^\top\mathbf{s}\le q} \quad \frac{1}{2}\sum_{j=1}^{J}\left[\frac{1}{2}\mathbf{Y}_j^\top\left(\mathbb{I}_p + \gamma\sum_{i\in[p]} s^i \mathbf{K}_j^i\right)^{-1}\mathbf{Y}_j\right] \tag{3.6}$$

where we have used $\mathbf{K}_j^i$ to denote the micro-kernel in cluster j, that is, $\mathbf{K}_j^i := \mathbf{X}_j^i \mathbf{X}_j^{i\top}$ and $\mathbf{X}_j^{i\top}$ is a column of $\mathbf{X}_j$ corresponding to the i^{th} feature.

Proof. Proof: The proof follows from the argument in [13]. For a fixed support vector $\mathbf{s}$, the problem admits an explicit solution

$$c(\mathbf{s}) = \frac{1}{2}\sum_{j=1}^{J}\left[\mathbf{Y}_j^\top\left(\mathbb{I}_n - \mathbf{X}_j\left(\frac{1}{\gamma}\mathbb{I}_p + \mathbf{X}_j^\top\mathbf{S}\mathbf{X}\right)^{-1}\mathbf{X}_j^\top\right)\mathbf{Y}_j\right] \tag{3.7}$$

$$= \frac{1}{2}\sum_{j=1}^{J}\left[\mathbf{Y}_j^\top\left(\mathbb{I}_p + \gamma\mathbf{X}_j^\top\mathbf{S}\mathbf{X}_j\right)^{-1}\mathbf{Y}_j\right], \tag{3.8}$$

where $\mathbf{S} = \text{diag}(\mathbf{s})$. The optimum is attained at the set of weights

$$\mathbf{w}_j^* = \left(\frac{1}{\gamma}\mathbb{I}_n + \mathbf{X}_j^\top\mathbf{S}\mathbf{X}_j\right)^{-1}\mathbf{X}_j^\top\mathbf{Y}_j. \tag{3.9}$$

This allows us to write the sparse regression problem of minimizing (3.7) in a form that explicitly represents the objective as a convex function of the constrained binary vector $\mathbf{s}$. □

The convexity of the objective function in $\mathbf{s}$ enables us to apply a cutting plane algorithm to solve (3.6). It is convenient to consider the dual of (3.7) in order to derive the form of the cuts. The dual problem has the following form (Theorem 2 [13]):

$$\max_{\boldsymbol{\alpha}_j, j=1,\dots,J} \quad \frac{-\gamma}{2}\sum_{j=1}^{J} \boldsymbol{\alpha}_j^\top \mathbf{K}_j(\mathbf{s})\boldsymbol{\alpha}_j - \frac{1}{2}\boldsymbol{\alpha}_k^\top \boldsymbol{\alpha}_j + \mathbf{Y}_k^\top \boldsymbol{\alpha}_j \tag{3.10}$$

$$\text{s.t.} \quad \boldsymbol{\alpha}_j \in \mathbb{R}^{n_j} \quad \forall j \tag{3.11}$$

where $\mathbf{K}_j(\mathbf{s}) = \mathbf{X}_j\mathbf{S}\mathbf{X}_j^\top$ and n_j is the number of observations in cluster j. Here $\boldsymbol{\alpha}_j$ can be interpreted as the Lagrangian dual variables corresponding to constraints of the form $\mathbf{Y}_j = \mathbf{X}_j\mathbf{W}_j\mathbf{s}$. As (3.10) is an unconstrained quadratic problem, we can derive a closed form solution for the optimal dual variables $\boldsymbol{\alpha}_j^*$:

$$\boldsymbol{\alpha}_j^* = (\mathbb{I}_n + \gamma\mathbf{K}_j)^{-1}\,\mathbf{Y} \quad \forall j = 1,\dots,J. \tag{3.12}$$

Now, at a given candidate solution $\hat{\mathbf{s}}$, we have that our kernel matrices $\mathbf{K}_j(\mathbf{s})$ are differentiable and furthermore

$$\frac{d\mathbf{K}_j(\mathbf{s})}{d\mathbf{s}} = \sum_{i=1}^{p} s^i \mathbf{X}_j^i \mathbf{X}_j^{i\top} \quad \forall j = 1,\dots J$$

so we can always attain a subgradient as follows (Lemma 2 [13]):

$$\nabla c(\mathbf{s}) = -\frac{\gamma}{2}\sum_{j=1}^{J} \boldsymbol{\alpha}_j^* \mathbf{K}_j^\top \frac{d\mathbf{K}_j(\mathbf{s})}{d\mathbf{s}}. \tag{3.13}$$

In practice, to avoid computing the inverse of the $n \times n$ matrix in (3.12), we compute the capacitance matrix

$$\mathbf{CM}_j = \frac{I}{\gamma} + \mathbf{Z}_j(\mathbf{s})^\top \mathbf{Z}_j(\mathbf{s})$$

where $Z_j(\mathbf{s})$ is formed by taking the columns of $\mathbf{X}_j$ that are in the support vector $\mathbf{s}$, and this enables us to evaluate the matrix inverse as

$$(\mathbb{I} + \gamma\mathbf{K}_j(\mathbf{s}))^{-1} = \mathbb{I} - \mathbf{Z}_j(\mathbf{s})(\mathbf{CM}_j)^{-1}\mathbf{Z}_j(\mathbf{s})^\top.$$

This formulation gives rise to the SparClur cutting plane algorithm.

Algorithm 1 SparClur

Input: $\mathbf{X}_j \in \mathbb{R}^{n_j \times p}, \mathbf{Y}_j \in \mathbb{R}^{n_j}, j = 1, \dots, J, q \in [p], \gamma \in \mathbb{R}_{++}$
Output: $\mathbf{s} \in \{0,1\}^p$

```
procedure CUTTING PLANE ALGORITHM
    $\mathbf{s}_1 \leftarrow$ warm start
    $\eta_1 \leftarrow 0$
    $\nu \leftarrow 0$
    $c(\mathbf{s}_1) \leftarrow \infty$
    while $\eta_\nu < c(\mathbf{s}_\nu)$
        $\nu \leftarrow \nu + 1$
        for $j \in [J]$
            $\mathbf{S} \leftarrow \mathrm{diag}(\mathbf{s})$
            $\boldsymbol{\alpha}_j^* \leftarrow \mathbf{Y}_j - \mathbf{X}_j \mathbf{S}(\mathbb{I}_q/\gamma + \mathbf{X}_j^\top \mathbf{S} \mathbf{X}_j)^{-1} \mathbf{X}_j^\top \mathbf{S} \mathbf{Y}_j$
        $c(\mathbf{s}_\nu) \leftarrow \frac{1}{2} \sum_j \mathbf{Y}_j^\top \alpha_j^*$
        for $i \in [p]$
            $\nabla c_i \leftarrow \frac{\gamma}{2} \sum_j (\mathbf{X}_j^{i\top} \boldsymbol{\alpha}_j^*)^2$
        $\mathbf{s}_{\nu+1}, \eta_{\nu+1} \leftarrow \arg\min_{\mathbf{s},\eta} \{\eta : \eta \geq c(\mathbf{s}_t) + \nabla c(\mathbf{s}_t)^\top (\mathbf{s} - \mathbf{s}_t)\} \quad \forall t \in [\nu]$
```

As well as the given exact cutting plane algorithm, the SparClur formulation is amenable to algorithms for the relaxation of (3.2) such as the subgradient descent algorithm suggested in [11] for the convex relaxation of the sparse regression problem. The convex relaxation is useful for providing warm starts to a mixed integer solver, but can also provide high quality solutions on its own. The convex relaxation takes the form:

$$\min_{\mathbf{s} \in Conv(\{0,1\}^p : \mathbf{1}^\top \mathbf{s} \leq q)} \max_{\boldsymbol{\alpha}_j \in \mathbb{R}^{n_j}} \quad f(\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_J, \mathbf{s}) = \tag{3.14}$$

$$\min_{\mathbf{s} \in Conv(\{0,1\}^p : \mathbf{1}^\top \mathbf{s} \leq q)} \max_{\boldsymbol{\alpha}_j \in \mathbb{R}^{n(j)}} \quad \frac{-\gamma}{2} \sum_{j=1}^{J} \boldsymbol{\alpha}_j^\top \mathbf{K}_j(s) \boldsymbol{\alpha}_j - \frac{1}{2} \boldsymbol{\alpha}_j^\top \boldsymbol{\alpha}_j + \mathbf{Y}_j^\top \boldsymbol{\alpha}_j \tag{3.15}$$

and we can exchange the order of the global minimization and maximization operators, so (3.14) is equivalent to

$$\max_{\boldsymbol{\alpha}_j \in \mathbb{R}^{n_j}} \; -\frac{1}{2} \sum_{j=1}^{J} \left[\boldsymbol{\alpha}_j^\top \boldsymbol{\alpha}_j +_{\mathbf{w}} \mathbf{Y}_k^\top \boldsymbol{\alpha}_j - \max_{\mathbf{s} \in Conv(\{0,1\}^p : \mathbf{1}^\top \mathbf{s} \leq q)} \frac{\gamma}{2} \sum_i s_i \boldsymbol{\alpha}_j^\top \mathbf{X}_j^i \mathbf{X}_j^{i\top} \boldsymbol{\alpha}_j \right]. \tag{3.16}$$

The inner maximization problem always has at least one analytic solution that can be constructed by finding q indices i where $\sum_j \alpha_j^\top \mathbf{X}_j^i \mathbf{X}_j^{i\top} \alpha_j$ take on the largest values in order, and assigning $s_i = 1$ to those indices. The outer maximization problem can be solved via a non-smooth optimization algorithm. That is, for a given candidate dual solution $\hat{\alpha}_1, \dots, \hat{\alpha}_J$ we can analytically compute the optimal support vector and a subgradient $\nabla f(\hat{\alpha}_j, \mathbf{s})$ and apply a suitable global first order method to (3.16).

3.3 Experiments with Synthetic Data

An exact method for sparse regression is only successful if it can be demonstrated that the method is capable of producing solutions that contain the true sparsity pattern when this pattern is known, without including features that are not truly relevant ("false positives") in the solution. We will demonstrate that Algorithm 1 is capable of recovering solutions that capture all relevant features in synthetic datasets with a known underlying sparsity pattern, and no features that are not part of the true underlying sparsity pattern. In other words, SparClur recovers *the whole truth and nothing but the truth.* In this section we look into experiments using synthetic data in order to address three key questions:

1. Does our mixed integer formulation recover correct solutions to the sparse regression problem, particularly in the presence of noise?

2. Does SparClur enjoy practical solving times as the dimensionality of a problem grows?

3. What is the cost of imposing the assumption of common support among clusters, when there is no such phenomenon in the underlying data?

We measure the ability of our formulation to recover the truth by reporting the accuracy A and the false positive rate F, defined below. Let $\mathrm{supp}(\mathbf{w}_{true})$ denote the known true support in a synthetic dataset. Then for solution $\mathbf{w}^*$ we have

$$\begin{aligned} A &= \frac{|\mathrm{supp}(\mathbf{w}_{true}) \cap \mathrm{supp}(\mathbf{w}^*)|}{q} \\ F &= \frac{|\mathrm{supp}(\mathbf{w}^*) \setminus \mathrm{supp}(\mathbf{w}_{true})|}{|\mathrm{supp}(\mathbf{w}^*)|}. \end{aligned}$$

All experiments were run on a Linux system with an Intel Xeon CPU E5-2650 processor. All time related results report the time taken to perform tasks on a single processor. All our formulations were written in Julia [14], and all optimization problems were built using JuMP [30] and solved in Cplex 12.8.

3.3.1 Support Recovery

In order to investigate the ability of our formulation to recover the true support in the presence of noise, synthetic data were generated as follows. Each entry of the matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ was independently generated from a $\mathcal{N}(0,1)$ distribution for n ranging between 100 and 500 observations and $p = 2000$. Our observations were randomly and evenly divided among J clusters, creating clusters of observations $\mathbf{X}_1, \dots, \mathbf{X}_J$. The value of J was taken from all values in the range $\{1, 2, 5, 10, 20\}$. The set of features in the support $\mathcal{S}$ was fixed with $|\mathcal{S}| = q = 10$ randomly selected features. For each feature i in the true support, a corresponding coefficient $w^i \in \{-1, 1\}$ was sampled. We compute the target variable $\mathbf{Y}_j = \mathbf{X}_j \mathbf{w}_j + \boldsymbol{\xi}_j$ where $\boldsymbol{\xi}_j \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ was scaled so that we have a signal-to-noise ratio $||\mathbf{Y}_j||/||\boldsymbol{\xi}_j|| = 20$.

We generated five synthetic datasets as described above for varying values of n, and we report the mean out of sample accuracy and false positive rate for each n. These are shown in the plots of Figure 3.2. For each datapoint shown, the value of γ was taken to be some constant multiplied by q/n. The value of this constant was chosen following a cross validation procedure for each value n used in testing out of sample. The plots demonstrate the occurrence of a *phase transition*, and demonstrate that the point of this phase transition depends on the number of observations in each cluster.

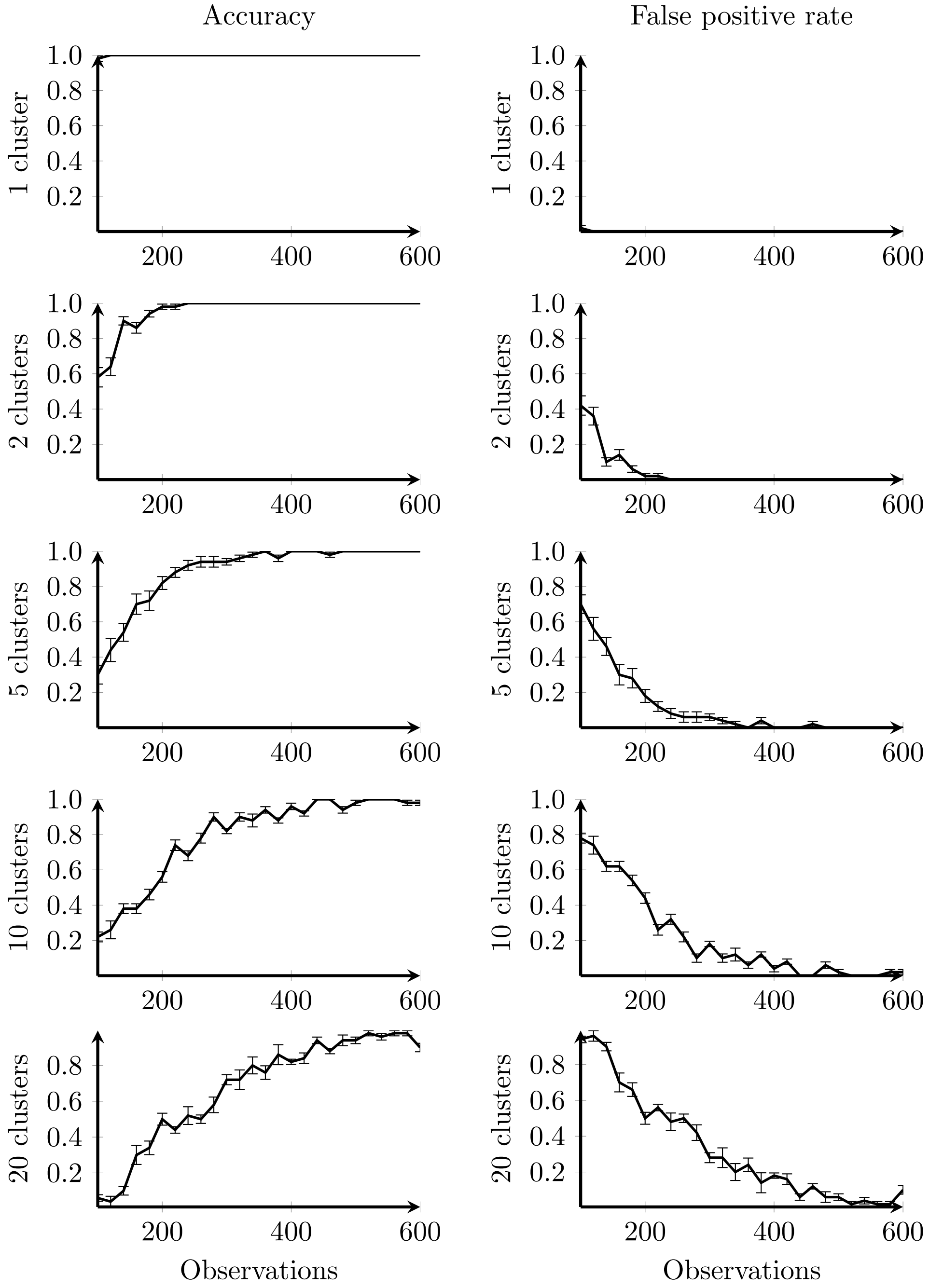
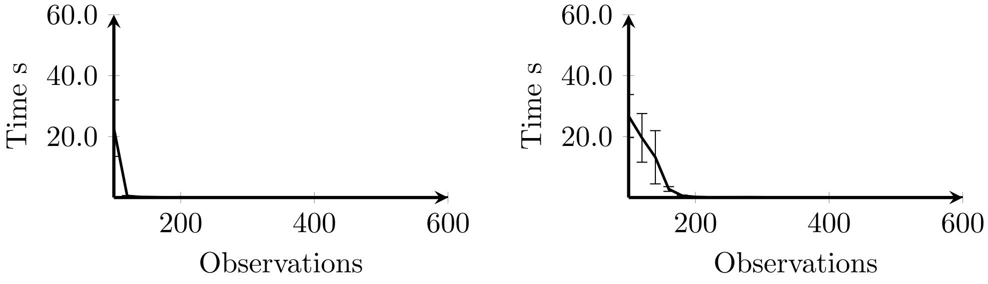


Figure 3.2: Accuracy and false positive rate as a function of number of observations for synthetic data with SNR=20, q=10. All problems were solved using Algorithm 1. Each curve passes through the average measurement made over five sets of synthetic data and error bars correspond to one standard deviation.

Time of Phase Transition

Our experiments reveal an interesting phenomenon, where the solving time of a problem begins to *decrease* once some critical number of observations is exceeded. This has an interesting implication for the SparClur formulation. As an example, consider the computational time for the synthetic problem described above, shown in Figures 3.3a for $J = 1$ and 3.3b for $J = 5$. For a modeler computing regression weights at several leaf nodes, it is desirable for the number of observations at each leaf to be greater than the critical value mentioned, since this allows them to enjoy significantly lower solving times. A key advantage of SparClur is that the coordination imposed reduces the number of observations necessary to attain the solving times observed beyond the *phase transition*. For the case illustrated in Figures 3.3a and 3.3b, if a model consisted of five leaf nodes, then around 200 observations would be sufficient to achieve phase transition with SparClur. On the other hand, the single cluster model experiences a phase transition beyond 140 observations, meaning that if a modeler was to use an uncoordinated regression model at each leaf, 5 × 140 = 700 observations would be necessary.



(a) One cluster

(b) Five clusters

Figure 3.3: Computational times in seconds as a function of number of observations for a model with binary weights and SNR = 20.

3.3.2 Scalability

As well as being able to attain 100% accuracy and 0% false positive rate, we want to ensure that the SparClur formulation continues to allow practical solving times as the size of the input data grows. Table 3.1 summarizes the solving times we observe as we increase the number of observations to the range of the hundreds of thousands, and the number of features to the tens of thousands. At this scale, we are able to recover the full support with no false detection in *seconds*.

γ	p	$n = 20{,}000$	$n = 50{,}000$	$n = 100{,}000$
0.005	20,000	16.3	33.6	63.6
0.01	20,000	14.9	33.1	63.2
0.02	20,000	15.1	35.2	68.9
0.005	50,000	6.99	14.6	26.8
0.01	50,000	6.45	13.9	26.0
0.02	50,000	6.53	13.9	27.2

Table 3.1: Computational times in seconds (mean over five datasets) for different values of n, p, and γ. In each experiment accuracy was 100% and false positive rate was 0%.

3.3.3 Effects of Clusters with Varying Support

In these experiments we seek to explore the behavior of SparClur when it is applied to observations that do not truly share the same support. To do so, we generate data for observations $\mathbf{X}_j$ and $\mathbf{Y}_j$ as described in the previous section, with the number of clusters $J = 2$. The weights for each cluster $\mathbf{w}_1, \mathbf{w}_2$ were generated so that there are 10 features in the true support of each cluster, but not necessarily the same 10 features in both clusters. The number of features in $\{\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)\}$ was varied.

When we come to build a model for our synthetic data, we must assume some underlying sparsity q which may be lower than, or greater than, the total number of features in both clusters $|w_1 \cup w_2|$. Of course when $q < |\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$, it is not possible to attain an accuracy of 100%. Instead, the maximum attainable accuracy is $\frac{q}{|\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|}$.

In Figures 3.4 and 3.5, the dashed curves correspond to the maximum attainable accuracy. The points correspond to the accuracy attained each time the problem is solved with SparClur. In every case, the accuracy matches closely with the maximum attainable accuracy, and we never detect any features not in the support of one of the two clusters, except when $q > |\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$. We do not claim, however, that when $q < |\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$, features in the support of both clusters, $|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)|$, are always in the set of features discovered. Rather, any of the features in any support vector may be in the solution.

We note that selecting a value of q that is too large is not an issue for the performance of a model if weights are obtained with a least squares calculation, once the support is chosen. In practice, the choice of q would be determined following a cross validation procedure. When the ground truth is not known, this would be done by measuring the out of sample R^2 for different choices of q. Figure 3.6 shows the out of sample R^2 for different values of q. We see that the optimal choices of q are 18 in the first experiment, and 15 in the second experiment from the figures (which conforms with $|\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$).

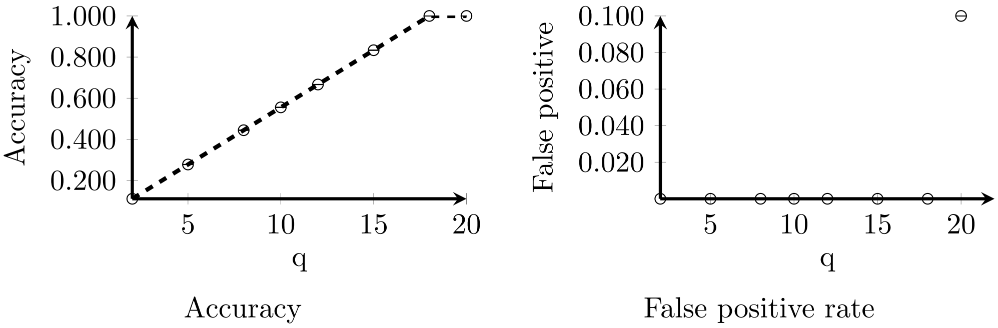
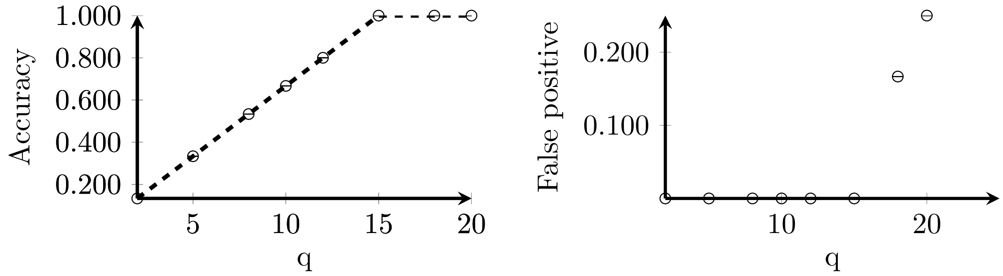


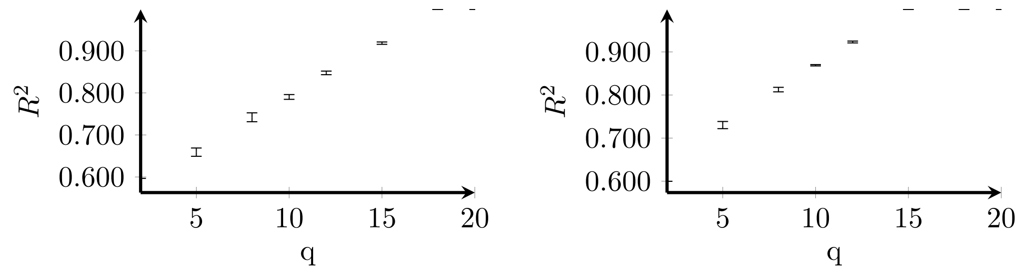
Figure 3.4: Accuracy and false positive rate when $|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)| = 2$.



Accuracy

False positive rate

Figure 3.5: Accuracy and false positive rate when $|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)| = 5$.



$|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)| = 2$.

$|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)| = 5$.

Figure 3.6: Out of sample R^2 as a function of q.

3.3.4 Findings from Synthetic Experiments

Our experimentation with synthetic data suggests that SparClur enjoys several properties that make the approach useful in practical settings.

1. The algorithm recovers the true support in a set of features when this support is known, and is capable of successfully ignoring irrelevant features.

2. The algorithm is practical for offline problems where the number of observations is in the hundreds of thousands, and the number of features is in the tens of thousands. That is, we can attain high quality solutions for problems of such scale in seconds. Furthermore, the quality of solutions for a fixed number of features has the potential to be higher with SparClur than with uncoordinated sparse regression, because we often require a *smaller* number of observations to be present before the phase transition phenomenon occurs.

3. Any increase in q (which generally reduces the interpretability of a model in terms of the number of features included) has never resulted in features being included in the model that should not be part of the true support, when q was chosen to be smaller than the total number of relevant features. When the underlying regression models in clusters do not truly share a common sparsity pattern, there is a tradeoff to be made between interpretability (which improves as q decreases) and accuracy (which improves up to a limit as q increases).

3.4 Case Studies with High Dimensional Data

One of the key motivations for the SparClur formulation arises in medical studies, particularly in the realm of personalized medicine. In this section, we investigate the performance of the algorithm in three different real-world prediction problems.

The first medical prediction task we look at is the problem of predicting glucose levels in diabetes patients. From our experience, many clinicians intuitively use a series of questions to arrive at an estimate for patient outcomes. Key characteristics of the patient such as weight, age, and medical history could be taken into consideration. However with an algorithmic approach, the unique features of each patient can be used in a model to predict glucose levels. A related prescriptive problem has previously been studied in [9], where k-nearest neighbor regression was applied to predict prescriptive outcomes. Using decision trees, we can further personalize the prediction problem for each patient by taking into account the fact that different features may play varying roles in the progression of a person's health. For example, the past adherence to a line of treatment may affect glucose levels in a different way for elderly male patients than it does for middle-aged females. At the same time, it makes medical sense that the same covariates have a nonzero impact across all patients regardless of their differences.

The second high dimensional dataset we turn our attention to is derived from the Framingham Heart Study [3]. This dataset has previously been studied in [4] in a classification setting, to predict the event occurrence of stroke in subjects. The same dataset has been used to study a number of other medical conditions, due to the richness of the features collected and the longitudinal aspect of the study. Here, we seek to predict two continuous outcomes of interest. The first, is the change in blood pressure of patients at subsequent visits. The second, is time of stroke occurrence from the first observation of a subject. We had access to demographic features including age, gender, and BMI, as well as biological information about patients derived from blood test data.

In stroke management, as in diabetes management, clinicians often use a checklist of questions to determine a patient's risk for the condition [34, 48]. Currently, these risk score calculators give points to each individual risk factor and assign a risk percentage based on thresholds of the cumulative score. For example, one of the most widely used approaches for stroke risk assessment is CHADS$_2$ [60], which assigns a score 0—6 to patients, each corresponding to a likelihood of stroke between 0 and 18%. Generally, such approaches have several shortcomings. For instance, in CHADS$_2$, equal weight is assigned to each risk condition, a linear relationship between factors is assumed, there is a limited number of features which may not be predictive across different patient groups, and there is no differentiation of risk within buckets assigned, providing a discrete numeric score rather than a continuous measure. Nonetheless the advantage of a method like CHADS$_2$ is that clinicians prefer a highly explainable model.

3.4.1 Description of Data

We obtained electronic medical records (EMR) for over 1.1 million patients at Boston Medical Center (BMC) from 1999 to 2014 for the glucose prediction problem. In this dataset, 10,806 patients met the inclusion criteria described in [9]. We had access to demographic data, including date of birth, sex, and ethnicity, and to all BMC EMR data, including a history of drug prescriptions and measurements of height, weight, BMI, and HbA1c (an indicator of past blood sugar levels) as well as creatinine levels. All together, the model considered 85 features.

The Framingham Heart Study contains the examination data from 41 clinical exams which started in 1948 and have been followed up through 2010. Our data comprises two cohorts: the *Original Cohort*, consisting of 5,209 respondents of a random sample of two thirds of the adult population of Framingham, Massachusetts. These respondents were 30 to 62 years of age by household, starting in 1948 with follow-ups until 2010. The *Offspring Study* Cohort was initiated in 1971 with a sample of 5,124 men and women, consisting of the offspring of the Original Cohort and their spouses.

Unifying the two cohorts, we have patient characteristic data at each visitation, for 10,092 unique patients. Of those, 1,266 (roughly 10%), went on to have an occurrence of stroke by the end of the study.

We considered only the patients who experienced a stroke for the time of stroke prediction problem. We retained the health information from their initial baseline visit and computed the number of days from when that data was collected to the date of their first occurrence of stroke. Overall, the model we built had 1,266 observations and 40 features.

For the blood pressure prediction problem, we calculated the change in systolic blood pressure of patients between consecutive visits, and treated each pair of consecutive visits as an observation. The final model had 91,955 observations and 41 features.

3.4.2 Comparison of Methods

We examined the performance of optimal regression trees with point predictions at leaf nodes, as well as linear prediction models at leaf nodes. All linear prediction models were constructed after the optimal regression tree (with point predictions at the leaves) was found and the parameters for the tree were cross validated. The leaves of each tree were treated as clusters. We considered building linear models using Lasso[1], sparse regression (without coordination), and SparClur.

The linear models built with SparClur were found by utilizing Algorithm 1, as well as using the convex relaxation of our formulation. When solving with cutting planes, we set a time limit of two minutes in CPLEX and use the incumbent solution if the time limit is reached. We did not employ an exact cutting plane algorithm to solve the uncoordinated sparse regression problem. The deepest trees we obtained typically had hundreds of leaf nodes. Therefore, training and cross-validating for appropriate parameters using uncoordinated sparse regression, which involves building linear models within each leaf separately, would take over a week of computational time. In contrast, the results we are able to report using the exact mixed integer formulation with SparClur correspond to only hours or days of computation.

We measured the average out of sample accuracy from five different training and testing splits of our data and report the out of sample R^2 for trees of increasing depth. Our results are summarized in Tables 3.2 — 3.4 and depicted in Figures 3.7 — 3.9. Each tree we built was created using the software package `OptimalTrees.jl` described in [29] and for each regression model, the hyperparameters q (in the range 1—10) and γ were chosen following a cross validation

[1] For Lasso regression we use the implementation of `https://github.com/JuliaStats/GLMNet.jl` [43].

procedure. We chose to restrict the sparsity of our models to 10 features (including a bias term) or fewer, as we consider this to be an appropriate number of features for clinicians to reflect on.

Depth	Lasso	ORT	SparClur:	SparClur:	Sparse
	at leaves		exact	relaxation	(uncoordinated)
0	0.506	0.000	0.499	0.464	0.464
1	0.521	0.323	0.514	0.498	0.490
2	0.524	0.438	0.517	0.502	0.490
3	0.532	0.476	0.524	0.509	0.490
4	0.535	0.502	0.530	0.525	0.495
5	0.535	0.511	0.530	0.526	0.497
6	0.535	0.516	0.530	0.527	0.497
7	0.535	0.516	0.530	0.527	0.497
8	0.535	0.516	0.530	0.527	0.497
9	0.535	0.516	0.530	0.526	0.497
10	0.535	0.516	0.530	0.526	0.497

Table 3.2: Out of sample R^2 for prediction of glucose levels using different depths.

Depth	Lasso	ORT	SparClur:	SparClur:	Sparse
	at leaves		exact	relaxation	(uncoordinated)
0	0.364	-0.005	0.352	0.261	0.261
1	0.359	0.172	0.365	0.303	0.272
2	0.358	0.234	0.350	0.312	0.316
3	0.357	0.253	0.342	0.316	0.320
4	0.357	0.253	0.341	0.316	0.320
5	0.357	0.253	0.341	0.316	0.312
6	0.357	0.253	0.341	0.316	0.312
7	0.357	0.253	0.341	0.316	0.312
8	0.357	0.253	0.341	0.316	0.312
9	0.357	0.253	0.341	0.316	0.312
10	0.357	0.253	0.341	0.316	0.312

Table 3.3: Mean out of sample R^2 for prediction of days until stroke onset using different depths.

Depth	Lasso	ORT	SparClur:	SparClur:	Sparse
	at leaves		exact	relaxation	(uncoordinated)
0	0.306	0.000	0.296	0.281	0.281
1	0.341	0.151	0.328	0.319	0.300
2	0.521	0.372	0.516	0.503	0.465
3	0.527	0.459	0.523	0.513	0.468
4	0.528	0.487	0.522	0.518	0.506
5	0.528	0.498	0.524	0.520	0.506
6	0.528	0.505	0.524	0.519	0.507
7	0.528	0.508	0.524	0.519	0.511
8	0.528	0.508	0.524	0.519	0.511
9	0.528	0.508	0.524	0.519	0.511
10	0.528	0.508	0.524	0.519	0.511

Table 3.4: Mean out of sample R^2 for prediction of change in blood pressure using different depths.

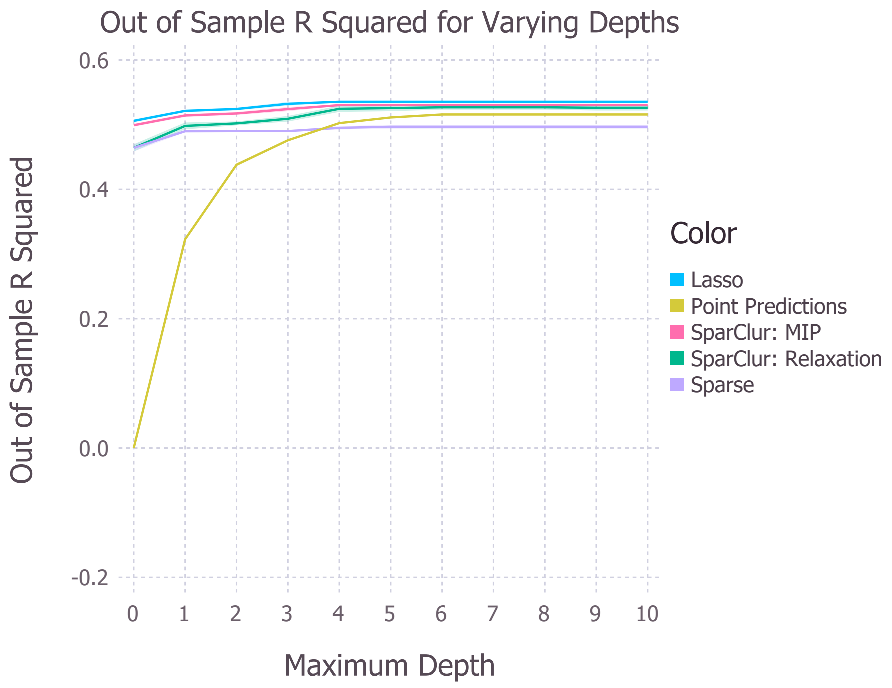


Figure 3.7: Out of sample R^2 as a function of depth for prediction of glucose levels.

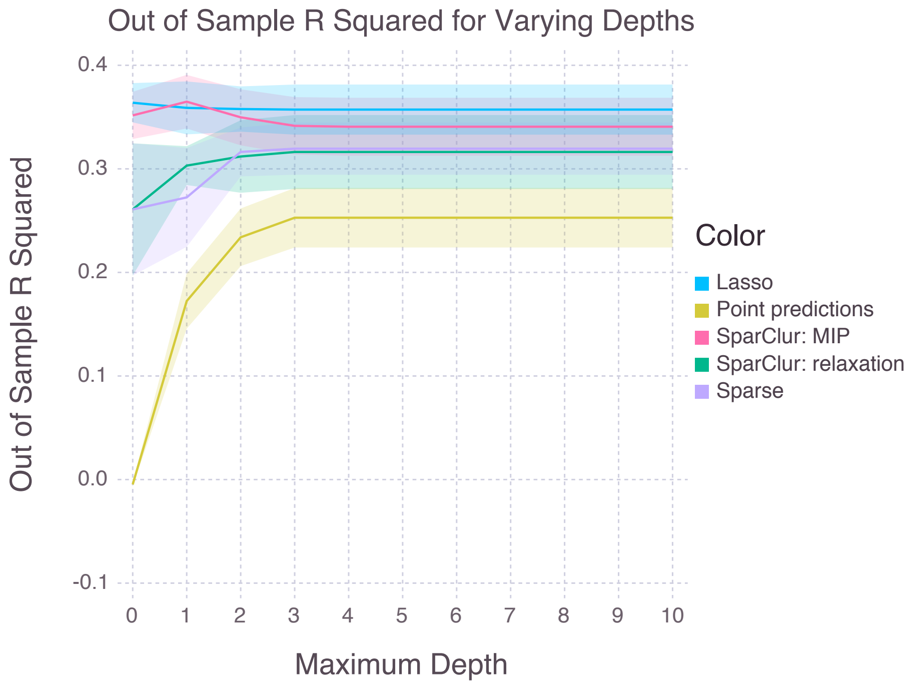


Figure 3.8: R^2 as a function of tree depth for prediction of days until stroke onset.

The models created from our larger datasets (in glucose and blood pressure prediction) attained R^2 scores of around 0.5. The best models we produced for predicting time of stroke,

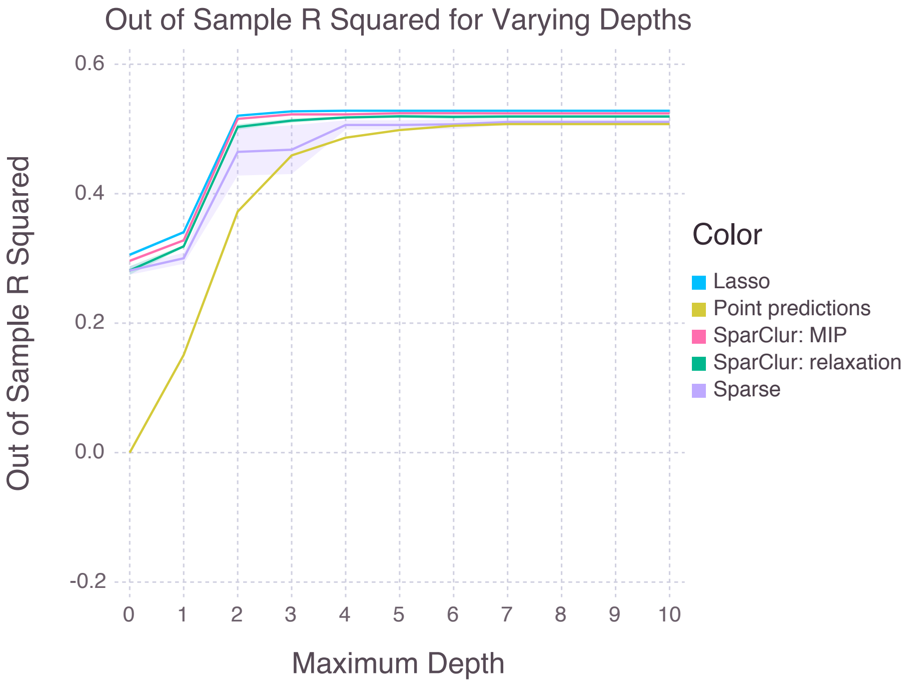


Figure 3.9: R^2 as a function of tree depth for prediction of change in blood pressure.

attained R^2 scores of 0.3 — 0.4. In the stroke prediction problem, we also had a lot more variance in the R^2 across folds. This is likely due to the smaller sample size of 1266, while the diabetes and blood pressure datasets had around 10,000 and 91,000 observations.

The models we obtained with sparse regression methods (both coordinated and uncoordinated) had significantly fewer features than the trees obtained with Lasso at the leaves, in each example we studied.

In the glucose prediction problem, some of the trees modified with Lasso had close to 50 features in the support set. In contrast, the trees obtained with SparClur contained at most 10 variables which participated in the entire tree, and these variables appeared repeatedly among different trees. An example of a tree we obtained (with features denoted by $x_1, x_2, \ldots x_{85}$) is depicted in Appendix 3.6.2. A few of the features present in the support for almost all testing folds in our deepest tree were:

- Age of patient,
- Time since diagnosis,
- Whether a previously assigned treatment was *none*
- One or several metrics relating to past HbA1c (either the change from the last two visits, or a quantile of HbA1c),
- Whether previously assigned regimen included Metformin,
- Whether the second to last, third to last, or fourth to last treatment was *none*.

In the blood pressure prediction models, several trees modified with Lasso had over 20 features in the support set, often varying among different trees. SparClur again resulted in trees with at most 10 variables participating in the support of all leaves in the tree. An example tree we obtained can be found in Appendix 3.6.3. A few features present in the support for most of the leaves were:

- Age,
- Gender,
- Past SBP,
- Whether they were on AHT,
- BMI.

These features are clearly medically significant.

3.4.3 General Trends

The results in Tables 3.2—3.4 display some interesting patterns. Each of the regression methods examined attained an R^2 score within 5% of the other methods, apart from the point-prediction model for time of stroke. Given that the difference in interpretability between the models employed is significant, the similarity in R^2 scores demonstrates that the price to be paid for imposing additional structure that favors this interpretability, tends to be small.

Interestingly, the uncoordinated sparse regression approach had inferior out of sample performance to optimal regression trees with point predictions in the glucose prediction problem, and inferior performance to SparClur in the stroke related problems. This would typically be an indicator of overfitting by a model. In this case, the behavior could be an artifact of a large number of clusters and an insufficiently large number of observations within each cluster, for optimal sparse regression to be performant. The performance of SparClur is always a few percentage points higher than optimal regression trees at high depths.

We also make the observation that the optimization of the exact coordinated regression problem provides a relatively small improvement in the out of sample R^2 over the relaxation solution in each case study. We note that a warm-start for the cutting plane algorithm is always computed using the convex relaxation of the exact method, and the integer optimization solution often consists of swapping a small number of variables selected by the relaxation with unselected variables (this can bee seen in the example in Appendix 3.6.2). This suggests that the convex relaxation of the problem is able to provide high quality solutions for the problem studied.

Notably, the models we obtained with sparse regression methods (both coordinated and uncoordinated) gave rise to trees that were significantly more interpretable than the trees obtained with Lasso at the leaves. In each case, there were Lasso models that contained not only significantly more covariates in the leaves, but different subsets as well. In addition, for each of our cases, scores would plateau at a depth of around 3—4. This indicates that particularly deep trees are unnecessary, maintaining explainable models.

For stroke prediction, the tree structure did not provide as much lift (depth 0/1 performed best). For the other two cases, however, the R^2 increased significantly with depth. This could be because the features did not differ much in the leaves, indicating that the problem of stroke prediction is more uniform across certain demographics than believed.

In the stroke prediction problem, we also had a lot more variance in the R^2 across folds. This is likely due to the smaller sample size of 1266, while the diabetes and blood pressure datasets had 10,00 and 91,000 observations.

In comparison to current state of the art patient prediction methodologies which rely on a fixed series of questions, SparClur leverages the predictive power of ORTs, allows us to capture nonlinear relationship between observations and the target variable, and naturally performs different splits at different regions of the tree, suggesting that there is a potential for improvement.

3.5 Conclusions

We offer SparClur as an approach for building regression models within tree based prediction methods that combines state of the art accuracy, and interpretability. SparClur enforces additional structure within predictive models, but leads to models that are arguably more interpretable than other linear regression methods. Furthermore, we have shown with synthetic data that the method is correct, scalable, and capable of attaining stronger result than sparse regression without coordination when the number of observations available is below a certain threshold. In the large scale datasets we have studied, SparClur improves on the accuracy of ORTs with point predictions, and has very similar out of sample accuracy to models utilizing uncoordinated sparse regression, and Lasso regression. In other words, we see a substantial gain in interpretability at a very small cost to accuracy.

3.6 Appendices

3.6.1 Listing of Model Covariates

x_1	Visit Number	x_{18}	Past metformin user
x_2	Line Number	x_{19}	Adherence
x_3	Age	x_{20}	Number of drugs last prescribed
x_4	Time from diagnosis	x_{21}	Number of drugs prescribed two visits ago
x_5	Past HbA1c	x_{22}	Number of drugs prescribed three visits ago
x_6	Last HbA1c increment	x_{23}	Kidney contraindication
x_7	HbA1c on second to last visit	x_{24}	Line choice
x_8	HbA1c.change	x_{25}	Current number of drugs
x_9	Median HbA1c	x_{26}	Sex
x_{10}	75th quantile HbA1c	x_{27}	Race: is hispanic
x_{11}	25th quantile HbA1c	x_{28}	Race: is white
x_{12}	Mean HbA1c	x_{29}	Race: other
x_{13}	Previous BMI	x_{30-40}	Categories of last treatment
x_{14}	BMI median	x_{41-51}	Categories of second to last treatment
x_{15}	BMI 75% quantile	x_{52-62}	Categories of third to last treatment
x_{16}	BMI 25% quantile	x_{63-73}	Categories of fourth to last treatment
x_{17}	BMI mean	x_{74-84}	Categories of current treatment

Table 3.5: Description of all features used to build models in the diabetes case study. There are 85 features in total, including a dummy variable for offset.

x_1	Age	x_{19}	BMI
x_2	Gender	x_{20}	Hemat
x_3	Systolic Blood Pressure	x_{21}	Glucose (blood)
x_4	Diastolic Blood Pressure	x_{22}	Glucose (urine)
x_5	Anti Hypertension Treatment	x_{23}	Albumin (urine)
x_6	Nitrates	x_{24}	XRay Enlarg Before
x_7	Diuretics	x_{25}	Ventricular Rate
x_8	Diabetes 200	x_{26}	Left Ventricular Hypertrophy
x_9	Diabetes 150	x_{27}	Intravent block
x_{10}	Diabetes 140	x_{28}	Atrioventr block
x_{11}	Smoking	x_{29}	T wave
x_{12}	Cardiovascular Disease	x_{30}	ST segment
x_{13}	Afib	x_{31}	Prem beats
x_{14}	Coronary Artery Bypass Graft	x_{32}	Hypertension
x_{15}	Percutaneous Coronary Intervention	x_{33}	Cholest total
x_{16}	Myocardial Infarction	x_{34}	HDL
x_{17}	Transient Ischemic Attack	x_{35}	Exam Number
x_{18}	Marital Status	x_{36}	Cohort

Table 3.6: Description of all features used to build models in the stroke case study. There are 37 features in total, including a dummy variable for offset.

x_1	Dates	x_{21}	Hemat
x_2	Age	x_{22}	Glucose (blood)
x_3	Gender	x_{23}	Glucose (urine)
x_4	Systolic Blood Pressure	x_{24}	Albumin (urine)
x_5	Diastolic Blood Pressure	x_{25}	XRay Enlarg Before
x_6	Anti Hypertension Treatment	x_{26}	XRay Enlarg After
x_7	Nitrates	x_{27}	Ventricular Rate
x_8	Diuretics	x_{28}	Left Ventricular Hypertrophy
x_9	Diabetes 200	x_{29}	Intravent block
x_{10}	Diabetes 150	x_{30}	Atrioventr block
x_{11}	Diabetes 140	x_{31}	T wave
x_{12}	Smoking	x_{32}	ST segment
x_{13}	Cardiovascular Disease	x_{33}	U wave
x_{14}	Afib	x_{34}	Prem Beats
x_{15}	Coronary Artery Bypass Graft	x_{35}	Hypertension
x_{16}	Percutaneous Coronary Intervention	x_{36}	Cholest Total
x_{17}	Myocardial Infarction	x_{37}	HDL
x_{18}	Transient Ischemic Attack	x_{38}	Exam Number
x_{19}	Marital Status	x_{39}	Cohort
x_{20}	BMI	x_{40}	Had Stroke

Table 3.7: Description of all features used to build models in the blood pressure case study. There are 41 features in total, including a dummy variable for offset.

3.6.2 Tree for Glucose Level Prediction

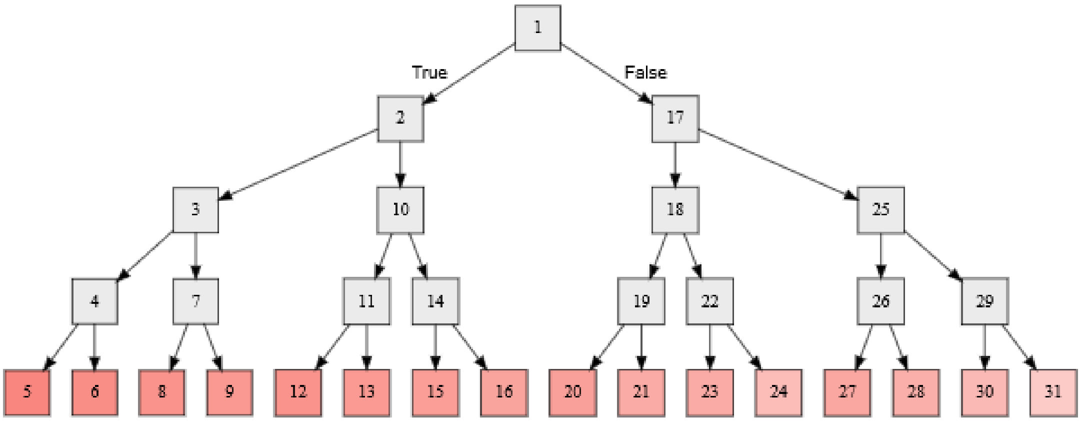


Figure 3.10: Optimal regression tree structure for the diabetes case study when maximum depth is four. Out of sample R^2 was highest at this depth.

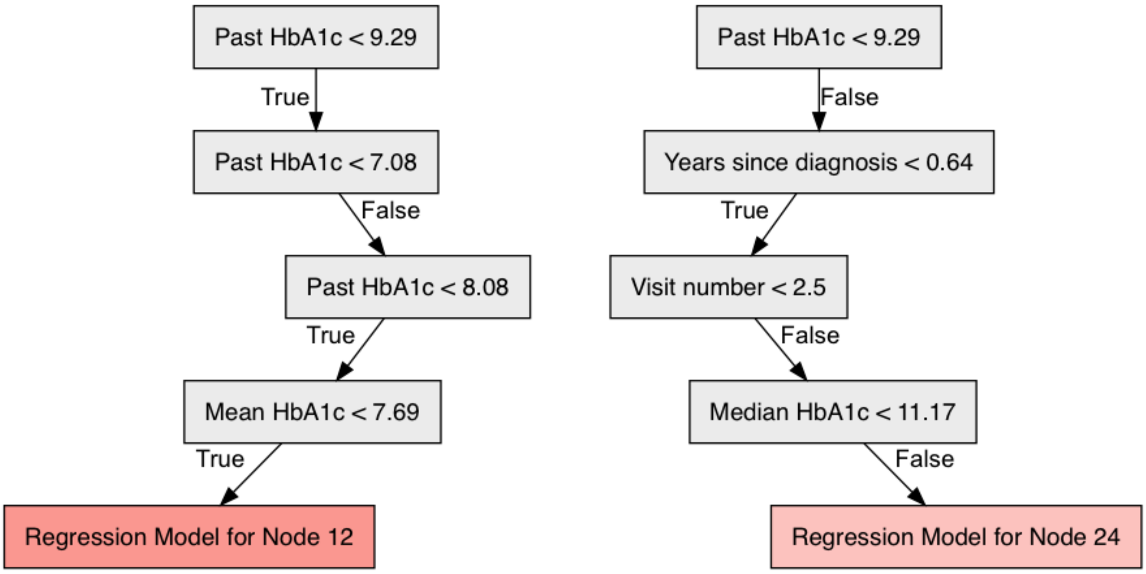


Figure 3.11: Example of trajectories of patient characteristics that correspond to nodes 12 and 24 in the optimal regression tree above.

	Node 12 linear models
Lasso at leaves	Model with 48 variables
SparClur: relaxation	$0.2 - 0.44x_3 + 0.108x_4 + 0.291x_{10} + 0.163x_{14} - 0.391x_{15}$ $-0.337x_{17} + 0.001x_{19} + 0.057x_{37}0.035 + x_{81}$
SparClur: exact	$0.21 - 0.466x_3 + 0.103x_4 + 0.015x_6 + 0.24x_7 + 0.892x_9$ $+2.323x_{10} - 0.003x_{18} + 0.059x_{37}$
Sparse (uncoordinated)	$-0.05 + 0.002x_3 + 7.145x_7 + 0.542x_{14} + 0.059x_{15} + 0.119x_{16}$ $-1.151x_{17} + 0.107x_{19} - 0.012x_{34} + 0.195x_{26} + 0.11x_{48}$
	Node 24 linear models
Lasso at leaves	Model with 20 variables
SparClur: relaxation	$0.48 - 1.614x_3 + 0.426x_4 + 0.022x_{10} + 0.038x_{14} + 0.292x_{15}$ $+0.625x_{17} + 0.128x_{37} - 0.155x_{81}$
SparClur: exact	$0.25 - 1.6x_3 + 0.574x_4 - 0.065x_6 + 0.062x_7 - 6.572x_9$ $+6.564x_{10} + 0.069x_{37}$
Sparse (uncoordinated)	$0.13 + 0.655x_4 + 0.328x_5 + 0.084x_8 + 0.075x_{10} + 0.293x_{15}$ $+0.49x_{17} + 0.099x_{48} + 1.292x_{59} + 0.887x_{70}$

Figure 3.12: Regression models using different approaches for nodes 12 and 24 in the tree of Figure 3.10. At both nodes, SparClur gives the same support. Sparse regression gives models with the same number of features but different support. Lasso produced models that varied substantially in different leaves in terms of the number of variables and the features chosen in the model.

3.6.3 Tree for Blood Pressure Prediction

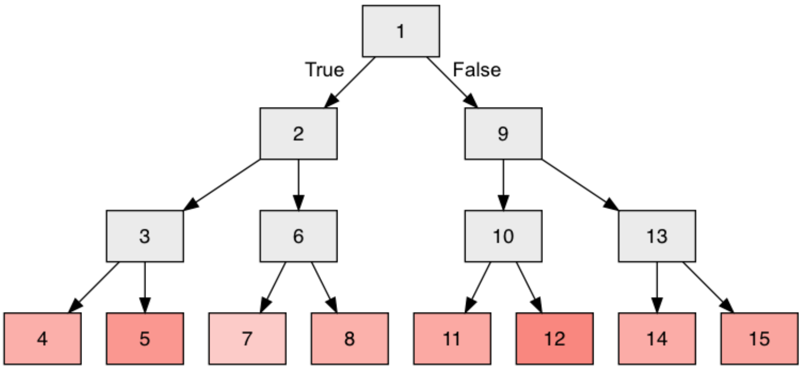


Figure 3.13: Optimal regression tree structure for the blood pressure case study when maximum depth is three.

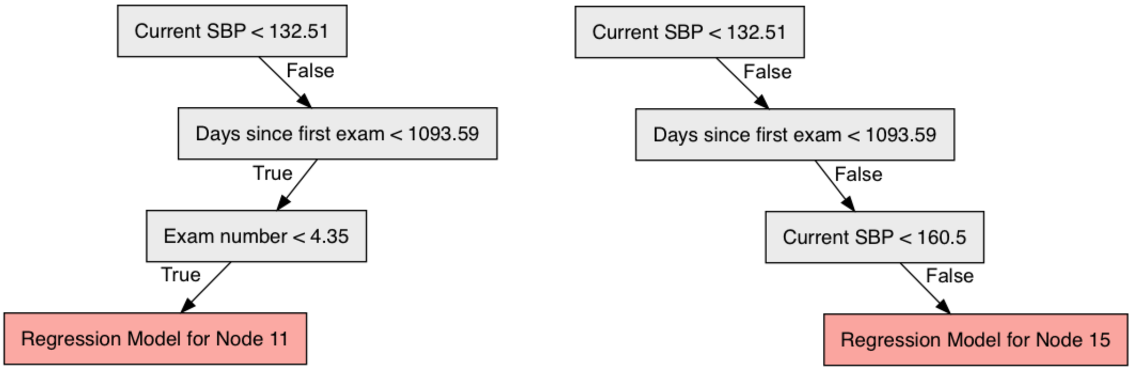

Current SBP < 132.51
False
Days since first exam < 1093.59
True
Exam number < 4.35
True
Regression Model for Node 11
Current SBP < 132.51
False
Days since first exam < 1093.59
False
Current SBP < 160.5
False
Regression Model for Node 15

Table 3.8: Example of trajectories of patient characteristics that correspond to nodes 11 and 15 in the optimal regression tree above.

	Node 11 linear models
Lasso at leaves	Model with 15 variables
SparClur: relaxation	$0.569 + 0.008x_4 - 0.337x_5 + 0.07x_6 + 0.02x_7 + 0.005x_{20}$ $+0.008x_{21} - 0.032x_{28} + 0.06x_{37} - 0.023x_{38}$
SparClur: exact	$0.558 + 0.1003x_3 + 0.005x_4 - 0.383x_5 + 0.084x_6$ $+0.015x_7 + 0.007x_{21} + 0.012x_{29} - 0.001x_{40}$
Sparse (uncoordinated)	$0.63 + 0.005x_4 - 0.29x_5 - 0.005x_{13} + 0.013x_{21}$ $+0.001x_{26} - 0.04x_{28} + 0.013x_{29} - 0.045x_{31} - 0.019x_{38}$
	Node 15 linear models
Lasso at leaves	Model with 20 variables
SparClur: relaxation	$0.594 + 0.008x_4 - 0.347x_5 + 0.019x_6 + 0.005x_7$ $-0.02x_{20} + 0.075x_{28} + 0.048x_{37} - 0.018x_{38}$
SparClur: exact	$0.66 - 0.084x_2 + 0.043x_3 + 0.001x_4 - 0.377x_5$ $-0.01x_6 + 0.013x_7 + 0.015x_{21} + 0.008x_{29} - 0.034x_{40}$
Sparse (uncoordinated)	$0.652 - 0.116x_2 + 0.047x_3 - 0.367x_5 - 0.008x_6 - 0.009x_{20}$ $+0.02x_{37} - 0.01x_{38} + 0.042x_{39} - 0.022x_{40}$

Figure 3.14: Regression models using different approaches for nodes 11 and 15 in the tree of Figure 3.13. At both nodes, SparClur gives the same support. Sparse regression gives models with the same number of features but different support. Lasso produced models that varied substantially in different leaves in terms of the number of variables and the features chosen in the model.

3.6.4 Tree for Time of Stroke Prediction

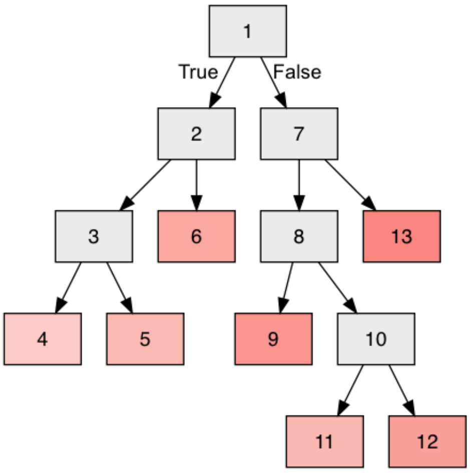


Figure 3.15: Optimal regression tree structure for the stroke case study when maximum depth is four.

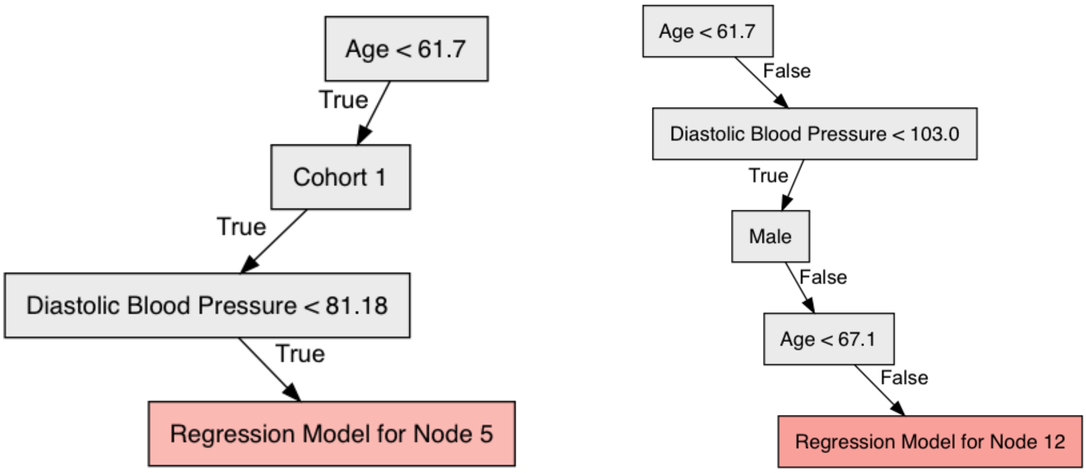

Age < 61.7
True
Cohort 1
True
Diastolic Blood Pressure < 81.18
True
Regression Model for Node 5
Age < 61.7
False
Diastolic Blood Pressure < 103.0
True
Male
False
Age < 67.1
False
Regression Model for Node 12

Table 3.9: Example of trajectories of patient characteristics that correspond to nodes 5 and 12 in the optimal regression tree above.

	Node 5 linear models
Lasso at leaves	Model with 13 variables
SparClur: relaxation	$0.644 - 0.020x_3 - 0.105x_{12} - 0.212x_{23} - 0.226x_{24}$
SparClur: exact	$0.741 - 0.228x_2 - 0.096x_{12} - 0.200x_{23} - 0.202x_{24} - 0.138x_{27}$
Sparse (uncoordinated)	$0.33 + 0.026x_2 + 0.007x_5 - 0.08x_{12} + 0.124x_{19}$
	$+0.143x_{21} + 0.121x_{26} + 0.134x_{34} + 0.130x_{35}$
	Node 12 linear models
Lasso at leaves	Model with 17 variables
SparClur: relaxation	$0.206 + 0.206x_3 - 0.045x_{12} - 0.347x_{14} - 0.008x_{23} - 0.022x_{24}$
SparClur: exact	$0.517 - 0.124x_2 - 0.105x_9 - 0.042x_{12} - 0.245x_{13}$
	$-0.284x_{14} - 0.025x_{23} - 0.002x_{24} - 0.161x_{27} - 0.038x_{33}$
Sparse (uncoordinated)	$-0.266x_9 - 0.064x_{11} - 0.013x_{12} - 0.499x_{13} + 0.803x_{22}$
	$-0.023x_{25} - 0.1x_{33} + 0.721x_{34} - 0.048x_{37}$

Figure 3.16: Regression models using different approaches for nodes 5 and 12 in the tree of Figure 3.15. At both nodes, SparClur gives the same support. Sparse regression gives models with the same number of features but different support. Lasso produced models that varied substantially in different leaves in terms of the number of variables and the features chosen in the model.

Chapter 4

Stroke Prediction from Radiology Reports

4.1 Introduction

Unstructured text in the form of radiology reports or patient-notes contains some of the most useful real-time and patient-specific information to practicing clinicians, but can be difficult to access and organize in a retrospective and scaled fashion. This often results in studies that must either eschew the wealth of information contained in these reports for analyses, or institute a labor-intensive and manual hand-labeling of pertinent features that substantially reduces sample size. These barriers deter the regular use of unstructured text in “big-data” studies, which can lead to missing important modifiers of the outcome studied. Specifically within the field of Neurology, the radiologic report of stroke is frequently diagnostic, and often considered the gold standard when determined by Magnetic Resonance Imaging [49].

The ability to extract this information quickly and accurately would provide a considerable improvement over traditional methods of identifying stroke retrospectively in large data-sets. ICD-9/10 codes for ischemic stroke are not immune to misclassification [42] and furthermore do not accurately distinguish acuity or location. An algorithm that correctly identifies diagnoses would also have substantial value in helping to triage critical reports in the clinical setting [72].

Fortunately, increased computing power has led to a resurgence of employing machine learning techniques, in which computer-algorithms trained on sufficiently large data-sets to accurately classify information better than traditional and commonly employed heuristic methods like simple logistic or linear regression. In this study, we approached the classification of radiology reports using natural language processing methods to determine three binary outcomes:

1. Whether ischemic stroke is present
2. Whether the location of stroke is in the middle cerebral artery (MCA) territory
3. Whether the stroke is acute

Our purpose was to compare different methods to determine whether automated methods could adequately classify these relevant findings in reports.

4.1.1 Existing Methodologies

Previous efforts to automate diagnoses from radiologic text have resulted in algorithms able to identify the presence or absence of breast cancer [62] and of pneumonia [28]. In [62], the authors used an almost entirely manual approach to classify reports describing breast tissue as belonging to one of four composition categories. Starting from the Breast Imaging Reporting and Data System (BI-RADS) full lexicon of key terms, they used regular expressions to find these terms in the report body and then manually reviewed the neighboring words in a certain window to identify those that were informative, iteratively increasing the window. Though their final algorithm had very high accuracy, there was little automation and no use of machine learning to identify key phrases, optimal windows, and other parameters.

In [28], the authors took a slightly more sophisticated approach by training an out-of-the-box system, ONYX, with the goal of reducing the amount of manual review needed by clinicians rather than fully automating the process. ONYX takes in raw text as input and outputs identified key concepts in the form of phrase groups [23]. Clinicians then generated a set of decision rules on this output to classify reports as consistent or inconsistent with pneumonia, or needing review, depending on what combination of concepts were present.

Only in the last few years have more advanced machine learning techniques been applied to radiology reports. A recent study by [72] sought to classify whether radiology reports contained certain findings, using multiple Natural Language Processing methods including Bag of Words, Latent Dirichlet Allocation and word embeddings. They found that simpler featurization and classification techniques perform comparably to more sophisticated deep learning approaches in identifying binary critical head CT classifiers (i.e. critical v. non critical; ischemia v. no ischemia).

4.1.2 Contributions

In this study, we conduct a thorough analysis of head CT and brain MRI reports using a completely automated end-to-end natural language processing framework for classifying presence, location, and acuity of ischemic stroke. We empirically test the combination of three text featurization techniques with seven different machine learning classifiers. In addition, we train a novel set of word vector embeddings specific to the stroke neuroradiology context, and demonstrate the validity of these embeddings using multiple quantitative and qualitative metrics.

Further, no existing methods have gone so far to specify acuity and location of ischemia. In particular, language used to characterize stroke features is diverse. For instance, “sub-acute” is a relative term, and is used to characterize strokes hours to weeks or even months old. Findings that describe characterization of hypodensities in the case of head CTs, or MRI characteristics like Apparent Diffusion Coefficient (ADC) correlation provides better clinical insight. We demonstrate that our custom embeddings, combined with deep methods like recurrent neural networks, can achieve extremely high AUC scores for all three classification tasks. We also find that far more interpretable combination of featurizations and classifiers perform comparably, showing that in this context, interpretability comes at a very low cost.

4.2 Methods

4.2.1 Study Population

In this study, approved by the Partners Human Research Committee, we collected radiology reports from a cohort of patients with ICD9 labeled diagnosis codes of ischemic stroke from 2003-2018 from the Research Patient Data Registry, a clinical repository of patient information from Massachusetts General and Brigham and Women's Hospitals. Additional eligibility criteria for study inclusion consisted of full reports of Head Computed Tomography (CT) or CT Angiography studies, Brain Magnetic Resonance Imaging or Angiography studies of patients over 18 years of age. 1359 original reports were collected and hand-labeled.

4.2.2 Text Preprocessing

Unstructured text data, like radiology reports, require a preprocessing step to remove basic non-uniformities that arise in language. On the radiology reports from our study population, we implemented the following steps:

1. We removed any reports that were incomplete, conducted at an outside institution, or lacked an "Impressions" section

2. From each report we removed header text that began before the main report, which included patient information, visit information, and details of the radiology procedure

3. Standardized language at the end of the report was removed, including names and electronic signatures of radiologists and providers

4. Reference texts included in the report body were removed, for example "==== "

5. Groups of word tokens (n-grams) that often appeared together to refer to a single entity were replaced with the n-gram without spaces, for example "middlecerebralartery"

6. All whitespace was standardized, punctuation removed, and all text was made lowercase

4.2.3 Featurization

In this section we describe the methods that we used for feature encoding of the pre-processed radiology reports. Machine learning methods require structured information as input and thus it is impossible to leverage raw text directly [35]. Thus, each radiology report needs to be represented by a vector in order to be utilized by any supervised learning algorithm [53]. Multiple approaches have been proposed towards this goal. In the medical literature, researchers have followed three different streams: a pure rule based, an ML and a hybrid approach [64]. We followed the ML driven paradigm which included the use of the state-of-the-art methods that featurize unstructured text.

Bag of words (BOW)

Bag of words is the simplest model for text featurization, disregarding context, semantic proximity and grammar. Each word included in the main corpus of the text is considered to be a distinct feature. Thus, every report can be represented with a D-dimensional vector, where D is equal to the vocabulary size found in the collective set of radiology reports. The value of each feature corresponds to the number of times a word was found in a given report. If the word was not present in the document, we assigned the value 0. For example, if the vocabulary size was 4,432, that means that each observation (report) would be encoded as 4,432 dimensional vector which would have positive values only for the words that were contained in this text. We also included 2-tuples of words to be included as a single feature in the case of common medical terminology.

Term Frequency-Inverse Document Frequency (tf-idf)

The term frequency-inverse document frequency method (tf-idf) builds upon the BOW framework, by re-weighting the document features based on the relative importance of the word in the text [52]. The weight of each word is positively correlated to the number of times a word appears in each document but it is offset by its frequency in the collection of all the training corpus. Let $f_{t,d}$ be the number of times that term t appears in report d and s_d be the number of distinct words that appear in document d. We can then define the following:

- Term Frequency: $tf(t,d) = \frac{f_{t,d}}{s_d}$
- Inverse Document Frequency: $idf(t,d) = \log\left(\frac{N}{\sum_{d=1}^{N} \mathbb{1}(TF_{t,d}>0)}\right)$

where N is the total number of documents. The latter term is a measure of how much information the word provides, i.e., if it's common or rare across all documents. Thus we can define:

$$\text{tf-idf}(t,d,N) = \text{tf}(t,d) \cdot \text{idf}(t,d)$$

For example, consider the case of a report that includes 100 different terms wherein the word stroke is encountered 5 distinct times. The term frequency (i.e., tf) for stroke is then (5 / 100) = 0.05. Assuming that we have 10 million documents and the term stroke appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the tf-idf weight is the product of these quantities: 0.05 * 4 = 0.2. This method does not take into account the sequence of words in the text neither their semantic proximity. However, it is more successful in distinguishing the importance of words in the text based on their relative frequency.

GloVe

BOW and tf-idf are techniques for converting documents into structured numeric representations. It has become increasingly common in NLP to instead use word embeddings, which represent individual words as d-dimensional vectors and have been popularized through techniques like [55]. While each dimension value in the vector does not have an absolute interpretation, word embedding vectors allow for complex pairwise comparisons between words that capture underlying semantic relationships. See Figure 4.1 for an illustration.

The current state of the art word embedding is Global Vectors for Word Representation (GloVe) [61]. GloVe takes a corpus of text and looks at how often pairs of words co-occur in some window, since these frequencies have some sort of semantic meaning. For example, the pairs "ice"-"solid" and "steam"-"gas" co-occur much more frequently than the pairs "ice"-"gas" and "steam"-"solid", with exact frequencies depending on the training corpus GloVe uses. The algorithm then learns a d-dimensional vector for each word such that their dot product, a rough measure of how close they lie in the vector space, is a positively correlated function of the words' co-occurrence probability. The dimension d is usually chosen to be between 100 and 300. These word representations then are either fed as inputs one-by-one into sequential models, or are converted into document representations by simply taking an average over words.

While pre-trained GloVe vectors are available, radiology reports and other clinical text often contain domain-specific jargon and abbreviations that do not appear in most training corpuses. To learn GloVe representations suited for our specific application, we gathered a corpus of clinically relevant texts:

- UpToDate the complete set of Neurology articles, to capture general medical language
- Stroke, Pathophysiology Diagnosis and Management, to capture disease specific language
- Yousem's Neuroradiology: The Requisites textbook, to capture neuroradiology specific language
- A sample of Partners' Healthcare radiology reports from 2010-2017, to capture radiology report specific local language

As a result, we now have available the first known neuroradiology-specific vector representations ready to be released for other applications of clinical NLP. We evaluate the results of our GloVe training in the Results section.

4.2.4 Clinical Labeling Methods

4.2.5 Supervised Learning

In this section, we outline the supervised learning methods that we used to classify the radiology reports for three outcomes of interest: (1) presence of stroke, (2) stroke MCA location, (3) acuity of stroke. We compared the performance of the most prevalent state-of-the-art ML algorithms to predict human-generated reference-standard document labels for all the feature encoding approaches outlined above.

kNN

The k-Nearest Neighbors (kNN) algorithm [24] is a supervised technique that can be applied to both classification and regression problems. In a class prediction setting, given an observation in the testing set to be classified, the algorithm searches for the k observations in the labeled

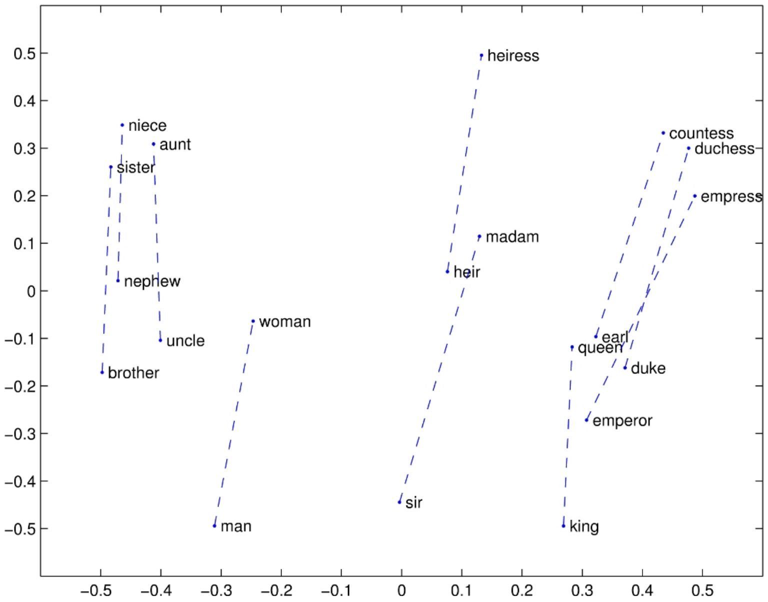


Figure 4.1: GloVe vectors projected into the two-dimensional plane showing relationships between analogous pairs

training set that are nearest in feature space, where k is a small integer. The observation is then assigned to the class to which the majority of its neighbors belong. Though the kNN algorithm is the simplest of machine learning algorithms, it is a technique with strong guarantees and often has powerful empirical performance. Its simplicity is also an advantage in terms of interpretability: we can assess why a point was predicted to fall into a certain class by looking at its neighbors and in which features they are most similar.

Logistic Regression

Logistic Regression is one of the simplest yet powerful classification algorithms used in the literature. It is similar to the linear regression function, but uses a nonlinear transformation to transform the output of the function to a probability [68]. These probabilities are compared to a threshold value to predict a binary class. If one looks at the 'Logit' function, that is the logarithm of the odds, the coefficients of the logistic response function can be interpreted in a similar fashion as those of the linear regression. To improve the regression, we have added the “l1”/Lasso regularization term to protect it against feature-wise perturbations. This will ensure a greater robustness of the regression.

CART

The Classification and Regression Trees (CART) methodology [18] trains a decision tree by splitting on variables with a greedy and top-down approach [16]. The tree is built by branching on the value of a single variable after solving a local optimisation problem that does not take into account previous splits. The tree starts with the root node and recurses on the resulting nodes. The algorithm stops when the predefined minimum number of observation per node is achieved. All the splits that do not decrease the impurity sufficiently are subsequently pruned to respect the maximum depth. CART has two major benefits: it does not assume a linear model and is interpretable as a result of the tree structure and its simple splits. To predict a class for an observation one has to follow the splits and at the end predict the most frequent outcome of the obtained leaf.

Optimal Classification Trees

OCT is an innovative advanced algorithm that trains highly accurate and interpretable classification decision-trees [6, 27]. Recently developed at MIT, this methodology leverages state-of-the art optimization techniques to construct the best decision tree for the training data in a single step. CART tree has splits that are locally-optimal, but the resulting tree could be far from optimal. OCT overcomes this problem by solving for global optimality (as opposed to traditional greedy heuristics). The model achieves therefore high accuracy and interpretability simultaneously. Contrary to most of the modern high-accuracy but opaque ML techniques (e.g. neural networks and random forests), the tree structure of the OCT method makes the model interpretable. Indeed, each node of the tree will only be split through a few high-importance variables in a straightforward manner. Accuracy on the other hand, is maintained because OCT reboots themselves with each variable and are extremely adaptive. Due to its unique combination of high predictive performance and interpretability, the OCT method has led to the creation of innovative personalized risk prediction models for the medical practice [8, 7].

A variant of the OCT is the Optimal Classification Tree with Hyperplane (OCT-H) splits [6, 27]. While each split of the OCT is based on a single variable OCT-H authorizes multivariable splits. This allows the algorithm to substantially improve its accuracy while only marginally impacting its interpretability. To better illustrated the concept of OCT and OCT-H an example is displayed in Figures 4.2,4.3.

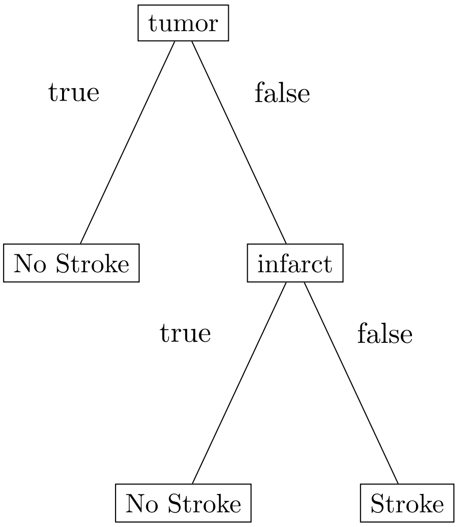


Figure 4.2: An example of an OCT model with two partition nodes and three leaf nodes.

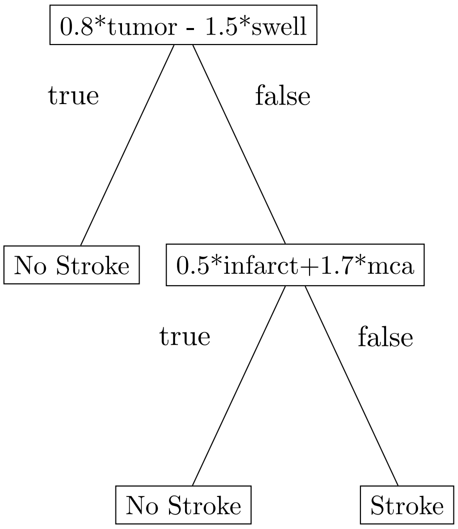


Figure 4.3: An example of an OCT-H model with two partition nodes and three leaf nodes.

Random Forest

Random forest is an ensemble machine learning method designed to improve the prediction accuracy of CART [17]. It builds a large number of CART trees in parallel and combines them into a strong learner. Each CART tree only uses a random subset of the variables and is trained with a sample of the training data. The random forest makes prediction by letting every trained tree vote and selects the outcome with the most votes. This techniques works very well in practice as the combined findings of each individual tree uncover very complex patterns. Given the number of small trees trained (in our case 500) this model's interpretability decreases significantly.

Recurrent Neural Networks

Neural networks are computational nonlinear models, whose structure resembles the one of the human brain, that are able to perform various ML tasks like classification and regression [36]. Their key components are artificial neurons or processing elements which are organized in three interconnected layers: input, hidden that may include more than one layer, and output [38]. Recurrent Neural Networks, unlike feed-forward neural networks, allow for back-propagation of the information in the model. This creates loops in the neural network architecture which act as a "memory state" for its components. This state provides the neurons with the ability to remember what have been learned so far [38]. This structure has been particularly successful in Natural Language Processing applications where the sequence of words in the text can significantly impact the overall meaning of the corpus [67]. We trained our models on a particular subclass of recurrent neural networks that utilize an efficient, gradient based method called Long Short-Term memory (LSTM) [40].

4.3 Results

4.3.1 Machine Learning Representation of Reports

In this section, we describe the results of our GloVe training process. As described above, there are many parameters to be chosen, and each combination will yield slightly different embeddings. There are two main ways of evaluating the quality of our vector representations to decide whether additional training is needed.

Word Analogies

The claim of embedding techniques like GloVe and word2vec is that the high-dimensional representations are able to capture complex structural relationships. These are often illustrated by examples of words related in a particular manner through analogies. In Figure 4.1 we see examples of word pairs, related by gender, and that their vector differences, represented by the dashed lines, are all roughly equal.

In clinical text, finding such examples is a difficult task, as medical pairs of terms that exhibit the same relation types are rare due to physiological variations in the body's various systems. In the context of neurology in particular, one can examine the word pairing "heart-carotidartery:brain-mca". In Figure 4.4 we see a projection of the 100 dimensional word vectors onto the two-dimensional plane, and observe that they exhibit similar vector differences. This structural relationship becomes even more similar if we were to project all other keywords onto the same space and observe their vector differences.

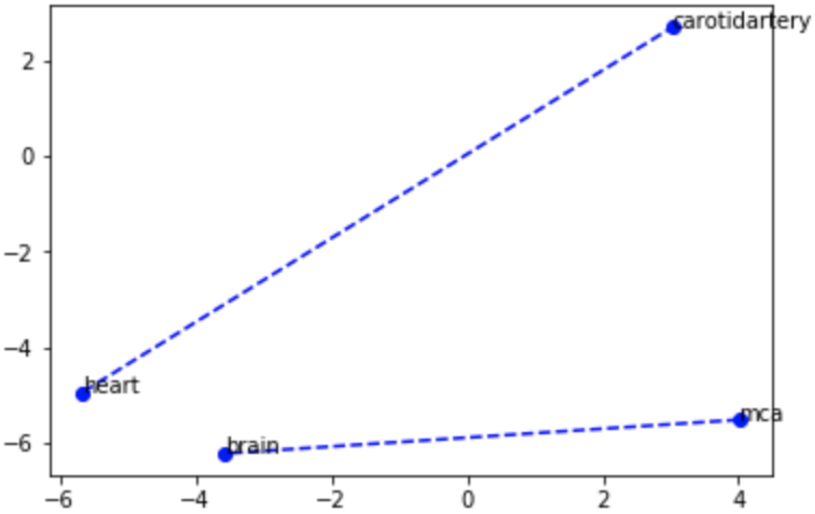


Figure 4.4: A projection of two word pairs, heart-carotidartery, and brain-mca, showing the vector differences.

Nearest Neighbors

For a given word, we can look at that word's vector representation and find its nearest neighbors. The nearest neighbors can be determined using any appropriate distance metric for vectors; here we choose cosine similarity as is often done in literature, though euclidean would also work.

To test how well our candidates overlapped with true synonyms, we used the Jaccard Index as a metric. For several important keywords, we had clinicians come up with a list of relevant words, which could be synonyms or commonly associated words like descriptors. We then generated a set of the nearest neighbors in our embeddings of the same size as the set of true synonyms.

Formally, suppose T = the set of true synonyms given by clinicians. Then we generate S = set of the $|T|$ number of nearest neighbors. Then we use as a metric the Jaccard index:

$$\frac{|S \cap T|}{|S \cup T|} \tag{4.1}$$

which looks at the number of words that are present in both sets divided by the total number of unique words in either set. A Jaccard index = 1 means our set matches the true set exactly, while a Jaccard index = 0 means our set did not generate a single one of the true synonyms.

To determine the best set of parameters and text corpuses to use in GloVe, we focused on the quality of nearest neighbors of five key words of interest: "artery","chronic","edema", "hemorrhage", and "stroke". The resulting "true" list of relevant words can be seen in Table 4.1. Then, using the vector representations generated from 100, 200, 300 dimensional vectors,

Word	Neighbors
Artery	a1, a2, a3, aca, aica, basilar, branch, cca, circulation, ica, lenticulostriate, lumen, m1, m2, m3, m4, mca, p1, p2, p3, pca, perforators, pica, sma, supraclinoid, vasculature, vert, vertebral, vessel, vessels
Chronic	atrophic, atrophy, encephalomalacia, encephalomalacic, gliosis, old, remote
Edema	compression, cytotoxic, effacement, herniation, masseffect, medialization, midlineshift, mls, subfalcine, swelling, uncal, vasogenic
Hemorrhage	basalganglia, bleed, bleeding, blood, cc, cerebellar, collection, conversion, epidural, frank, hematoma, hemorrhagicconversion, hemorrhagicinfarct, ich, intracerebralhemorrhage, intracranialhemorrhage, intraparenchymalhemorrhage, intraventricular, iph, large, lobar, micro, microhemorrhage, perimesencephalic, petechiae, petechial, pontine, putaminal, sah, small, spontaneous, subarachnoidhemorrhage, subduralhemorrhage, subdural, thalamic, transformation
Stroke	cardioembolism, cerebralinfarction, cerebrovascularaccident, cerebrovascularevent, clot, cva, emboli, embolic, embolism, hemorrhage, infarct, infarcted, infarctions, infarcts, ischemia, ischemic, tia

Table 4.1: Five keywords and the neurologist-determined ideal neighbors.

the 16 possible combinations of any number of the four corpuses, a window of 5 or 10 words, and either ignoring or including co-occurrences of words across sentences. From this process we found that the combination of parameters that has the best performance according the Jaccard index, across the five keywords, was the GloVe representation using 200 dimensions, a 10-word window, split across sentences using all four corpuses.

4.3.2 Classifiers

Our comprehensive use of machine learning methods both from the supervised and unsupervised learning literature led to the development of highly accurate and applicable models. We created classifiers that are able to detect the occurrence of stroke, its location and acuity with accuracy above 90%.

Table 4.2 provides a summary of the final classifier results. We present the out-of-sample AUC performance for each combination of unsupervised and supervised learning method. We notice that our trained word embedding using GloVe combined with RNN provide the highest performance (96.1%). However, even if this pair outperforms the rest it is not interpretable and does not provide any intuition regarding the classification outcome. Logistic regression coupled with BOW is associated with comparable results (95.9%) while also being less of a "black box" to the user. On the task of predicting stroke presence, we notice that the GloVe embedding leads to performance improvements only in the case of RNN across the three tasks compared to BOW which seems to be more applicable to other classifiers. Random Forest has equivalent performance to Logistic Regression with a slight edge over OCT-H. We observe this same pattern for the other two tasks of predicting stroke location and presence.

Average AUC	kNN	CART	OCT	OCT-H	Logistic Regression	RF	RNN
BOW	0.808	0.889	0.805	0.915	**0.951**	0.922	0.838
tf-idf	0.857	0.883	0.813	0.894	**0.939**	0.929	0.844
GloVe	0.867	0.734	0.722	0.767	0.904	0.892	**0.961**

Stroke

Average AUC	kNN	CART	OCT	OCT-H	Logistic Regression	RF	RNN
BOW	0.841	0.949	0.867	0.937	0.959	**0.960**	0.896
tf-idf	0.903	0.944	0.862	0.934	0.962	**0.965**	0.956
GloVe	0.843	0.734	0.699	0.809	0.906	0.873	**0.976**

Location

Average AUC	kNN	CART	OCT	OCT-H	Logistic Regression	RF	RNN
BOW	0.815	0.797	0.735	0.797	0.898	**0.901**	0.754
tf-idf	0.857	0.801	0.733	0.807	0.893	**0.9**	0.899
GloVe	0.842	0.73	0.719	0.82	0.881	0.866	**0.925**

Acuity

Table 4.2: Out-of-sample mean AUC across five randomized splits between the training and testing sets.

4.4 Discussion

We provide a comprehensive framework for the creation of accurate machine learning models that leverage natural language methods to identify patients with stroke, its location and acuity from radiology reports. Our work serves as a paradigm for future researchers that would like to leverage these techniques in the neurology field. We found that NLP methods perform very well at extracting featurized information from radiology reports. Predicting acuity from a report appears to be the most difficult for both machines as well as neurologist raters, while determining whether a stroke occurred in the MCA territory was most straightforward. Notably, AUCs above 90% were achieved for all three tasks using models that combine sophisticated artificial intelligence algorithms, such as GloVe or RNN.

We also present more interpretable classifiers that physicians can use in practice such as the one presented in Figure 4.5. This is the case of an OCT-H model which requires at most two separate calculations to determine whether a radiology report refers to a stroke patient or not. More specifically, if combination of phrases hemmorhagic transformation, infarctions, infarcts, insula and subacute infarctions with the corresponding coefficients 0.024, 0.213, 0.111, 0.245 is higher or equal to 0.008, the report is considered a no stroke patient. In case, the answer to the previous calculation is lower than 0.008, then the user needs to determine whether the expression 0.058*infarct+0.038*infarction+0.268*mca<0.087 is satisfied. A phrase or a word take the value of 1 if they are present in the text and zero otherwise. The tree can be broken down to independent components (splits) each of which is characterized by set of coefficients, similar to logistic regression. Thus, the model is able to identify in a non-linear but still transparent way what are the patients who have suffered a stroke outcome.

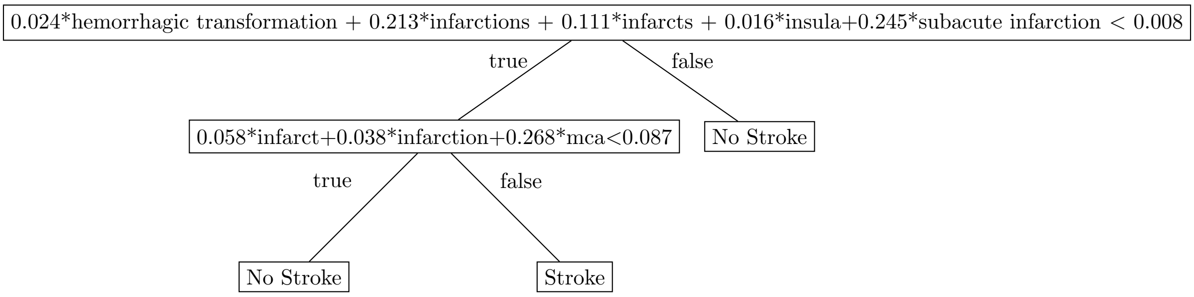

0.024*hemorrhagic transformation + 0.213*infarctions + 0.111*infarcts + 0.016*insula+0.245*subacute infarction < 0.008
true
false
0.058*infarct+0.038*infarction+0.268*mca<0.087
No Stroke
true
false
No Stroke
Stroke

Figure 4.5: An example of an OCT model with two partition nodes and three leaf nodes.

Our work is consistent with other studies that also report that simpler methods may be suitable to effectively extract unstructured text information. Zech et al found that BOW paired with lasso logistic regression had high performance with AUCs of >95% for critical head CT findings [72]. Kim et al's comparison of multiple machine learning methods to identify acute ischemic stroke on MRI and found that a single decision tree outperformed more complicated support vector machines [44]. However, for more nuanced and complex data, an embedded vector approach such as the one we used with GloVe may be increasingly valuable. We observed that it outperformed other methods by a wider margin in correctly classifying stroke acuity, particularly when paired with a neural network structure. Because RNNs account for word order, we expect that these methods will be increasingly used for accurate natural language processing of medical text data.

It is also noteworthy that of our 1359 radiographic reports, only 925 (68%) were identified as having had an ischemic stroke, a noteworthy finding in itself as our inclusion criteria consisted of patients with an ICD-9/10 billing code of stroke. Other studies have reported on the difficulties of using ICD to classify CNS disease [63, 1]. Our postulated discrepancies between ICD 9/10 codes of ischemic stroke and radiologic diagnosis include 1) Inaccuracies in billing coding; 2) Failure to report chronic known findings in radiology reports; and 3) Failure to detect previously MRI-identified strokes on head CT. Given that ICD-9 codes for ischemic stroke have reported sensitivity of up to 80%, and 75% positive predictive value [42] when validated by physician review, we feel that automated extraction from radiographic text provide more sensitive patient capture.

Results from our empirical study indicate not only that NLP methods perform well at extracting featurized information from radiology reports, but that interpretable classifiers paired with simple featurization like logistic regression with BOW can be nearly as strong as highly complex, black-box techniques like RNN paired with the uninterpretable GloVe embeddings. Given the immense overhead needed to train GloVe and deep neural networks, both in terms of time and computing resources, clinicians may want to consider the simpler machine learning approach superior when it comes to clinical implementation, and practitioners should give renewed attention to potential novel interpretable NLP techniques.

4.4.1 Limitations

There are several important limitations to our work. Similar to [72], our radiology corpus consisted of reports from two hospitals, which may affect our generalizability in other systems.

Also, the use of both computed tomography and magnetic resonance imaging reports increases heterogeneity for model development—however given that reporting language details a finite number of ways in which it describes stroke characteristics regardless of the imaging modality, we sought to test a method that could be widely applied to radiographic text.

4.5 Conclusion

Automated machine learning methods can be employed to extract diagnosis, location and acuity of stroke with high accuracy. Simpler statistical techniques like logistic regression paired with NLP methods like Bag of Words perform comparably to more sophisticated word-embedding GloVe techniques paired with deep learning classification. While these results require external validation, they provide a framework for expeditiously identifying salient stroke features from radiology text that can improve triaging high-risk scans for clinical workflow, identification of populations of interest for research and quality improvement efforts.

Chapter 5

Interpretable NLP

5.1 Introduction

Natural language processing (NLP) refers to the subfield of artificial intelligence that aims to allow machines to parse and analyze unstructured text data on par with human ability. NLP first became a problem of interest around the 1950s, spurred by work done by prominent linguists [57]. Notably, it was around this time that Chomsky put forth his theoretical analysis of language grammars [22] that soon led to the creation of a "context-free" grammar, used today to represent programming syntax. His work on grammars eventually became the basis of regular expressions, one of the most basic examples of machine-automated text analysis. It wasn't until the 1980s that NLP evolved beyond such rule-based analysis, when statistical NLP gained popularity. These probabilistic techniques focused on using simple but rigorous mathematical approximations, and large corpora of annotated text were increasingly available for machine learning algorithms to be trained. Since this reorientation over three decades ago, work in NLP has largely continued in this direction. However, motivated by particular challenges and failure modes of popular NLP techniques in specific medical applications, we propose in this chapter a set of methods that bring together statistical and rule-based NLP approaches, learning jointly from linguistic experts and from large amounts of real data, and further doing so in an interpretable way.

Typically, machine learning method for NLP rely on a corpus to train on and therefore generate a model that varies depending on the application. This makes sense for classical problems, where the samples represent different data and covariates from potentially differing populations. However, word senses in the English language have fixed meaning, and follow a highly structured pattern in all data. In this chapter, we aim to develop methods that utilize this known structure in language to their advantage. Our ideas involve utilizing WordNet, the best resource for a canonical representation of the English language, for this purpose [56]. WordNet is a human-curated graph whose nodes are synsets, groups of lemmas that share the same meaning, and whose (directed) edges are their relationships to one another. There are 16 types of word relations, with some only possible between certain types of word pairs. The main way words are related is that of hyper/hyponymy, which is the "X is a kind of Y" relation. Figure 5.1 illustrates this concept by displaying a segment of WordNet.

Popular NLP problems include document classification, document summarization, sentiment

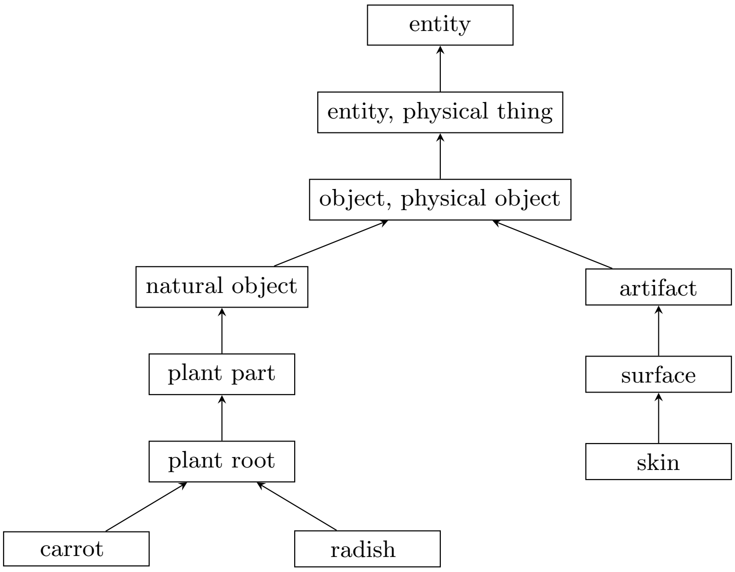


Figure 5.1: “is a” relation example in WordNet.

analysis, word sense disambiguation, named entity recognition (NER), question answering, and translation, among many others. These problems vary both in difficulty and in scope, with some as necessary obstacles on the way to tackling others. As the focus of this thesis is ultimately to improve outcomes for patients in the medical domain, we concentrate on just a few of these NLP tasks necessary to make predictions based on unstructured text data. To process a text document and develop better predictive models, our algorithm will need to go through three main steps:

1. Word sense disambiguation: figure out the meaning of each word and discriminate between different senses

2. Generate interpretable embeddings: featurize the word meanings into a structured, numeric representation

3. Calculate document distances: go from a variable collection of words to document similarity

In the following sections, we first discuss our proposed approach to each task and how it differs from current approaches. We then describe our work in progress, including some challenges and implementations so far. We finally conclude with potential next directions in which to take this work.

5.2 Proposed Methods

5.2.1 Word Sense Disambiguation

To address the problem of word sense disambiguation, which precedes the embedding problem, we can utilize WordNet. Take a sentence, for example “I went fishing for bass”. The word “bass” is one of many homographs, words with various meanings spelled identically, in the English language. On its own, it is unclear which WordNet synset the token matches to: the lowest musical range, the stringed instrument, or the fish. However, by using the surrounding context, we can make an informed guess that it refers to the fish. This method is promising not only because it approximates how humans naturally would decipher senses from words, but because it is very interpretable, as the set of candidate synsets and their distances can be easily examined.

Formally, suppose a sentence is comprised of words $w_1, \dots w_n$. Each word w_i can correspond to any number of synsets $s^i_1 \dots s^i_{m_i}$. We then choose the n corresponding synsets $s^1 \dots s^n$, one for each word, that in a sense span the smallest subset of the WordNet graph. One way of doing so is to select $s^1 \dots s^n$ that has the minimum maximum shortest path between each pair of synsets – that is, the distance between the two most unrelated words is minimized. In cases where sentences may contain two extremely distant words that dominate this metric, other alternative objectives can also be considered, for example minimizing the total sum of the shortest paths between every pair of synsets.

5.2.2 Interpretable word embedding, word relations

Word2vec and GloVe, the two state-of-the-art techniques for word embeddings, map words to a d-dimensional space, for some arbitrary d. The representations themselves have no meaning; only the vector relationships between pairs of words in the universe do. The same vector representations could be translated or scaled by a fixed amount without changing their application. For example, the vector operation $paris - france$ should result in a vector that is "closest" by some similarity measure, like cosine similarity or Euclidean distance, as that of $berlin - germany$.

By using inherent relations between words, as represented by WordNet, we aim to create interpretable word embeddings, where each dimension represents the level of the word in the hierarchy of that relation type. For example, imagine our universe consists of the words {animal, mammal, dog, poodle}. By WordNet's hierarchy, each word is a *hyponym* and child of its previous word (a poodle is a type-of dog, a dog is a type-of mammal). Then, we want

$$\begin{aligned} animal &= (a_1, \dots, a_{16}) \\ mammal &= (m_1, \dots, m_{16}) \\ dog &= (d_1, \dots, d_{16}) \\ poodle &= (p_1, \dots, p_{16}) \end{aligned}$$

such that

$$\begin{aligned} a_1 &> m_1 \\ m_1 &> d_1 \\ d_1 &> p_1 \end{aligned}$$

where we have let the 1st dimension correspond to the "is a kind of" relationship. An alternative formulation could be specifying a, m, d, p such that

$$\begin{aligned} a_1 - m_1 &= \alpha_a - \alpha_m \\ m_1 - d_1 &= \alpha_m - \alpha_d \\ d_1 - p_1 &= \alpha_d - \alpha_p \end{aligned}$$

where α_a is the rank of a in our corpus of words, according to WordNet. In this case, $\alpha_a = 1, \alpha_m = 2, \alpha_d = 3, \alpha_p = 4$.

In this way, the vector embeddings we construct will both be interpretable, and capture more exact relationships between word pairs. So, $animal - dog$ will result in a vector whose 1st entry has the largest absolute magnitude, and $fruit - apple$ will be nearly identical.

Another approach to interpretable word embeddings is to instead of having the dimensions represent word relationships, have dimensions corresponding to different clusters the synsets belong to. Since the WordNet graph is hierarchical for certain word relations, we can find key nodes or clusters of nodes that other synsets filter up to, and vectorize words with binary markers which they belong to or distance to the node.

5.2.3 Document Distance with Ordering

Documents are commonly represented as a bag-of-words (BOW), which as a simply word frequency count fails to capture word ordering which can be key in certain domains. The same is true of term frequency-inverse document frequency (tf-idf) representation, which is essentially a re-weighted version of a BOW representation. Even with more sophisticated featurization techniques like word embeddings, a common approach is to simply sum or average the vector representations of each word to get a document vector of the same dimension, which again ignores the structure of the text.

Only sequence models such as recurrent neural networks take into account the ordering of the words, but as they are deep, uninterpretable methods, we aim to construct an alternative approach using document similarity. Combined with a classifier like k-nearest neighbors (kNN), which has a certificate of interpretability, document similarity can capture the effects of ordering when going from word to document representations.

[45] recently introduced the idea of treating document distances like the earth mover's distance, a well-known problem in the optimization and transportation space. Their metric, called the Word Mover's Distance (WMD) finds the minimum total distance every word in one document A must "travel" to reach some other word in document B, where the distance traveled by the word is their vector similarity. While this captures pairwise word similarity semantically, we propose to also account for word ordering by incorporating how "far" they need to move that word, which we will refer to as the lexical distance. For example, "acute stroke, no cancer" should have a further distance from "acute cancer, no stroke" and instead should be closer to "acute infarct, no cancer". Under WMD, the former two would have a distance of exactly zero.

Suppose we have two documents a and b, of length m and n respectively. Let c_{ij} be the cosine similarity between the vector representations of word i in sentence a and word j in sentence b. Let ℓ_{ij} be the difference in locations of word i and word j in their relative documents. Then, we aim to find a T_{ij} flow from sentence a to sentence b that minimizes the cost. There are at least three different ways we can formulate this problem, all derived from a language-based perspective of how humans might approach the problem.

Short-to-long distance

In this scenario we have |sentence a| $\leq$ |sentence b| and let each word have weight/supply 1. The document distance in this case is simply the minimum cost of moving the shorter sentence to the longer sentence.

$$\begin{aligned} \min_T \; & \sum_{i=1}^{m}\sum_{j=1}^{n} T_{ij}(\theta c_{ij} + (1-\theta)\ell_{ij}) \\ \text{s.t.} & \sum_i T_{ij} \leq 1 \quad \forall j \\ & \sum_j T_{ij} = 1 \quad \forall i \\ & T_{ij} \geq 0 \quad \forall i, j \end{aligned} \tag{5.1}$$

where θ is a hyperparameter chosen based on the specific application to balance the tradeoff between semantic distance and lexical distance between words.

Equal weighting

The first formulation assumes that longer sentences may have just fillers and disregards those. Often, there can be more verbose texts that convey the same meaning as shorter texts. In a second formulation, we weight each word in a sentence uniformly and find the minimum cost of moving all of sentence a to sentence b.

$$\begin{aligned} \min_T \; & \sum_{i=1}^{m}\sum_{j=1}^{n} T_{ij}(\theta c_{ij} + (1-\theta)\ell_{ij}) \\ \text{s.t.} & \sum_i T_{ij} = \frac{1}{|\text{sentence b}|} \quad \forall j \\ & \sum_j T_{ij} = \frac{1}{|\text{sentence a}|} \quad \forall i \\ & T_{ij} \geq 0 \quad \forall i, j \end{aligned} \tag{5.2}$$

Variable weighting

Even the second formulation has a shortcoming - it assumes every word is uniformly important in lending meaning to a sentence. If we are trying to classify whether a patient has a medical condition, we may want to place more weight on relevant medical terms and less on other words where small variations may not matter. In this formulation, we let the model choose freely the weights of the words, but if there is an exact match in words, the model would choose to place all weight in that word and set everything else to 0. To prevent this, we add a penalty to the

L2 norm of the weights in our objective. Again, θ and λ are hyperparameters to be tuned on the data.

$$
\begin{aligned}
\min_{T,w^a,w^b} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} T_{ij}(\theta c_{ij} + (1-\theta)\ell_{ij}) + \lambda(||w^a||_2^2 + ||w^b||_2^2) \\
\text{s.t.} \quad & \sum_i T_{ij} = w_j^b \quad \forall j \\
& \sum_j T_{ij} = w_i^a \quad \forall i \\
& \sum_i w_i^a = 1 \\
& \sum_j w_j^b = 1 \\
& T_{ij}, w_i^a, w_j^b \geq 0 \quad \forall i, j
\end{aligned}
\tag{5.3}
$$

5.3 Implementation and Challenges

5.3.1 Interpretable Embeddings

A number of outstanding questions remain in implementing the formulation. A few of these include:

1. How many pairs do we need in the constraint? For example, if a is higher in the hierarchy than b which is higher than c and so on, do we need:
   - only consecutive constraints based on word hierarchy? $a > b, b > c, c > d, \ldots$
   - all possible pairings between words? $a > b, a > c, a > d, \ldots$
2. How do we decrease the universe we work with? Since WordNet contains over 100,000 synsets, we do not need to solve the optimization problem for all the words in the English language just to get embeddings for the words in our corpus, especially if we want it to run efficiently. We could perhaps take an on-the-fly approach, where we only add constraints and objective terms for the words in our sample, and if any unseen word appears out-of-sample, we could use a heuristically use its position in WordNet to generate a new vector embedding.

5.3.2 Word Mover Distance

Currently we have implemented and tested our three approaches against the standard Word Mover's Distance as a baseline. We used a dataset of about 250 short impressions from a radiology dataset for the presence of ischemic stroke and used GloVe vector representations trained on a custom corpus as described in Chapter 4. For every pair of impression sentences in the dataset, we calculated our metrics as described in the previous section. After creating these pairwise-sentence distances, we use the distance matrix in a kNN classification model, the results of which are reported in Table 5.1. We note that our empirical results are highly dependent/sensitive to parameter tuning.

Model	**AUC**
Model 1	0.660
Model 2	0.821
Model 3	0.652
WMD	0.785

Table 5.1: Mean out-of-sample AUCs across five splits of our radiology reports.

We note that our second model performs quite well, even outperforming the WMD baseline. However, our third model, which allows for variable weighting and should be at least as strong as our second model, surprisingly performs the worst, even worse than the simple first model which discards parts of a longer text. We hypothesize this may be the result of overfitting to the training set, or still placing too much weight on overlapping words that may not have medical significance in an attempt to minimize the distance.

To continue improving model 3, we can try:

- adding a proximal constraint where we penalize divergence from the uniform weighting
- using not the cosine similarity of the GloVe vectors, but some distance between word synsets based on WordNet

Since the document distance generated is only a step on the way to the end goal of classifying documents correctly, there is also potential to have a supervised version of the problem where some empirical error of the predicted outcome is minimized. The document distance would then be informed by the specific application and likely result in much higher classification performance.

5.4 Conclusion

Throughout this thesis, we have demonstrated the need for interpretable machine learning methods as well as their promise. Specifically in the field of NLP, much progress can be made on this front. As the preliminary work in this chapter shows, there exist optimization-based methods that can capture linguistic properties of textual data in a way that is complementary to, rather than opposes, mathematically-based algorithms that learn from data. Further refinement of the proposed ideas in this chapter could potentially result in improved NLP techniques that approach the performance of deep black-box methods while maintaining interpretability.

Chapter 6

Summary

In this thesis, we present a collection of works using interpretable machine learning approaches to predict characteristics or outcomes of stroke at various hospitals.

At Hartford HealthCare, we demonstrate the power of Optimal Classification Trees as a technique for predicting in-hospital mortality and mortality within a year from discharge, as well as the more difficult task of recurrence of stroke within a year, of patients admitted for ischemic stroke, hemorrhagic stroke, or TIA. We show that it not only attains good out-of-sample AUCs, either on par with or outperforming other machine learning techniques and outperforming stroke mortality risk scores, but that the resulting trees are easy to interpret and align with clinical understanding of stroke. The resultant risk calculators from these algorithms are an adaptive and interactive way for patients and doctors to understand the non-linearity of stroke risk factors for mortality.

Our work at Hartford HealthCare showed clear evidence that health outcomes manifest themselves differently among different parts of the population. In classification tasks, this is captured by how Optimal Classification Trees splits among certain demographics to arrive at a prediction or risk score at the leaves. We were inspired to devise a similar approach for regression problems, where some real-valued outcome depends on a set of features that may differ for various subpopulations. We present SparClur as a method that achieves this generalizability on the feature set while still ensuring state of the art accuracy. Using an optimal tree to segment the population, we then train a separate regression model at each of the leaves but coordinate the models so they share the same support. We demonstrate on synthetic data that the method is correct and achieves stronger results, and show on real-world medical datasets that the increased interpretability of the method comes at a very low cost to the accuracy.

Though many machine learning algorithms have been applied to structured data, the majority of available data in healthcare is in unstructured form. We examine the task of predicting presence, location, and acuity of ischemic stroke in patients at Partners HealthCare from the raw text of radiology reports. Working with text data comes with its own set of challenges, and we give a rigorous, comprehensive overview of how a combination of popular ML classifiers combined with NLP featurizations perform on a sample of MR and CT scan reports. For each of our three tasks, deep neural networks (LSTMs) combined with GloVe embeddings had the highest performance, but methods like logistic regression combined with a BOW document representation performed comparably. This suggests that when it comes to prediction from radiology texts, more interpretable and simpler methods may actually be preferable to deep black-box methods that require endless parameter tuning as well as complex vector training, especially when time and resources are limited.

Finally, we conclude by discussing the need for interpretable NLP techniques, and presenting initial work in this direction. This includes language-based approaches for the NLP problems of word sense disambiguation, word representation, and classification. We introduce WordNet and show promising preliminary results indicating that optimization-based approaches to these problems may be more powerful than traditional approaches on both regular and medical text.

Altogether, this thesis presents a comprehensive overview of interpretable methods in predicting stroke outcomes, from applications in real-world problems faced by actual clinicians to demonstrating the importance and promise of interpretable machine learning. We conclude that the issues of model accuracy and model interpretability in healthcare should not be tackled with a focus on one and at a cost to the other, but instead in tandem, and present promising novel methodologies to that end.

Bibliography

[1] Marzia Baldereschi, Daniela Balzi, Valeria Di Fabrizio, Lucia De Vito, Renzo Ricci, Paola D'Onofrio, Antonio Di Carlo, Maria Teresa Mechi, Francesco Bellomo, and Domenico Inzitari. Administrative data underestimate acute ischemic stroke events and thrombolysis treatments: data from a multicenter validation survey in italy. *PloS One*, 13(3):e0193776, 2018.

[2] Miguel Viana Baptista, Guy van Melle, and Julien Bogousslavsky. Prediction of in-hospital mortality after first-ever stroke: the lausanne stroke registry. *Journal of the Neurological Sciences*, 166(2):107–114, 1999.

[3] Emelia J Benjamin, Daniel Levy, Sonya M Vaziri, Ralph B D'agostino, Albert J Belanger, and Philip A Wolf. Independent risk factors for atrial fibrillation in a population-based cohort: the framingham heart study. *JAMA*, 271(11):840–844, 1994.

[4] Dimitris Bertsimas, Christian Cadisch, Emma Chesley, Alberts Mark, Amre Nouh, and Agni Orfanoudaki. Development and validation of a novel, non-linear stroke risk assessment tool. *Submitted in Stroke*, 2018.

[5] Dimitris Bertsimas and Martin S. Copenhaver. Characterization of the equivalence of robustification and regularization in linear and matrix regression. *European Journal of Operational Research*, 270(3):931–942, 2018.

[6] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

[7] Dimitris Bertsimas, Jack Dunn, Colin Pawlowski, John Silberholz, Alexander Weinstein, Ying Daisy Zhuo, Eddy Chen, and Aymen A Elfiky. Applied informatics decision support tool for mortality predictions in patients with cancer. *JCO Clinical Cancer Informatics*, 2:1–11, 2018.

[8] Dimitris Bertsimas, Jack Dunn, George C Velmahos, and Haytham MA Kaafarani. Surgical risk is not linear: Derivation and validation of a novel, user-friendly, and machine-learning-based predictive optimal trees in emergency surgery risk (potter) calculator. *Annals of Surgery*, 268(4):574–583, 2018.

[9] Dimitris Bertsimas, Nathan Kallus, Alexander M Weinstein, and Ying Daisy Zhuo. Personalized diabetes management using electronic medical records. *Diabetes Care*, 40(2):210–217, 2017.

[10] Dimitris Bertsimas, Angela King, Rahul Mazumder, et al. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

[11] Dimitris Bertsimas, Jean Pauphilet, and Bart Van Parys. Sparse classification and phase transitions: A discrete optimization perspective. *arXiv preprint arXiv:1710.01352*, 2017.

[12] Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. From predictive methods to missing data imputation: an optimization approach. *The Journal of Machine Learning Research*, 18(1):7133–7171, 2017.

[13] Dimitris Bertsimas and Bart Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *arXiv preprint arXiv:1709.10029*, 2017.

[14] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[15] Karl Y Bilimoria, Yaoming Liu, Jennifer L Paruch, Lynn Zhou, Thomas E Kmiecik, Clifford Y Ko, and Mark E Cohen. Development and evaluation of the universal acs nsqip surgical risk calculator: a decision aid and informed consent tool for patients and surgeons. *Journal of the American College of Surgeons*, 217(5):833–842, 2013.

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

[17] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[18] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

[19] Lloyd E Chambless, Gerardo Heiss, Eyal Shahar, Mary Jo Earp, and James Toole. Prediction of ischemic stroke risk in the atherosclerosis risk in communities study. *American Journal of Epidemiology*, 160(3):259–269, 2004.

[20] Jonathan H Chen and Steven M Asch. Machine learning and prediction in medicine—beyond the peak of inflated expectations. *The New England Journal of Medicine*, 376(26):2507, 2017.

[21] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SigKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

[22] Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.

[23] Lee M Christensen, Henk Harkema, Peter J Haug, Jeannie Y Irwin, and Wendy W Chapman. Onyx: a system for the semantic analysis of clinical text. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 19–27. Association for Computational Linguistics, 2009.

[24] Thomas M Cover, Peter E Hart, et al. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[25] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[26] Bart M Demaerschalk, Ha-Mill Hwang, and Grace Leung. Us cost burden of ischemic stroke: a systematic literature review. *The American Journal of Managed Care*, 16(7):525–533, 2010.

[27] Dimitris Bertsimas and Jack Dunn. *Machine Learning under a Modern Optimization Lens.* Dynamic Ideas, Belmont, 2018. to appear.

[28] Sascha Dublin, Eric Baldwin, Rod L Walker, Lee M Christensen, Peter J Haug, Michael L Jacksonand, Jennifer C Nelson, Jeffrey Ferraro, David Carrell, and Wendy W Chapman. Natural language processing to identify pneumonia from radiology reports. *Pharmacoepidemiology and Drug Safety*, 22(8):834–841, 2013.

[29] Jack Dunn. *Optimal Trees for Prediction and Prescription.* PhD thesis, Massachusetts Institute of Technology, 2018.

[30] Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[31] B Farrell, J Godwin, S Richards, and C Warlow. The united kingdom transient ischaemic attack (uk-tia) aspirin trial: final results. *Journal of Neurology, Neurosurgery & Psychiatry*, 54(12):1044–1054, 1991.

[32] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4), 2009.

[33] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[34] Brian F Gage, Amy D Waterman, William Shannon, Michael Boechler, Michael W Rich, and Martha J Radford. Validation of clinical classification schemes for predicting stroke: results from the national registry of atrial fibrillation. *JAMA*, 285(22):2864–2870, 2001.

[35] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10:1–309, 04 2017.

[36] Kevin Gurney. *An Introduction to Neural Networks.* Taylor & Francis, Inc., Bristol, PA, USA, 1997.

[37] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[38] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[39] J Claude Hemphill, David C Bonovich, Lavrentios Besmertis, Geoffrey T Manley, and S Claiborne Johnston. The ich score. *Stroke*, 32(4):891–897, 2001.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[41] Joseph G Ibrahim, Haitao Chu, and Ming-Hui Chen. Missing data in clinical studies: issues and methods. *Journal of Clinical Oncology*, 30(26):3297, 2012.

[42] Sydney A Jones, Rebecca F Gottesman, Eyal Shahar, Lisa Wruck, and Wayne D Rosamond. Validity of hospital discharge diagnosis codes for stroke: the atherosclerosis risk in communities study. *Stroke*, 45(11):3219–3225, 2014.

[43] Julia Statistics. Julia wrapper for fitting lasso/elasticnet glm models using glmnet. `https://github.com/JuliaStats/GLMNet.jl`, 2018. [Online; accessed 2018-06-08].

[44] Chulho Kim, Vivienne Zhu, Jihad Obeid, and Leslie Lenert. Natural language processing and machine learning algorithm to identify brain mri reports with acute ischemic stroke. *PloS One*, 14(2):e0212778, 2019.

[45] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[46] Karim S Ladha, Kevin Zhao, Sadeq A Quraishi, Tobias Kurth, Matthias Eikermann, Haytham MA Kaafarani, Eric N Klein, Raghu Seethala, and Jarone Lee. The deyo-charlson and elixhauser-van walraven comorbidity indices as predictors of mortality in critically ill patients. *BMJ Open*, 5(9):e008990, 2015.

[47] Susanna C Larsson, Alice Wallin, Alicja Wolk, and Hugh S Markus. Differing association of alcohol consumption with different stroke types: a systematic review and meta-analysis. *BMC Medicine*, 14(1):178, 2016.

[48] Andreas Laupacis, Nandita Sekar, et al. Clinical prediction rules: a review and suggested modifications of methodological standards. *JAMA*, 277(6):488–494, 1997.

[49] David S Liebeskind and Andrei V Alexandrov. Advanced multimodal ct/mri approaches to hyperacute stroke diagnosis, treatment, and monitoring. *Annals of the New York Academy of Sciences*, 1268:1, 2012.

[50] Eric M Liotta, Mandeep Singh, Adam R Kosteva, Jennifer L Beaumont, James C Guth, Rebecca M Bauer, Shyam Prabhakaran, Neil F Rosenberg, Matthew B Maas, and Andrew M Naidech. Predictors of 30 day readmission after intracerebral hemorrhage: a single-center approach for identifying potentially modifiable associations with readmission. *Critical Care Medicine*, 41(12), 2013.

[51] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[52] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[53] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[54] Teri A Manolio, Richard A Kronmal, Gregory L Burke, Daniel H O'Leary, and Thomas R Price. Short-term predictors of incident stroke in older adults: the cardiovascular health study. *Stroke*, 27(9):1479–1486, 1996.

[55] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[56] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[57] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[58] Amre M Nouh, Lauren McCormick, Janhavi Modak, Gilbert Fortunato, and Ilene Staff. High mortality among 30-day readmission after stroke: Predictors and etiologies of readmission. *Frontiers in Neurology*, 8:632, 2017.

[59] The Office of the National Coordinator for Health Information Technology. Office-based physician electronic health record adoption. `https://dashboard.healthit.gov/quickstats/pages/physician-ehr-adoption-trends.php`, 2019. [Online; accessed 2019-08-03].

[60] Jonas Bjerring Olesen, Christian Torp-Pedersen, Morten Lock Hansen, and Gregory YH Lip. The value of the cha2ds2-vasc score for refining stroke risk stratification in patients with atrial fibrillation with a chads2 score 0–1: a nationwide cohort study. *Thrombosis and Haemostasis*, 108(06):1172–1179, 2012.

[61] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[62] Bethany Percha, Houssam Nassif, Jafi Lipso, Elizabeth Burnside, and Daniel Rubin. Automatic classification of mammography reports by bi-rads breast tissue composition class. *Journal of the American Medical Informatics Association*, 19(5):913–916, 2012.

[63] Paisith Piriyawat, Miriam Šmajsová, Melinda A Smith, Sanjay Pallegar, Areej Al-Wabil, Nelda M Garcia, Jan M Risser, Lemuel A Moyé, and Lewis B Morgenstern. Comparison of active and passive surveillance for cerebrovascular disease: The brain attack surveillance

in corpus christi (basic) project. *American Journal of Epidemiology*, 156(11):1062–1069, 2002.

[64] Ewoud Pons, Loes M. M. Braun, M. G. Myriam Hunink, and Jan A. Kors. Natural language processing in radiology: A systematic review. *Radiology*, 279(2):329–343, 2016. PMID: 27089187.

[65] Naveen F Sangji, Jordan D Bohnen, Elie P Ramly, George C Velmahos, David C Chang, and Haytham MA Kaafarani. Derivation and validation of a novel physiological emergency surgery acuity score (pesas). *World Journal of Surgery*, 41(7):1782–1789, 2017.

[66] Edward H Shortliffe. Mycin: a rule-based computer program for advising physicians regarding antimicrobial therapy selection. Technical report, Stanford University Dept of Computer Science, 1974.

[67] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 129–136, 2011.

[68] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[69] Andrey Nikolayevich Tikhonov. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198, 1943.

[70] Yang Wang, Lynette L-Y Lim, Richard F Heller, Janet Fisher, and Christopher R Levi. A prediction model of 1-year mortality for acute ischemic stroke patients. *Archives of physical medicine and rehabilitation*, 84(7):1006–1011, 2003.

[71] Philip A Wolf, Ralph B D'Agostino, Albert J Belanger, and William B Kannel. Probability of stroke: a risk profile from the framingham study. *Stroke*, 22(3):312–318, 1991.

[72] John Zech, Margaret Pain, Joseph Titano, Marcus Badgeley, Javin Schefflein, Andres Su, Anthony Costa, Joshua Bederson, Joseph Lehar, and Eric Karl Oermann. Natural language–based machine learning models for the annotation of clinical radiology reports. *Radiology*, 287(2):570–580, 2018.