Addressing Bias and Subjectivity in Machine Learning

A dissertation

submitted by

Yijun Zhao

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

TUFTS UNIVERSITY

May 2017

ADVISOR: Prof. Carla E. Brodley

This thesis is dedicated to my family for their unconditional support.

Acknowledgments

First and foremost, I would like to express my sincerest gratitude to my advisor, Prof. Carla E. Brodley. Without her guidance and mentorship, I would not have been able to reach the research frontier explored in this work. Her expertise and contagious energy repeatedly helped me step out of my comfort zone and acquire techniques that I thought were beyond my limits. The high standards and academic rigor she insisted upon throughout this fulfilling journey awarded me with professional benefits I will enjoy for the rest of my career. I am very grateful for the generous time and assistance of my thesis committee members, Prof. Roni Khardon, Prof. Ben Hescott, Prof. Jennifer G. Dy, and Prof. Shuchin Aeron. Their insightful comments and valuable advice were indispensable to the completion of this thesis.

I would like to express my gratitude to the staff members at Tufts University. I thank George Preble, Patrick Hynes, Michael Bauer, and Stephen Gemme for their prompt and patient help with my linux, desktop and laptop issues. I thank Donna Cirelli, Sarah Richmond, Megan Monaghan, Jeannine Vangelist, and Gail Fitzgerald for their support during my graduate studies.

I would like to thank the Henry LUCE foundation for its Clare Boothe Luce (CBL) fellowship which funded my research from 2013 to 2015 and ensured a smooth progression in my studies.

I greatly appreciate the support, encouragement and friendship from my fellow doctorate students. I thank Bilal Ahmed for many fruitful discussions and Mike Shan for studying with me while preparing for our qualifying exams.

Finally, I am deeply indebted to my family whose unabating support has made the completion of this thesis a reality. In particular, I would like to thank my son, Tim, who followed a parallel educational path, and who kept me company while

struggling with his own high school challenges; our mutual encouragement (as well as friendly GPA and grade competitions) shortened this long journey and brought us both to the last milestones of our trips.

Yijun Zhao

TUFTS UNIVERSITY

May 2017

Addressing Bias and Subjectivity in Machine Learning

Yijun Zhao

ADVISOR: Prof. Carla E. Brodley

The success of supervised machine learning algorithms rests on the assumption that data are drawn from the same underlying distribution. However, this assumption is often violated in real world applications where collected data involves human judgement. The contribution of this thesis is a collection of approaches that address bias and subjectivity in real world data. We illustrate our work through three applications: predicting disease progression in Multiple Sclerosis (MS) patients, detecting epileptogenic lesions in focal cortical dysplasia (FCD) patients and selecting the best performing students in the graduate admission process. In each of these applications, subjectivity and/or bias manifest themselves in different ways.

We present a total of four models each of which takes on unique challenges associated with each task. In the MS research, we introduce two models to estimate the prognosis for MS patients while addressing the patient bias and physician subjectivity in the data: a classification model that predicts the MS disease progression ('high' versus 'low'), and a regression model that forecasts the actual MS severity scores. In the epilepsy research, we present a model that addresses the paucity of features from MRI images and biases in the data originated from inter-patient variability. Lastly, in the third application, we introduce a new variant of SVM that exploits both labeled and unlabeled data and addresses the subjectivity arising from the admission process.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the last decade, technical developments and the explosion in the availability of data have created a fertile environment for machine learning. The adoption of data-intensive, evidence-based, decision making algorithms is pervasive in our daily lives. For example, companies such as Amazon and Google have been applying machine learning methods to customize recommendations, detect spam e-mails and optimize ad efficiency. In the medical field, machine learning has been deployed to assist in predicting the course of incurable diseases based on the large number of empirical cases. Within artificial intelligence (AI), machine learning/deep learning has emerged as the method of choice for developing computer vision, speech recognition, robot control and many other applications.

Conceptually, machine learning can be viewed as discovering the underlying pattern in a large collection of data (i.e., training examples), guided by various learning algorithms. The effectiveness of this process relies on the assumption that the collected data are drawn independently from the same distribution. For example, support vector machine (SVM) [20], back-propagation for Neural Networks [24], and many other common algorithms implicitly make this assumption as part of their derivation. Nevertheless, practitioners applying machine learning to real world data often find themselves in a common predicament: data are collected from heterogeneous sources and as a consequence have different underlying distributions. In this thesis, we further classify this heterogeneity into two categories: bias and

subjectivity. "Bias" refers to situations in which data may form groups due to some intrinsic properties. For example, in the task of estimating survival rates for cancer patients, the predictors may be inherently different (but not completely independent) for patients of different age or demographics. Thus, training a model using the data from all patients may lead to poor performance. Another example is in the task of automatic spam filtering where the distribution over spam or not-spam emails differs across different user groups.

We define "subjectivity" to be the factor affecting the distribution of data based on people's opinions or life experiences. For example, doctors' individualized interpretations of patients' performance on their cognitive tests may result in a patient's neurological function being rated differently by different doctors. Other examples include movie ratings collected from customers and interviewers' feedback on candidates. Bias and subjectivity are common in real world data and their existence violates the fundamental assumption of machine learning algorithms, i.e., that data points are i.i.d. As a result, learning that uses the entire dataset without considering these factors can lead to unsatisfactory performance.

The contribution of this thesis is a collection of approaches that address bias and subjectivity in real world data. We illustrate our work through three applications: predicting disease progression in Multiple Sclerosis (MS) patients, detecting epileptogenic lesions in focal cortical dysplasia (FCD) patients and selecting the best performing students in the graduate admission process. In each of these applications, subjectivity and/or bias manifest themselves in different ways. In the first application, patients' longitudinal data may contain subjective judgement from different physicians. The bias in the second application stems from the inter-patient variability, i.e., the morphometry of the human brain such as its thickness, curvature and the overall structure are affected by different factors such as age, gender, handedness, etc. [73]. In the third application, the subjectivity in the data arises from the change of committee memberships across academic years and their potentially subjective decisions on student applications. Furthermore, each of these applications exhibits idiosyncratic characteristics that entail different treatment in designing an

effective model. For example, the second application offers very limited features extracted from the original MRI images and we employ restricted Boltzmann machines (RBMs) [76, 94] to address this issue, whereas the third application contains both labeled and unlabeled training instances and we design our model under the semi-supervised learning framework. In Sections 1.1, 1.2, 1.3, we introduce each application, its unique machine learning challenges and we outline our approach to handling each challenge.

1.1 Predicting Disease Progression in Multiple Sclerosis Patients

Multiple sclerosis (MS) affects approximately 400,000 people in the United States and 2.5 million worldwide [82]. The majority of cases present with relapses involving neurological deficits such as vision blurring or loss, weakness, numbness, imbalance or cognitive deficits. In the early stages of disease, relapses generally improve or remit, however later on, there may be residual deficits due to relapses [56]. In addition, there is a superimposed process of progressive disability that results in permanent deficits [89]. Cumulative disability is typically measured using a zero to ten Expanded Disability Status Scale (EDSS) [46, 61] score, in which zero is normal and six corresponds to walking with a cane. There is considerable variability in MS disease course with some patients demonstrating a benign course of disease, while others progress to an EDSS of six within five years [40, 31]. All currently approved MS therapies primarily target relapses, and have limited effects on overall disability progression [62, 34, 41]. However there is increasing evidence suggesting that early and more aggressive treatment targeting relapses may delay or prevent long-term accumulation of disability [44, 42], and this effect must be balanced with the potential increased side effects associated with more potent therapies. The identification of patients who are more likely to accrue disability would allow clinicians to institute more rigorous monitoring procedures, and potentially to initiate more potent therapies early in the course of disease.

In our research, we work closely with doctors from Harvard Medical School and Brigham and Women's Hospital (BWH) in Boston, Massachusetts to predict the disease course of MS patients at the fifth year mark using their first two year's longitudinal data. Our clinical data are collected as part of the CLIMB (Comprehensive Longitudinal Investigation of Multiple Sclerosis at Brigham and Women's Hospital) study [30], which is a large-scale, long-term study of patients with MS. It is designed to investigate the course of the disease in the current era of treatment. The main goals of the study are to identify predictors of future disease course when patients are at the beginning of their illness and determine the effects of treatment on disease progression and accumulation of disability.

Our prediction task faces two unique data analysis challenges because the data are collected from multiple organizations/physicians. First, a patient's data may come from one doctor on the first visit and different doctors on subsequent visits and, thus, it can be difficult to form an accurate predictor of disease outcome. In particular, some physicians may be biased in one direction, scoring each patient as more severe than would other physicians, while others may be biased in the opposite direction. Second, patients have their preferences in their selection of physicians [26]. This means we cannot solve physician subjectivity by normalizing the data with respect to physician. We call this second complication "patient bias." One of our proposed solutions is a new transfer learning approach to handle both physician subjectivity and patient bias in their clinical data. Transfer learning [64] is a technique to improve performance leveraging related knowledge. In our framework, we partition the entire dataset into primary dataset (instances from the doctor) and auxiliary dataset (instances from other doctors). We build a single predictor for each physician making use of their dataset and the data of the auxiliary set. The details of our model and its efficacy compared to other existing approaches are described in Chapter 2.

While our transfer learning approach provides a classification model which predicts an MS patient's disability status ("high" versus "low") in five years, forecasting the patients' actual disability level measured by the EDSS [46, 61] score is

of great value to help physicians deliver the right type of treatment given the expanding array of disease-modifying medications. To this end, we design a regression model coupled with a clustering algorithm in which each component represents instances with similar bias characteristics. Because the relation between the features and the disability level is unknown and complex, we resort to a flexible Gaussian process (GP) regression [68] model. We further apply a non-parametric Dirichlet process-based mixture model (DPM) [77] to infer the number of components from the data because we have no a priori knowledge to estimate various types of bias in our data. Thus, our main prediction model is a non-parametric mixture of Gaussian Processes (DPMGP) model [70].

In addition to patient data, we have domain knowledge that the disease characteristics of patients with different initial levels of disability are likely to be different (i.e., patients with higher initial disability level are likely older, have longer disease duration, and have been previously exposed to treatments). Furthermore, the potential treatment options may differ for subjects belonging to different disease progression subgroups. Hence, it is essential that we accurately discover these domain-informed subgroups and apply them as constraints to guide our main prediction model. Because our experts provide us an estimate of forming these subgroups based on the initial disability levels, one solution is to take a two-level hierarchical clustering approach by first partitioning the patients into subgroups using domain knowledge and then modeling each subgroup with a DPMGP model, which takes patient bias into account and provides us with a better prediction model compared to a single GP. A second approach is to learn the two-level hierarchical clusters of disease subgroups and the prediction model simultaneously. The latter approach not only allows us to use the domain knowledge to steer the prediction model, but also the clusters induced automatically from the prediction model can provide valuable feedback to refine our domain knowledge and lead to better discovery of the disease progression subgroups. It is, however, more challenging because these two tasks (discovering the disease subgroups and predicting the disease progression) need to be modeled separately and the two learning algorithms need to communicate to

achieve an optimal solution. To this end, we employ a k-means algorithm [54] on the baseline level of disability to model the disease progression sub-groups. Our new model is a domain induced Dirichlet mixture of Gaussian processes (DI-DPMGP) model that maximizes the consistency between the data and our domain knowledge. The details of our model and its performance are described in Chapter 3.

1.2 Epileptogenic Lesions Detection in Focal Cortical Dysplasia Patients

Epilepsy is a neurological disorder caused by malfunctioning nerves cells in the brain. As a result, patients suffer recurrent, unprovoked seizures. Epilepsy affects 1% of the population, among which 1/3 is resistant to medical treatment [48]. For these patients, surgical removal of the lesional areas in their brains is the last hope of living a seizure-free life. Visual detection of lesional areas on a cortical surface is critical in rendering a successful surgical operation for Treatment Resistant Epilepsy (TRE) patients. The most widely used technology in identifying the epileptic lesions is MRI plus intracranial EEG. Unfortunately, 45% of Focal Cortical Dysplasia (FCD), the most common kind of TRE, patients have no visual abnormalities in their brain's 3D-MRI images [87], which means doctors cannot detect any abnormal areas in their MRI images. For these "MRI-negative" patients, the seizure-free rate after surgery is only 29%, versus 66% for "MRI-positive" patients [80].

We collaborate with doctors from NYU Comprehensive Epilepsy Center and apply machine learning methodologies to identify the resective zones for these "MRI-negative" FCD patients [96]. These patients are further categorized into three subtypes (type I, II and III) [12]. Our model identifies the abnormal areas in a patient's brain which in turn serve as a focus of attention mechanism for the neuroradiologists in placing the iEEG sensors on the patient's cortex. In particular, we apply surface-based morphometry (SBM) [22] to characterize the human brain by explicitly modeling the cortex using a suitable geometric model. SBM has been used successfully for analyzing and detecting neurological abnormalities in various neu-

rological disorders such as Schizophrenia [69], Autism [60], and Epilepsy [79, 39]. Using the SBM methodology, each human brain surface is represented by approximately 0.3 million vertices characterized by features extracted from the MRI image. The machine learning task is to train learning algorithms to distinguish between normal and lesional vertices on the extracted surfaces.

To form our training data, we examine "MRI-negative" patients who are seizure-free after their surgery, for whom the resected tissue has been histopathologically verified to be lesional. Six representative patients were selected from each subtype group resulting in a total of eighteen patients available to our research. Note that the reason for this dearth of patients is that only a few FCD patients proceed to surgery without visible lesion found on their MRI, and of those that do, less than a third experience complete seizure freedom after their surgery [5]. *(Indeed, the six type II patients in our collection represent the entire population of FCD type-II MRI-negative patients treated at NYU in a three year period.)* Vertices within these eighteen patients' resection zones constitute the positive instances. Because the size of the resection region varies from patient to patient, each patient contributes a different number of positive instances to our dataset ranging from 629 to 18,972. Consequently, the total number of positive instances in our dataset is 163,920. Our negative training samples come from fifty healthy individuals ("controls") who underwent the same MRI protocol. For each patient, we extract data from each of the fifty healthy images from the same corresponding location as the patient's resection region. Thus, our total number of negative instances, i.e., 8,196,000, is fifty times the size of the total of positive instances.

One of the major confounding factors inhibiting the development of an effective classifier for detecting FCD lesions is inter-patient variability. The morphology of the human brain such as its thickness and curvature are affected due to age, gender, genes, lifestyle, etc. [72, 73]. Because the data of each patient has its own unique morphological characteristics, treating the data from all the patients in an identical manner will lead to poor classification accuracy. Similarly, the distribution of pathological features that define an FCD lesion differs across FCD subtypes.

For example, in addition to causing other morphological abnormalities, FCD type I lesions appear on MRI as abnormally thin regions of cortex, while FCD type II is characterized by abnormally thick regions. In order to discover subgroups in the data that align with meaningful combinations of both patient and FCD subtype characteristics, we apply a Dirichlet process-based mixture model (DPM) [77] with each subgroup representing a particular bias group in the data.

Our second challenge is a lack of features associated with our task. Indeed, various studies have shown that there are five features (*Cortical thickness*, *Gray/white-matter contrast (GWC)*, *Sulcal depth*, *Curvature*, *Jacobian distortion*) which are effective in detection of FCD lesions [79]. To this end, we resort to restricted Boltzmann machines (RBMs) [76, 94], which have been shown to learn a set of nonlinear features that lead to enhanced performance in a variety of learning tasks [71, 67]. In particular, the RBM would produce an "enhanced" feature space for the data in which we will induce our bias subgroups to discriminate various disease subtypes and account for inter-patient variability. However, because learning a joint distribution for Dirichlet mixture of RBMs is computationally intractable, we propose a multi-stage learning procedure. We first employ a DPM model to partition the data into k clusters, where the clustering is carried out in the feature space pre-learned by the RBM(s). We then produce n ($n \leq k$) classifiers by training a supervised version of modified RBM for each non-empty cluster using the weights obtained from the DPM model. The final classification of a given instance is obtained by an ensemble of these n classifiers.

Our DPM-RBM model achieves up to 58% lesion detection rate on eighteen MRI-negative patients. Although a sample of eighteen may seem small, the results are significant since a board of experienced neuroradiologists failed to locate any lesion for all eighteen patients. We present the details of our model and experimental results in Chapter 4.

1.3 Student Selection in the Graduate Admission Process

Master's education is the fastest growing and largest component of the graduate enterprise in the United States. According to the 2016 joint survey conducted by the CGS (Council of Graduate Schools) and ETS (Educational Testing Service) [15], first-time enrollment in U.S. graduate programs reached a record high total of 506,927 students in Fall 2015. Because of the rise in applicants, the admissions process may become increasingly tedious and challenging. Accurately predicting which students are best suited for graduate programs is beneficial to both students and colleges.

In Chapter 5, we propose a quantitative machine learning approach to predict an applicant's potential performance in the graduate program [99]. Our work is based on a real world dataset consisting of MS in CS students in the College of Computer and Information Science program at Northeastern University. In particular, we aim to predict an applicant's potential success based on the performance of previously admitted students in the graduate program.

A major challenge associated with this task is admission committee's potentially subjective decisions on student applications. Because the membership of the admission committee changes over the years, data collected from students across multiple academic years do not conform to the same underlying distribution. Although the issue of bias is similar to the MS and FCD tasks, here we have the domain knowledge with respect to the exact division of our bias subgroups. To leverage this knowledge, we adopt an approach within the "Learning Using Privileged Information" (LUPI) [83] paradigm. LUPI is a variant of SVM [20] which can establish a unique classification hyperplane for each pre-defined data subgroup via modifying the objective function and constraints of a standard SVM. In our model, each data subgroup consists of students enrolled in the graduate program in each academic year.

Another challenge is the shortage of training data because only a limited

amount of applicants are admitted to the program each year. To this end, we leverage the large amount of applications that are either rejected (i.e., not admitted) or who declined (i.e., admitted but chose not to enroll), which serve as an unlabeled auxiliary dataset.

Our proposed model, illustrated in Chapter 5, is a new variant of SVM which incorporates the domain knowledge of bias subgroups and makes use of both labeled and unlabeled data in its learning process. Our experimental results demonstrate an effective predictive model that could serve as a Focus of Attention (FOA) tool for an admission committee.

1.4 Roadmap

The rest of the thesis is organized as follows. Chapters 2 and 3 present our work in our first application, i.e., predicting the disease course of Multiple Sclerosis (MS) patients [97, 98]. We describe a transfer learning approach and a Bayesian non-parametric mixture model: the "Domain Induced Dirichlet Mixture of Gaussian Processes Model (DI-DPMGM)" respectively. In Chapter 4, we present the work on detecting epileptogenic lesions and describe our approach of employing a non-parametric mixture of restricted Boltzmann machines (RBM) [96]. In Chapter 5, we present the work in our third application, i.e., masters students admission for professional institutions and describe our model which is a new variant of SVM [99]. We conclude and discuss future work in Chapter 6.

Chapter 2

Transfer Learning

2.1 Introduction

Practitioners applying machine learning methods to patient data often find themselves in a common predicament: data that involves physician judgment of clinical tests can be subjective and as a consequence lead to their having different data distributions. If the goal is to learn a classification model from data collected from a set of physicians, the assumption that the training data is drawn from the same distribution is violated. One solution is to learn a separate classifier for each physician, but this requires that each physician sees a sufficient number of patients to form a good generalization. A second naive solution is to normalize each physician's data, but this is only valid under the assumption that all physicians see the same distribution of patients – which is not the case. We call this second complication "patient bias" because patients have biases in their selection of physicians [26].

We propose a solution to physician subjectivity and patient bias via transfer learning [97]. In Section 2.2 we discuss in detail why predicting disease progression in patients with multiple sclerosis exhibits both types of bias and in Section 2.3 we explain why addressing subjectivity and bias is an instance of transfer learning. In Section 2.4, we propose three transfer learning methods designed to solve these problems. In Section 2.5, we prove the convergence of the methods. In Section 2.6, we present an empirical evaluation of our approach on a database of MS patients

	Dr1	Dr2	Dr3	Dr4
Ratio	2.3/1	4.8/1	4.0/1	3.8/1

Table 2.1: Female/Male patients ratio for four MS specialists.

and illustrate that the new methods consistently outperform two baseline methods: using only Doctor Y's data to build a classifier for Doctor Y, and using all of the available data to build a single classifier for all doctors. Our method addresses the additional issue that we have non-uniform misclassification costs: it is more costly to misclassify a patient with a poor prognosis as having a good prognosis than vice versa. A performance comparison of the new methods and two existing transfer learning methods are presented at the end of the section. Finally, we conclude in Section 2.7.

2.2 Challenges in Predicting Disease Outcome

Forming a classifier from patient data collected from multiple providers to predict disease outcome can lead to lower than acceptable accuracy for two reasons. First, each doctor may see a unique distribution of patients; patients often prefer doctors of the same gender and age [26] and more senior specialists may see more severe patients. For example, Table 2.1 reports the ratio of female to male patients in the CLIMB study [30] for four multiple sclerosis (MS) specialists. In the United States, MS is three times more likely in women than men and the disease course differs: on average men progress more rapidly then women [18]. From the table we see that in terms of patients treated by a particular specialist there can be a large difference in the ratio of women to men.

The second difficulty in using data from multiple physicians arises from physician subjectivity. In many medical domains, class labels and values for various diagnostic features are based on physician assessment of functionality rather than on objective medical tests. In such cases, forming a classifier from patient data collected from multiple providers to predict disease outcome can lead to lower than acceptable accuracy. In this section, we describe the source of subjectivity in the

domain of predicting disease severity for patients with MS. In the next section we explain how we can view physician subjectivity as an instance of transfer learning [64].

MS is an autoimmune disease of the central nervous system in which the immune system attacks the myelin sheath (a fatty layer of substance protecting the nerves), resulting in loss/blockage of signals from the brain [18]. Patients suffer from various levels of disability, and the rate of disability accumulation varies across patients. The machine learning goal for this domain is to predict which patients will accumulate disability and which are likely to remain without disability accumulation after five years from study entry. Patients who have a high likelihood of accumulating disability in five years are treated more aggressively and monitored more closely. But, aggressive treatment carries significant potential side effects, thus it is critical to be able to make this prediction accurately. The specific goal is to predict whether the patients' accumulated disability measured by their EDSS scores (explained below) will increase by at least two points at the five year mark using information from the first two years of clinical visits. Earlier research reveals that classifying MS patients based on their data from an initial visit performs only slightly better than random guessing [35] and thus doctors need to monitor patients for a short time period to accumulate sufficient data to make an accurate judgment.

Our data comes from patients enrolled in the CLIMB study [30] from 2008 to 2010 and consists of 257 patients who have been monitored for five years. The dataset contains the clinical visits of patients from nineteen different physicians. Each patient's data includes demographic data such as age and gender. In addition, patients in the study have a clinical visit every six months at which time clinical test results such as neurological function data are collected. The neurological functional data is measured subjectively as is the overall score of their disability level, which is scored using the Expanded Disability Status Scale (EDSS) [46], [61]. The EDSS is calculated by combining information from seven functional system (FS) scales that measure specific aspects of the patient using separate ordinal scales that range from 0-10. To demonstrate the potential subjectivity in scoring of the functional system,

one physician might give a specific patient a score of "1" for his/her joint mobility whereas another physician would give the patient a "2". The threshold given by our domain experts to differentiate a "high" versus "low" disease progression is a $\geq$2 points raise in the EDSS level, which means an increment of the EDSS score by at least 2 from initial visit to five year mark would classify the patients as the high EDSS ("H") class, otherwise, they would be classified as belonging to the low EDSS ("L") class.

A particular challenge of predicting disease outcome in MS is that correctly classifying the "H" class patients is more important than correctly classifying the "L" class patients. This is because an "H" classification engenders closer monitoring; only after close monitoring is a stronger drug administered to the patient. Thus an "L" misclassified as an "H" only costs more money and time in terms of visits, whereas an "H" misclassified as an "L" can mean a patient may be given the wrong medication and thus has a worse outcome.

For MS, it is usually more difficult to correctly classify the "H" cases than the "L" cases. We confirmed this empirically in our dataset of 257 patients. Table 2.2 shows the accuracy of each class for four specialists with most number of patients in our dataset. We applied a linear support vector machine (SVM) classifier [20] with a 10-fold cross validation to each of the four specialists' data separately. As expected the accuracy on the "L" patients is significantly higher. Note that the performance was measured on a dataset after replicating the minority class instances in the training data to eliminate class imbalance. To reflect the uneven misclassification cost and help ameliorate the asymmetric classification difficulty, we can vary the cost ratio of the "H" class to the "L" class. As the ratio increases, the accuracy of the "H" class will improve. Of course, this improvement will be at the cost of some decline in the performance on the "L" class. Depending on how much extra monitoring is considered to be reasonable, a doctor can choose his/her preferred cost ratio accordingly.

	H	L
Dr1	0.57	0.75
Dr2	0.40	0.83
Dr3	0.76	0.90
Dr4	0.27	0.67

Table 2.2: Accuracy of "H" and "L" classes with uniform misclassification costs, as predicted by a linear SVM trained on each specialist's patients.

2.3 Transfer Learning

In the traditional machine learning scenario, algorithms are applied to an isolated concept assuming abundant training data. In reality, researchers are often faced with a situation that the above assumption is invalid. Transfer learning arises naturally as a solution to enhance the learnability in many domains where the training data are limited. In the past fifteen years, transfer learning has been widely studied as a machine learning technique. Pan and Yang [64] gave a detailed survey on transfer learning.

Following the notations in Pan et al. [64], a *domain* $\mathcal{D} = \{\mathcal{X}, P(X)\}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$. A task $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ consists of a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$ where $f(\cdot)$ can be learned from the training data, which consists of observed pairs (x_i, y_i) where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ can be used to predict the corresponding label, $f(x)$, of a new instance x. From a probabilistic viewpoint, $f(x) = P(y|x)$.

Given a primary domain $\mathcal{D}_\mathcal{T} = \{\mathcal{X}_\mathcal{T}, P(X_T)\}$ and a learning task $\mathcal{T}_\mathcal{T} = \{\mathcal{Y}_\mathcal{T}, f_T(\cdot)\}$ together with an auxiliary domain $\mathcal{D}_\mathcal{S} = \{\mathcal{X}_\mathcal{S}, P(X_S)\}$ and a learning task $\mathcal{T}_\mathcal{S} = \{\mathcal{Y}_\mathcal{S}, f_S(\cdot)\}$, transfer learning aims to help improve the learning of the primary predictive function $f_T(\cdot)$ leveraging the knowledge in $\mathcal{D}_\mathcal{S}$ and $\mathcal{T}_\mathcal{S}$, i.e., we want to achieve a better estimate of the marginal distribution $P(Y|X)$ with additional information from the auxiliary data.

Pan et al. [64] further divide the transfer learning settings into two major

categories according to the similarities between the domains ($\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$) and tasks ($\mathcal{T}_\mathcal{S}$ and $\mathcal{T}_\mathcal{T}$). In particular, *Inductive Transfer Learning* refers to a setting when the $\mathcal{T}_\mathcal{T}$ differs from the $\mathcal{T}_\mathcal{S}$ regardless of their domain similarities. A typical example would be trying to classify a multi-class dataset with help from a two-class dataset. *Transfer Learning* refers to a setting when the $\mathcal{D}_\mathcal{T}$ differs from the $\mathcal{D}_\mathcal{S}$, but the primary task is the same as the auxiliary task. This happens when two datasets are closely related but have either a different feature space or different underlying distributions.

Predicting the disease course for an MS patient X who is being seen by physician Y using a classifier formed from data from multiple physicians is an example of *transfer learning*. In this scenario, classifying patients of physician Y is the primary task and, classifying patients of the other physicians is the auxiliary task(s). Transfer learning is applicable when two datasets are closely related but differ in the feature space or distribution. In the previous section, we noted that different physicians may see different distributions of patients and further that different physicians seeing the same patient might provide different values for various clinical tests based on the inherent subjective nature of the tests. Nevertheless, data from different doctors are closely related as they use the same feature space and similar criteria for diagnosis.

Wu et al. [90]'s method addresses the scenario when supplementary information comes in the form of examples from a related domain. They presented a leaf image classification algorithm when the primary training data are limited but low quality auxiliary data are abundant. In particular, the authors explored both k-nearest neighbor (KNN) and SVM as the learner. In the case of KNN, two classifiers were built for the primary and auxiliary data separately. The final decision was made on a weighted vote of the two classifiers. The weight parameter was obtained by cross validation. In the SVM case, different trade-off parameters were installed for the primary and auxiliary data. The parameters were again obtained by cross validation.

Nguyen et al. [59] had a different form of auxiliary information in which they

were given a confidence for each class label. In their framework, the accuracy and reliability of the confidence (presumably provided by the experts) are essential. The authors further developed a ranking approach to improve the noise tolerance related to this human subjectivity. In our particular task, this information is not available. At best we could give the physician's data a higher confidence than the auxiliary physicians', but this simple division ignores that some physician's data will be more similar to the primary physician's than others.

Perhaps the work most closely related to ours is the TrAdaBoost algorithm [21] that combines transfer learning with the AdaBoost algorithm. TrAdaBoost iteratively re-weights the auxiliary dataset and adds it to the primary training set. At each iteration, the weights of both the primary and auxiliary data are adjusted as follows: weights of the incorrectly classified instances are decreased (by $\beta \in (0,1]$) for the auxiliary dataset to reduce their contribution whereas for the primary dataset, they are boosted by $(1/\beta)$ to elevate their importance for the next iteration; weights of all correctly classified instances remain the same for both primary and auxiliary data. At the implementation level, instances are weighted in an SVM using a per instance tradeoff parameter.

One implicit assumption in the existing transfer learning literature is that incorporating auxiliary data in the training process does not introduce class imbalance. However, this is not the case in our task due to the asymmetric property that the "L" class is much easier to classify than the "H" class. In particular, for a method like TrAdaBoost, more "L" instances from the auxiliary dataset will have their weights increased at each iteration because they are correctly classified. As a result, more "L" than "H" instances from the auxiliary dataset will be added to the training data due to their higher weights. This eventually lead to significant class imbalance once the algorithm converges. We discuss this challenge in more detail in Section 2.6.

Figure 2.1: Auxiliary data selection algorithm.

Input: $Train_{D_k}$, $Test_{D_k}$, T_{aux}

1. *initialize* $i = 0$; $T_0 = Train_{D_k}$; $T'_{0,aux} = \emptyset$;
. *best-err = 1.0*
2. *repeat until convergence or* $i > max\_iteration$
3. *build classifier* S_i *over* T_i
4. *if* $err(S_i) <$ *best-err on* $Train_{D_k}$
. $S_{best} = S_i$
. *best-err* $= err(S_i)$
5. $T'_{i+1,aux} = get\text{-}subset(T_{aux}, S_i)$
6. *if* $(T'_{i+1,aux} \neq T'_{i,aux})$
7. $i = i + 1$
8. $T_i = Train_{D_k} \cup T'_{i,aux}$
9. *else: break*
10. *endRepeat*
11. *calculate* acc_H, acc_L, $acc_{overall}$ *on* $Test_{D_k}$

Output: acc_H, acc_L, $acc_{overall}$

2.4 Automatic Auxiliary Data Selection

In this section, we address how to determine which instances from the auxiliary data should be included in the training data. Recall that in transfer learning we have the primary dataset (in this case the particular doctor D_k's patient records T_{D_k}) and the auxiliary dataset T_{aux} (the data from the other doctors). Because our ultimate goal is to create an accurate classifier for each doctor, our performance goal is to find the subset of T_{aux} that improves classification of patients from doctor D_k. Thus we can evaluate a particular candidate subset T'_{aux} by forming a classifier from T'_{aux} $\cup$ Train$_{D_k}$ on Test$_{D_k}$, where T_{D_k} has been split into a training and test set.

We are seeking the *subset* of the auxiliary data that achieves the best performance on the primary set. However it is intractable to evaluate all possible subsets of T_{aux}. Thus each of our heuristic methods is geared at approximating this goal in a computationally tractable way.

The general outline of all three methods is an iterative process that differs

in how T'_{aux} is generated at each iteration. We show the algorithm sketch in Figure 2.1. Let T_i be the training set at iteration i. We begin with just the primary data; i.e., T_0 =Train$_{D_k}$ and S_0 is the classifier trained from T_0. At each iteration, we build a classifier S_i from the current candidate training set T_i and we then get a new candidate training set T_i =Train$_{D_k}$ $\cup$ $T'_{i,aux}$ which is comprised of all of the training data for doctor D_k and a candidate subset of T_{aux}. Line 4 evaluates the current candidate training set T_i on the training data by performing a 10-fold cross-validation (CV) where $T'_{i,aux}$ is added to each Train$_{D_k}$. The 10-fold CV is performed on Train$_{D_k}$ instead of T_i. This is because we are seeking a classifier that has the least training error on the primary dataset rather than the augmented dataset. The role of $T'_{i,aux}$ is limited to helping train the classifier. If the performance of S_i is better than the current S_{best}, it will become the new S_{best}. The algorithm repeats until $T'_{i,aux}$ has converged or we exceed the maximum number of iterations allowed. At this point the algorithm outputs the final classifier S_{best} for doctor D_k. Note that the pseudo code shows our experimental setting where we output the accuracies on the held out test set for doctor D_k, i.e., Test$_{D_k}$.

2.4.1 Method I

Our first approach is to find the maximal subset from the auxiliary data T_{aux} that is "consistent" with the primary dataset. This "consistency" is defined as zero contribution to the training error. Our assumption is that such a subset possesses a similar distribution to the primary dataset and, therefore, may improve the learning performance.

This heuristic algorithm starts with a classifier S_0 built from doctor D_k's data. All instances from T_{aux} that are correctly classified by S_0 constitute the subset $T'_{0,aux}$ which will be added to D_k. This process repeats with next classifier S_i built using combined data $D_k \cup T'_{i-1,aux}$ and next $T'_{i,aux}$ selected by classifier S_i. The process stops when either $i > max\_iteration$ or the subset $T'_{i,aux}$ converges, i.e., when $T'_{i,aux} = T'_{i-1,aux}$.

2.4.2 Method II

Our second approach relaxes the requirement for an instance in T_{aux} to be selected as a candidate. In particular we associate a probability $P_i(x_j)$ of being selected in iteration i to each instance $x_j \in T_{aux}$. To form $T'_{i,aux}$ we sample a subset from T_{aux} according to $P_i(x_j)$. We initialize these probabilities to $\alpha \in (0, 1)$. At each iteration, if the current classifier S_i classifies x_j correctly, we increase $P_i(x_j)$ otherwise we decrease $P_i(x_j)$. Specifically,

if (instance x_j is classified correctly by S_i) then

$$P_i(x_j) = P_i(x_j) + (1 - P_i(x_j)) * \beta$$

else

$$P_i(x_j) = P_i(x_j) * (1 - \beta)$$

where $\beta \in (0, 1)$. In this approach, α controls the initial size of the auxiliary set and β controls how quickly we converge on a final selected subset. In practice they can be set using a grid search wrapped around lines 1-10 of the general algorithm; i.e., before we test the final accuracies on the held out test set Test_{D_k}, the algorithm is executed on grid of different α and β values to select S_{best}. Compared to TrAdaBoost, Method II has similar treatment on the auxiliary data but different handling on the primary data. Specifically, Method II does not associate probabilities to the primary instances as TrAdaBoost does. Another difference is that Method II does not engage boosting as part of its algorithm.

2.4.3 Method III

The rationale for our third approach is based on the conjecture that an individual instance $x_j \in T_{aux}$, although incorrectly classified by a classifier S_i, may still be a member of the best subset. Similar to Method II, we associate a probability $P_i(x_j)$ of being selected in iteration i to each instance $x_j \in T_{aux}$. To form $T'_{i,aux}$ we sample a subset from T_{aux} according to $P_i(x_j)$. We initialize these probabilities to $\alpha \in (0, 1)$. The difference from the previous method is that here at each iteration,

Figure 2.2: Generic On-Line Algorithm

1. *init* $\beta \in [0,1]$
. $\mathbf{W}^1 = w_1^1, \cdots, w_N^1$ *where* $\sum_1^N w_j^1 = 1$

2. *for iteration i = 1, . . . T*

3. *normalize the instance selection probability:*
. $\mathbf{W}^i = \frac{\mathbf{W}^i}{\sum_1^N w_j^i}$

4 . *calculate loss for the* i^{th} *iteration:*
. $L^i = \sum_{j=1}^{N} w_j^i l_j^i$ *(l_j^i is the loss for x_j in iteration i)*

5. *update new weights:*
. $w_j^{i+1} = w_j^i \beta^{l_j^i}$

6. *endFor*

we determine whether S_i, which utilizes the current selected subset $T'_{i,aux}$, leads to better performance than a classifier based only on doctor D_k's data. Specifically, we update the probabilities as follows:

if $(\text{err}(S_i) < \text{err}(S_0)$ on $\text{Train}_{D_k})$ then

$$P_i(x_j) = P_i(x_j) + (1 - P_i(x_j)) * \beta \quad \forall x_j \in T'_{aux}$$

else

$$P_i(x_j) = P_i(x_j) * (1 - \beta) \qquad \forall x_j \in T'_{aux}$$

where $\beta \in (0,1)$. As with Method II, parameters α and β are set via grid search.

2.5 Convergence of the Methods

Method I does not converge without additional constraints. We introduce a modification to guarantee its convergence. Proof of convergence for Method II and III is given by mapping them to a Generic On-Line Algorithm [29].

2.5.1 Method I

Theoretically, it is possible to construct a dataset such that $T'_{i,aux}$ will oscillate among different subsets of the auxiliary data and, hence, will not converge. In practice, Method I converges in all cases for our dataset. Nevertheless, to prevent the unlikely event, the algorithm elects to permanently remove an instance from the auxiliary dataset if it oscillates in and out from $T'_{i,aux}$ more than c times; c can be chosen based on how long one wants to run the algorithm, but $c < 10$ seems like a reasonable choice. This modification guarantees convergence of Method I.

2.5.2 Method II and III

Method II and Method III can both be cast into the Generic On-Line Algorithm (GOLA) outlined in Figure 2.2. The algorithm and detailed proof of its convergence can be found in [29]. In GOLA, for each iteration i, each instance x_j is associated with a weight w_j^i and a loss function l_j^i. The weights are initialized to $\frac{1}{N}$ where N is total number of instances. At each iteration, the loss function l_j^i is re-evaluated for each instance x_j, and a new weight is calculated according to the new loss obtained for each x_j. In particular, instances that have suffered a loss will be reduced by a factor $\beta^{l_j^i}$ while instances that have no loss will retain the same weight. All weights are normalized (line 3) at onset of the next iteration. Note that the convergence of the algorithm is universal for any selection of initial weights and parameter β [29].

The probability $P_i(x_j)$ in Method II and III corresponds to the the weight w_j^i for instance x_j in iteration i. Parameter α corresponds to the initial weight for each instance. The definitions of the generic loss function l_j^i for instance x_j in iteration i are defined as follows:

Method II: $$l_j^i = \begin{cases} 0 & \text{if } x_j \text{ is correctly classified by } S_i \\ 1 & \text{otherwise} \end{cases}$$

$$\text{Method III:} \qquad l_j^i = \begin{cases} 0 & \text{if } x_j \in T'_{aux} \text{ and } D_k \cup T'_{aux} \text{ has} \\ & \text{better accuracy than } D_k \\ 1 & \text{otherwise} \end{cases}$$

Thus, although Method II and III differ in their definition of the loss function, this is irrelevant to the proof of convergence because the algorithm converges with any generic loss function.

Finally, after the normalization process in GOLA, the profitable weights are increased while the non-profitable ones are decreased. Indeed, the change in weights is equivalent to increasing all the profitable instances in the previous iteration by some factor $\beta_1 \in [0, 1]$ and decreasing all the non-profitable ones by some factor $\beta_2 \in [0, 1]$. This is in correspondence to the weight adjustment procedure for Methods II and III. Since the proof holds for all choice of $\beta \in [0, 1]$, this leads to convergence of Method II and III.

2.6 Experimental Results

In this section we first compare the three proposed transfer learning methods to two baseline non-transfer learning methods: learning a classifier for each doctor using only that doctor's data and learning a single classifier for all doctors. We then compare these methods to two transfer learning methods from the literature: KNN [90] and TrAdaBoost [21]. We start with a discussion about our choice of classifier and how we incorporate non-uniform misclassification costs. We next describe our data in more detail and outline the experimental method. We then present the results of our experiments which illustrate that the new transfer learning approaches outperform the two baseline methods (particularly for higher misclassification cost ratios) and the two transfer learning methods from the literature.

2.6.1 Choice of Classifier

As discussed in Section 2.2, it is more important to classify the "H" class correctly than the "L" class. Thus we need a method that allows us incorporate misclassification costs. We choose to use a soft margin linear SVM as our learner in order to introduce different trade-off parameters C_L, C_H for the "L" and "H" classes respectively. The objective function for the SVM becomes:

$$\min \left\{ \tfrac{1}{2}\|w\|^2 + C_L \textstyle\sum_i^n \xi_i^L + C_H \sum_i^m \xi_i^H \right\}$$

subject to: $\forall i$

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i^L, \quad \xi_i^L \geq 0 \quad for \;\; i = 1 \ldots n$$

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i^H, \;\; \xi_i^H \geq 0 \;\; for \;\; i = 1 \ldots m$$

where (x_i, y_i) is a training sample and its label, n and m are the number of instances in classes "L" and "H" respectively, and ξ_i^L and ξ_i^H are the slack variables for instances of classes "L" and "H" respectively.

2.6.2 Experimental Method

Our data consists of 257 patients in the MS dataset from nineteen different physicians. Our method requires that we have sufficient patients to assess performance, thus we restrict that each of our primary datasets come from a doctor with at least five instances in each class. Only four doctors in our database satisfy this criterion. Once a primary dataset is selected, all the patients from the remaining 18 doctors form the auxiliary dataset.

The features of our dataset can be categorized as demographic features, clinical test scores collected for five time periods (i.e., initial visit plus every six months for two years), and five EDSS scores, one for each time period. EDSS is a summary score of how well the patient is functioning. Because we are most interested in the change of disease course, we elect to use sum of changes of each functional score over the initial score for each of the seven functional categories. Similarly, for EDSS, we added a feature for each change of follow-up score over the initial score and for the

sum of changes. This pre-processing step can further achieve a smoothing effect in the feature values to reduce noise which arises for two reasons: 1) values can be elevated because the patient is suffering from an attack at scheduled visit time or 2) values can fluctuate based on physician bias because patients see different doctors at different visits. An empirical evaluation of using the data directly to using the data after this preprocessing step showed considerable difference. Note that the four doctors chosen for our experiments saw each of their patients for three or more of the five visits. Thus there is an underlying assumption in our work that, to build a classifier for a particular doctor, the majority of the visits in the first two years must be with that doctor. We address this assumption further in Chapter 6 when we discuss our future work. Finally, we normalize each feature using z-score normalization where the mean and standard deviation of each feature is computed across the entire dataset.

Another issue we need to address with our dataset is class imbalance. The total number of instances of "H" and "L" classes is 38 and 219 respectively. In order to overcome this imbalance issue, we adopted the SMOTE [17] technique to balance the dataset. This choice was made after experimental comparison with other existing methods such as undersampling and oversampling. In particular, for the two baseline methods (using all data and doctor's data alone), the SMOTE algorithm will compute the imbalance ratio as:

$$r = [\frac{\#of\text{``}L\text{''}instance}{\#of\text{``}H\text{''}instance}]$$

Then for each "H" instance x_j, one synthetic "H" instance is added along the direction of each of its $r-1$ nearest neighbors. The feature values of a synthetic instance are computed as:

$$\mathbf{f}_k = 0.9 * \mathbf{f}_{x_j} + 0.1 * \mathbf{f}_{n_k} \quad \text{for } k = 1, \ldots, r-1$$

where $\mathbf{f}_k$, $\mathbf{f}_{x_j}$ and $\mathbf{f}_{n_k}$ are the features of the kth synthetic "H" instance, x_j, and the kth nearest neighbor respectively.

For the three transfer learning methods, the above procedure is applied to the primary data and auxiliary data separately to achieve balanced datasets at onset

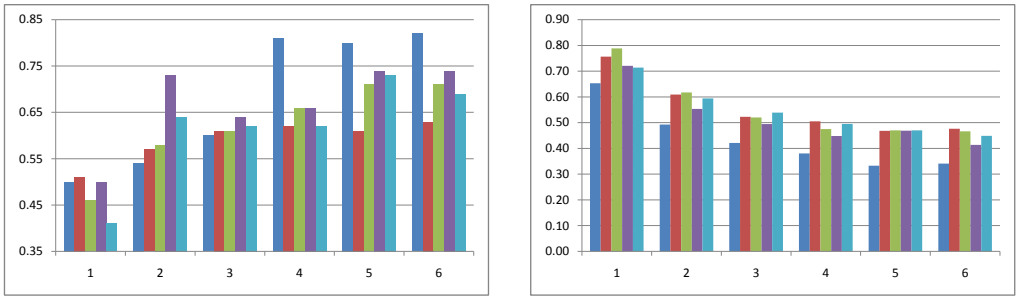
of the program execution. The reason we balance the two datasets separately is that each added data point has to be categorized as either a primary data or auxiliary data. Adding points across the primary and auxiliary datasets would not only have difficulty in categorizing the points but also distort the idiosyncratic distributions of the two datasets.

In our experiments we evaluate each method for a particular doctor by running five 10-fold cross-validations (CV) over that doctor's dataset and averaging across the five runs. Note a CV run here is distinct from the internal CV in our three methods used to select the subset of the auxiliary data. Methods II and III require that we set the parameters α and β. As discussed earlier we ran a grid search in which $\alpha \in \{0.5, 0.20, 0.10, 0.05\}$ and $\beta \in \{0.02, 0.04, 0.10, 0.25\}$. These values were chosen based on their functional roles in the algorithm; α controls the initial size of auxiliary data to be added to the primary set and β controls the speed of how fast we increase the probability of a favored instance in the auxiliary dataset.

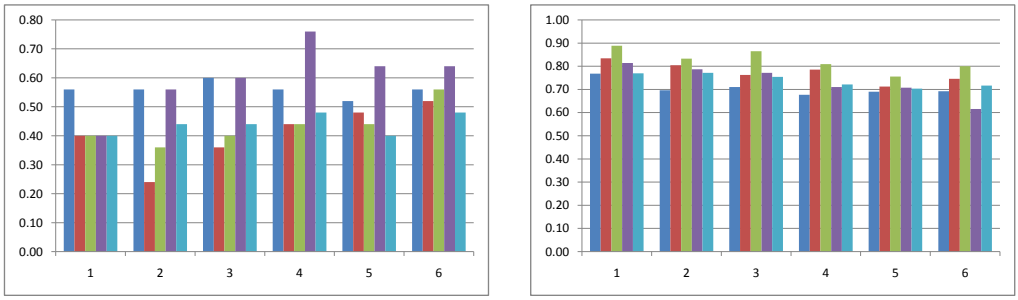
In our experiments we evaluate performance on cost ratios ranging from 1:1 to 6:1; increasing the ratio above 6:1 typically resulted in a degenerative classifier that classified almost all patients as "H". Because each doctor has his/her own patient distribution as well as a unique best auxiliary subset, the optimal cost ratio will vary for each doctor. In practice, one would show the results to the doctors and let them pick what is an acceptable number of false positives from the perspective of their clinical practice.

2.6.3 Primary Data Only versus all Data

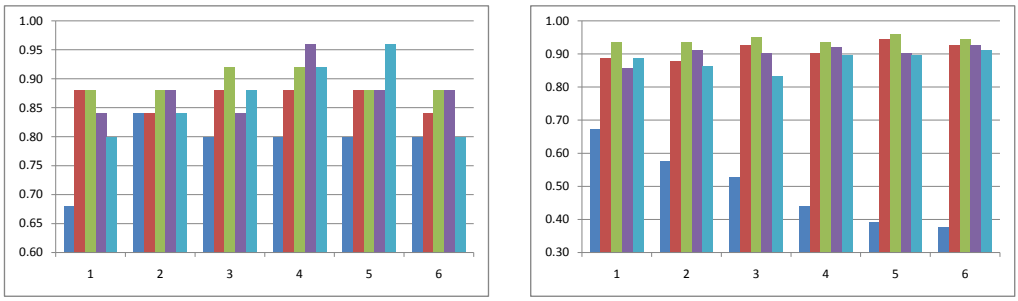
Figure 2.3 shows the results for all methods for all four doctors. For each doctor, the left-hand side shows the accuracy of the patients whose EDSS score has deteriorated by at least two points at five year mark and the right-hand side shows the accuracy for patients whose change in EDSS score is less than two points. We first focus on comparing our two baseline methods: using all of the data and using only the primary dataset for each doctor (the two left most bars for each cost ratio in the plots, respectively). These two baseline methods allow us to evaluate the impact of



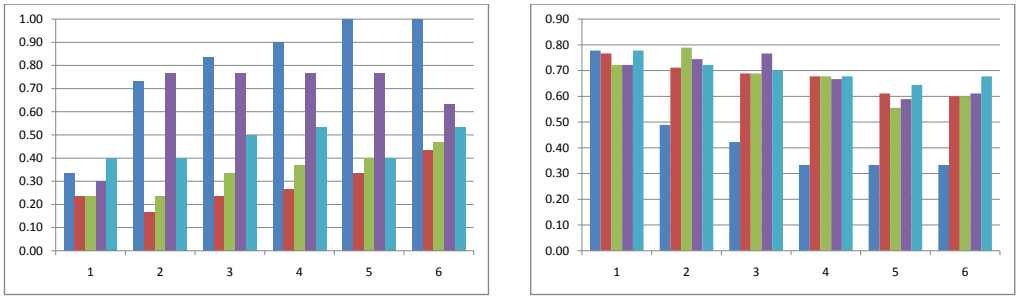
Doctor 1



Doctor 2



Doctor 3



Doctor 4

Figure 2.3: Methods from left to right are Using all Data, Using Primary Data, Method I, II and III respectively. For each doctor, the left graph is performance for the "H" class and the right graph is for the "L" class. The x-axis shows the relative misclassification cost of "H" to "L" and the y-axis shows the accuracy.

cost sensitive learning without transfer learning. In next section, we will analyze the additional exclusive contribution from transfer learning technique.

For the "H" class, using all the data leads to a better performance in general than doctor alone except for Doctor 3. This exception can be explained by a closer examination of the data which reveals that Doctor 3 has the greatest separation among all doctors in the average feature values of "L" versus "H" patients. This separation leads to a good performance with the doctor's data alone. The difference in separation of feature values can be seen in Table 2.3 which shows for each doctor the difference in EDSS values for "L" and "H" patients for each time period. Note that the table shows the unnormalized data and that EDSS values in our data range from zero to six. EDSS0 is the initial visit score and the rest are the corresponding every six month follow-up scores.

Although using all the data outperforms using the doctor's data alone for the "H" class in most of the cases, this high accuracy is gained at a cost of significant decrease on the performance of the "L" class. For example, for Doctor 4, using all the data gives 100% accuracy on the "H" class when the cost ratio is raised to five or six, but the corresponding "L" class accuracy is only 33%. Similarly, for Doctor 1, the "L" class accuracy drops to less than 40% when the "H" class accuracy reaches its maximum.

2.6.4 Performance of the Transfer Learning Methods

In most of the cases, all three transfer learning methods outperform using the doctor's data alone for the "H" class at negligible cost to the "L" class. For some doctors (Doctors 1 and 4), using all of the data outperforms the transfer learning methods on class "H", but at a significant drop on the accuracy for the "L" class. For Doctors 2 and 3 and a cost ratio of 4:1, one or all of the transfer learning methods achieve accuracies for the "H" class that are unattainable by either base-line methods. Overall, we can conclude that transfer learning improves the performance over the two baseline methods particularly at higher costs.

Another observation is that for Methods I and II, the improvements are most

	EDSS0	EDSS6	EDSS12	EDSS18	EDSS24
Dr1	0.66	0.06	0.10	0.26	0.05
Dr2	0.08	1.15	1.19	1.61	2.11
Dr3	**1.28**	**2.18**	**2.14**	**2.62**	**3.56**
Dr4	0.06	0.36	0.64	0.81	1.00

Table 2.3: Difference of the mean EDSS score for each of the time periods for "H" and "L" patients.

significant for the higher cost ratios. Recall that for Method I, the auxiliary subset T'_{aux} in each iteration is comprised of instances correctly labeled by the new SVM S_i. We observed in our experiments that for a cost ratio of 1:1, T'_{aux} contained far more "L" than "H" instances, which in turn leads to increased performance for class "L" as it introduces class imbalance. As the cost ratio increases, the selected subset T'_{aux} shifts towards having an increased percentage of "H" instances. This increase results in an increase in accuracy for the "H" class. It is worth noting that we do not balance the datasets again after each selection of T'_{aux}. This is because the transfer learning principle is based on selection of the "consistent" data (i.e., correctly labeled by S_i) from the auxiliary dataset. Balancing the auxiliary dataset after each selection of T'_{aux} (using the SMOTE algorithm) would result in adding incorrectly labeled data to the training set and, therefore, counteract the principle of transfer learning. For Method III, because the selection for T'_{aux} was not based on the correct label of the auxiliary instances, we observe a more balanced T'_{aux} even when the cost ratio is high.

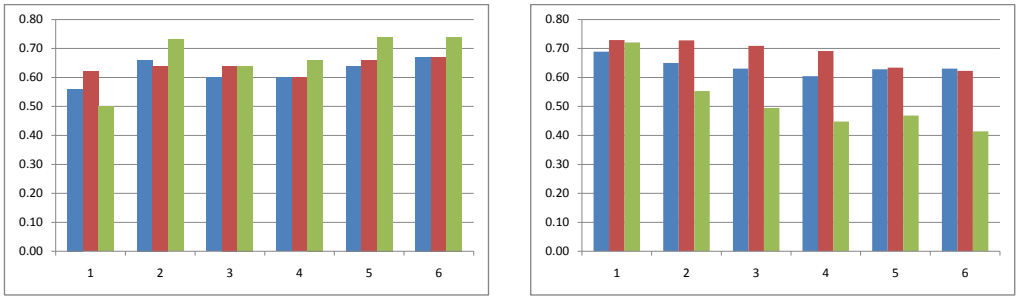
Comparing Methods I and II, we note that Method II eventually converges to a subset similar to Method I because the probabilities for the consistently correctly labeled instances in T_{aux} will asymptotically approach 1 while the others asymptotically approach 0. But the stochastic nature of Method II does perturb the subset selection process, resulting in exploring a larger space of subsets than Method I, and leads to better performance on the "H" class in most cases; i.e., in Figure 2.3, the 4th bar is higher than the 3rd bar for most of the cost ratios in the "H" class plots (left plot for each doctor).

Method III was designed to explore a different space of possible subsets and also to remove the requirement that each instance in the subset be classified correctly. Although for some doctors and cost ratios (e.g. Doctor 1, cost = 5; Doctor 3, cost = 5), Method III can achieve similar or better performance than the other two methods, it has worse performance in the majority of the cases. This is because the hypothesis space for Method III is substantially larger than the other two methods and the algorithm cannot efficiently explore the entire space.

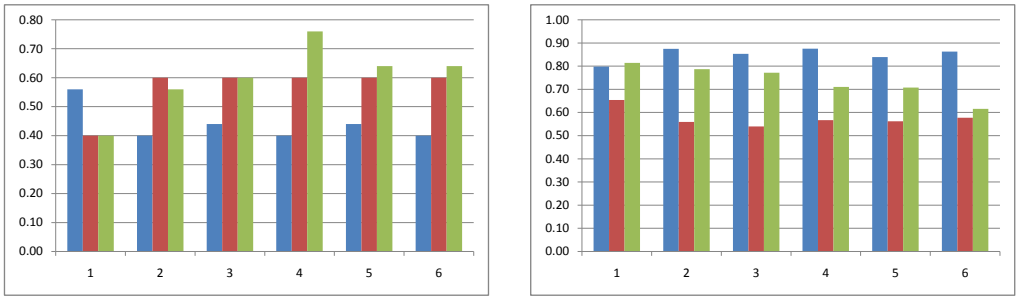
2.6.5 Comparison to other Transfer Learning Methods

We compare our approach to both the KNN [90] and TrAdaBoost [21] algorithms for transfer learning. In our experiments we examine performance when the misclassification cost ratio varies from 1:1 to 1:6 for the "L" and "H" classes respectively. Thus we modify both KNN and TrAdaBoost to accommodate non-uniform costs as follows: for KNN we multiply the "H" class vote by its cost and for TrAdaBoost we multiple the the "H" class weights by its cost. We report our results in Figure 2.4. Note that the figure shows only Method II, which performed the best of our three proposed methods, but that the relative performance of Methods I and III can be seen in Figure 2.3. In the results for each doctor in Figure 2.4 the bars from left to right show the performance of TrAdaBoost, KNN, and Method II respectively.

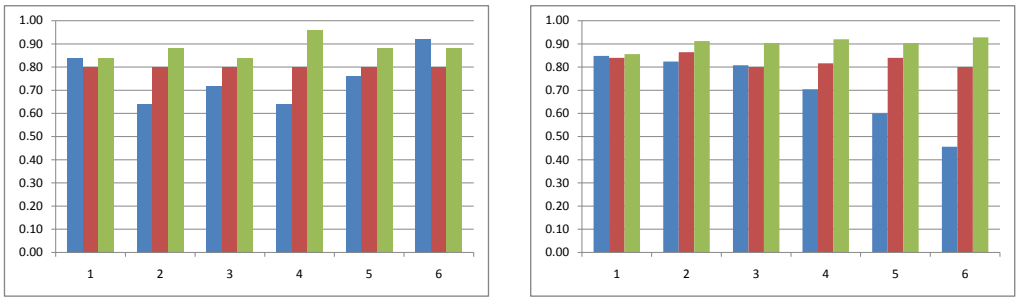
At a cost of 1:1, both KNN and TrAdaBoost perform comparably or slightly better than Method II. This suggests that when we have uniform misclassification costs TrAdaBoost might be preferable. Note that for three of the doctors using *all* of the data is perhaps the best choice at a cost ratio of 1:1 and thus transfer learning is not particularly helpful for this dataset when we have uniform costs. However, our method was specifically designed for the case of uneven misclassification costs. At higher cost rations, Method II performs the best overall for Doctors 2, 3 and 4; the highest accuracy (typically at a cost ratio of 1:4) for the "H" class (left plot for each doctor) that Method II achieves on these doctors is higher than the other two methods with almost no decrease in performance on the "L" class (as measured by its performance at cost ratio 1:1). For Doctor 1, Method II has lower performance



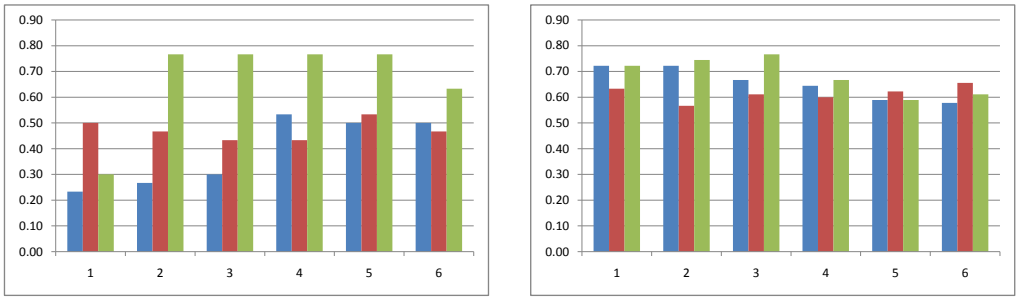
Doctor 1



Doctor 2



Doctor 3



Doctor 4

Figure 2.4: Comparison to other transfer learning methods. Methods from left to right are TrAdaBoost, KNN, Method II respectively. For each doctor, the left graph is performance for the "H" class and the right graph is for the "L" class. The x-axis shows the relative misclassification cost of "H" to "L" and the y-axis shows the accuracy.

on the "L" class when the cost ratio exceeds 1:1. KNN's decrease in performance at higher cost ratios is because instances are not explicitly selected based on their improvement to performance at higher cost ratios. TrAdaBoost's relative decrease in performance for higher cost ratios is likely attributable to the fact that our modification for incorporating misclassification costs requires that the instance weights serve two purposes: instance re-weighting for boosting and adjusting for non-uniform misclassification costs. Because these two purposes can be either positively or negatively correlated for a particular instance, they could provide confounding signals to the SVM learner and thus lower performance. Thus we conclude that for predicting MS progression when there are non-uniform misclassification costs, one should use Method II.

2.7 Conclusion

We proposed transfer learning as a solution for utilizing the data from multiple physicians when there is both physician subjectivity and patient bias. In our framework, the entire dataset is partitioned by physician. Each individual physician's patients are modeled as the primary data and the remainder are modeled as the auxiliary data. We introduced three new heuristic transfer learning algorithms that explicitly incorporate non-uniform misclassification costs and applied them to the problem of predicting disease progression in MS patients. We compared our results to two baseline non-transfer learning methods (using the entire dataset and using the doctor's data only) and to two methods in the literature. We concluded that one of the three proposed transfer learning methods improves performance for non-uniform misclassification costs.

A limitation of transfer learning is that it is only applicable to tasks in which there is a sufficient number of samples in the primary dataset. In the CLIMB dataset (see Section 2.6.2 for details), there are fourteen physicians with fewer than twenty patients, which means we could not build an effective model for those physicians and thus could not predict disease progression for patients seen by those physicians.

Another disadvantage of transfer learning is the exclusion of some available training samples because the model is built using the primary dataset plus a carefully selected subset of samples from the auxiliary dataset. In next chapter, we adopt a Bayesian non-parametric clustering framework to model the homogenous subgroups in the data. Specifically, the algorithm exploits the entire dataset by partitioning it into clusters and each cluster consists of instances of similar bias characteristics. Furthermore, because we do not have prior knowledge regarding the various types of bias in the data, the number of clusters are inferred from data directly. We apply this new approach to our regression task of forecasting the actual disability level measured by the EDSS [61] scores for MS patients.

Chapter 3

Domain Induced Dirichlet Mixture of Gaussian Processes

3.1 Introduction

In Chapter 2, we presented a transfer learning algorithm to predict MS patient disease course at the five year mark using the first two years' longitudinal clinical observations. In addition to classifying a patient's prospective disability status to be 'high' versus 'low', forecasting the actual disability level measured by the EDSS [61] score is of great interest to the physicians. This is because a critical component in the management of patients with MS is correctly predicting which patients will experience worsening disease over the short term. Prognostic information is very valuable in identifying the patients who may benefit from more potent or aggressive treatment or simple closer monitoring.

Our new goal is to predict the actual disability level of MS patients at the five year mark using their first two years' longitudinal data. Because the relationship between the features and progression is unknown, we model disease progression with Gaussian process (GP) regression to provide flexibility as it does not assume any particular functional form between the features and the target values [68]. Most importantly, similar to our last task, our choice of model needs to address the

complications of patient bias and physician subjectivity. Thus, a single regression model is insufficient and we utilize a mixture of GP in which each cluster component represents instances with similar bias characteristics. Because we have no a priori knowledge to estimate various types of bias in our data, we apply Dirichlet process (DP)-based clustering [77] to infer the number of mixing clusters from the data. The conjecture is that different clusters will contain patients with the same type of patient/physician bias. Thus, our main prediction model is a non-parametric mixture of Gaussian Processes (DPMGP) model [70].

In addition to patient data, physicians often have some domain knowledge about how to group patients into disease progression subgroups (e.g., for MS the initial level of disability is thought to be a useful grouping). Finding these subgroups is critical for two reasons: 1) which features are predictive of progression will vary based on subgroup; and 2) treatment options differ based on the likelihood of progression [35]. We utilize the physician-proposed subgroupings as hierarchical constraints for the DPMGP model. Specifically, we evaluate two approaches. The first partitions the patients into the K subgroups defined by the physician's domain knowledge and then learns a DPMGP model for each subgroup separately. This first approach works well if we assume our physicians' knowledge is perfect. Our second approach relaxes this assumption by using the physician-proposed subgroups as constraints; we learn both the subgroups and the prediction models simultaneously. In the second approach, not only can we use the domain knowledge to steer the prediction model, but also the clusters induced automatically from the prediction model can provide valuable feedback to refine our domain knowledge and lead to better discovery of the disease progression subgroups. We call this second approach Domain Induced DPMGP (DI-DPMGP) [98].

DI-DPMGP accommodates both clustering signals from the data and from experts' domain knowledge. We define the "level-1" clusters to be the domain knowledge subgroups and the "level-2" clusters to be induced by the data using the DPMGP model. In our first approach the level-1 clusters are defined by domain knowledge and then within each partition we apply a level-2 clustering via the

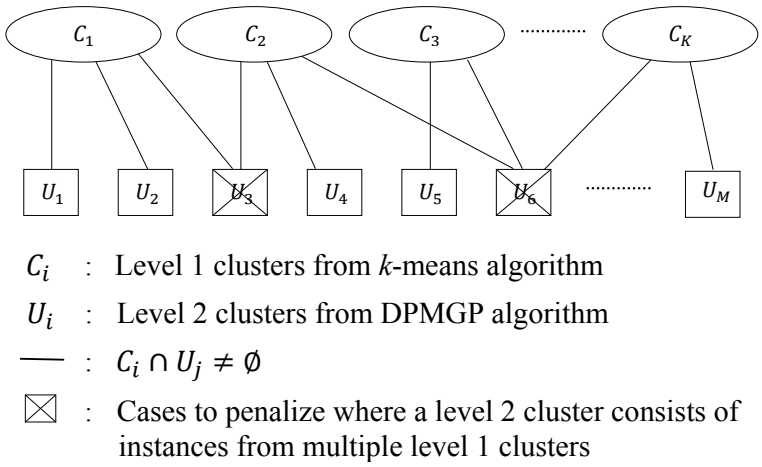


Figure 3.1: Relationship between the hierarchical clusters from the k-means and DPMGP processes.

DPMGP model applied to the partition. In our second approach (shown graphically in Figure 3.1), we use domain knowledge to define the features that should be used to find a level-1 partition and then we apply k-means to learn the level-1 clusters, $C_1 \cup C_2 \cup \cdots \cup C_K$, where K is specified via domain knowledge. The DPMGP algorithm produces another partition $U_1 \cup U_2 \cup \cdots \cup U_M$ of the same data and infers the number of clusters, M, automatically. In our second approach, the level-1 and level-2 clusterings are not independent. In particular, we show an edge from a cluster C_i to a cluster U_j if they contain some of the same patients. Ideally, the level-1 and level-2 clusters agree completely such that we have a perfect hierarchical decomposition (e.g., in the figure U_3 does not have edges to both C_1 and C_2). Thus in our algorithm we penalize such cases. As a result, the C_i's serve as constraints in producing the sub-learning tasks in the DPMGP algorithm and U_i's provide feedback to the k-means algorithm to refine the discovery of disease subgroups. We describe this process in detail in Section 3.4.

3.1.1 Related Work

Integrating domain knowledge into generative models is an active area of research. Chatzis and Demiris explored the Pitman-Yor process (a variation of a DP process) to model the heavy tail behavior of the dataset [16]. Another widely used technique

is constraint-based clustering where do-not-link (DL) and must-link (ML) constraint are specified between certain pairs of instances by domain experts. Ross and Dy employed Markov Random Field (MRF) to integrate DLs and MLs into the DP-MGP model [70]. As stated in their paper, the drawback of their model is that the constraints must be sparse for the algorithm to be computationally tractable. Unfortunately this approach is not feasible in promoting a partition of a dataset because the task would require specifying a constraint between every pair of instances.

Although our approach bears some superficial resemblance to a multi-view approach [92], they are fundamentally different. In multi-view learning, the goal is to maximize the mutual agreement among different sources/views of the data. In our case, the k-means clustering represents experts' opinion which is a high-level grouping of the data and the DPMGP clustering is a refined partition within the groups. There is a hierarchical implication between the two processes. If we indeed consider the two clustering processes as two views on the data, then they are not independent such that the DPMGP view is inside the k-means view. Technically, "inconsistency" in a multi-view approach is defined as two instances belonging to the same group in one view but different groups in the other view. In our case, however, "inconsistency" is defined only in one direction (not the other way around): two patients belonging to different k-means clusters but who are in a single DPMGP cluster.

3.1.2 Contributions

We propose a new approach to incorporate domain knowledge into a non-parametric mixture model. Specifically, our model offers a new method to introduce hierarchical constraints to a non-parametric model in the form of data subgroups. These types of constraints are common in real-world applications where the prediction tasks of a mixture model may be further characterized in different subsets of data due to geographic location, life style, and/or other biases; and that, domain experts may have limited knowledge about the subgroups based on some features.

Another contribution of this work is a solution to model clinical datasets

suffering from physician subjectivity and patient bias. The challenges associated with physician subjectivity are rarely discussed when building predictive models. Although we ground our research in the context of a particular neurological disorder, our model is applicable to any dataset compiled from multiple sources in which some part of the data collected involves human judgment.

3.1.3 Organization of this Chapter

Before presenting our approach, we first review Gaussian and Dirichlet processes in Sections 3.2 and 3.3. We introduce our new domain induced Dirichlet mixture of Gaussian processes (DI-DPMGP) model in Section 3.4. In Section 3.5, we present an empirical evaluation of our model and illustrate its efficacy on two real world clinical (MS and Parkinson's disease) datasets. Finally, we conclude in Section 3.6.

3.2 Gaussian Processes

Given n observations $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$, a regression problem tries to uncover the function $y = f(x)$ such that for a new input value x_*, we can accurately predict the corresponding value y_*. Often, a particular parametric form (such as linear) of f is stipulated and the parameters are inferred from the data (e.g., via the linear regression). In many practical situations, however, there is no natural way of identifying the form of the function f. A Gaussian process (GP) [68] avoids this difficulty by modeling the y values directly.

Formally, a Gaussian process is a distribution over a set of functions $f : \mathbf{x} \rightarrow R$, with the property that when those functions are sampled and then evaluated on a finite set of inputs $\{x_1, x_2, \ldots, x_n\} \in \mathbf{x}$, the obtained values $\{y_1, y_2, \ldots, y_n\} \in R^n$ are normally distributed. Alternatively, one can consider GP as a collection of random variables indexed by the input set $\mathbf{x}$, any finite subset of which has a joint multivariate Gaussian distribution. When there is no information suggesting otherwise, the mean function of GP is often assumed to be 0. Under this assumption, a GP is completely determined by its covariance matrix K (often referred to as GP's

"kernel"). We denote:

$$f(x) \sim GP(0, K)$$

Proper choice of the covariance function K determines a GP's flexibility and applicability to particular situations. One popular choice is the Radial Basis Function (RBF) kernel defined as:

$$k(x, x') = \sigma_f^2 exp[\frac{-(x-x')^2}{2l^2}]$$

where σ_f and l control the correlation strength and its rate of decay between two variables respectively.

Because observations are often noisy, it is common to consider Gaussian processes with added independent noise:

$$y \ \sim GP(0, K + \sigma^2 I)$$

The joint normality of any finite set of variables in GP allows derivation of the following formula for the distribution of a sample $\mathbf{y} = \{y_1, ..., y_n\}$ augmented by a new value $y* \in R^D$:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K + \sigma^2 I & K_*^T \\ K_* & K_{**} + \sigma^2 \end{bmatrix}\right)$$

where

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

$$K_* = [k(x_*, x_1), k(x_*, x_2), \dots, k(x_*, x_n)]^T$$

$$K_{**} = k(x_*, x_*)$$

Standard properties of the multivariate normal distribution give the formula for the conditional expectation of y_* given the sample $\mathbf{y} = \{y_1, ..., y_n\}$:

$$y_*|\mathbf{y} \sim N(\mu_*, \sigma_*^2)$$

where
$$\mu_* = K_*(K + \sigma^2 I)^{-1}\mathbf{y}$$
$$\sigma_*^2 = \sigma^2 + K_{**} - K_*(K + \sigma^2 I)^{-1}K_*^T$$

Hence the best estimate of y_* and the uncertainty around it are captured by μ_* and σ_*^2. The hyper-parameters of the covariance function ($\{\sigma_f, l\}$ in the RBF case), and the noise variance σ can be estimated by maximizing the following likelihood function:

$$\log p(\mathbf{y}|\mathbf{x}, \theta, \sigma^2) = -\tfrac{N}{2}\log 2\pi - \tfrac{1}{2}\log|K + \sigma^2 I| - \tfrac{1}{2}\mathbf{y}^T(K + \sigma^2 I)^{-1}\mathbf{y}.$$

3.3 Dirichlet Process Mixture Model

Dirichlet process (DP) is a family of Bayesian nonparametric models in which the model representations grow as more data are observed [93, 91, 63]. In particular, DP used as a prior in a generative mixture model allows the number of mixing components to adapt to the individual dataset automatically. DP can be interpreted as an extension to the traditional generative model with an arbitrary (infinite) number of mixing components.

Formally, a Dirichlet process is an infinite dimensional discrete distribution with two parameters α and H denoted as:

$$G \sim DP(\alpha, H)$$

where H is the base distribution and scalar α is the strength parameter. H serves as the mean of G and α controls the convergence of G towards H. A Dirichlet process can be constructed using the stick-breaking process [74] as follows:

$$\theta_k^* \sim H \qquad v_k \sim Beta(1, \alpha)$$

$$\pi_k^* = v_k \prod_{j=1}^{k-1}(1 - v_j) \qquad G = \sum_{k=1}^{\infty} \pi_k^* \delta(\theta_k^*) \tag{3.1}$$

where $k = 1, 2, \ldots.$ and δ is the Dirac delta function.

A DP mixture model uses $G(\alpha, H)$ as the prior under the Bayesian framework. The entire dataset is modeled as a mixture of components and each component is parameterized by a random draw (θ) from G. Each data observation belongs to one of the components and is modeled as a function of the parameter of its component, i.e., $f_i(\theta_i)$. Specifically,

$$G|\alpha, H \sim DP(\alpha, H) \qquad \theta_i|G \sim G$$
$$x_i|\theta_i \sim f_i(\theta_i)$$

Consider drawing N samples of θ_i ($i = 1, 2, \ldots, N$) from G. Because G is a discrete distribution, the probability at any given point in the probability space can be non-zero. This implies that the values of the θ_i's will repeat with a positive probability. Hence, these θ_i's exhibit clustering behavior (Polya Urn Scheme). Given the first N samples of θ_i from G, we assume they have produced a set of k distinct values:

$$\Theta^* = \{\theta_1^*, \theta_2^*, \ldots, \theta_k^*\} \text{ where } k < N.$$

It can be shown that the next new sample θ_{N+1} can be either a new value drawn from base distribution H with probability $\propto \alpha$ or can be taken from one of the existing members from Θ^* with probability $\propto c_i$, where c_i is the number of times θ_i^* has been repeated. Specifically,

$$\theta_{N+1}|\theta_1, \theta_2, \ldots, \theta_N \sim \frac{\alpha}{\alpha + N} H + \sum_{i=1}^{n} \frac{c_i}{\alpha + N} \delta_{\theta_i} \tag{3.2}$$

where δ_{θ_i} denotes the distribution concentrated at a single point θ_i.

Equation (3.2) illustrates two important properties of Dirichlet process. First, the concentration parameter α controls the number of distinct values of θ_i's, i.e., the number of mixing components. Second, DP exhibits a "rich get richer" property: the more frequently a θ_i^* has been adopted (i.e., the larger the c_i), the more likely it will be chosen again as the next θ_i value.

There are various techniques such as MCMC sampling[58] and variational inference[11] to conduct inference for a non-parametric model under the Bayesian framework. We adopt the variational approach and in Section 3.4, we describe how to incorporate domain knowledge to our formulation.

3.4 Domain Induced Clustering Dirichlet Mixture of Gaussian Processes Model

In this section, we present our model in detail. We first rationalize our choice of models for our main prediction task (DPMGP) and domain knowledge (k-means). We then describe how to communicate constraints between the two algorithms via a penalty term. Next we present the modified DPMGP model, with equations for posterior inference based on the variational Bayesian framework, and the modified k-means algorithm.

3.4.1 Choice of Models

As discussed in Section 3.1, our main task is to predict the disability level of Multiple Sclerosis (MS) patients at the five year mark using their first two year's longitudinal data. Because the relation (f) between the entire feature set ($\mathbf{x}$) and the disability level ($\mathbf{y}$) is unknown and complex, we resort to a flexible non-parametric Gaussian process (GP) regression [68] model. In our data, physician subjectivity is reflected in the clinical features, whereas patient bias appears in the demographic features of the patients. These two types of bias may not yield to the same form of parametric model. We further model our data as a mixture of GPs due to the existence of multiple types of bias characteristics from different health professionals and patient demographics. Because we have no a priori knowledge to estimate the various types of bias in our data, we impose a Dirichlet process (DP) [77] as the prior in our mixture model. As a result, our main prediction model is a Dirichlet Mixture of Gaussian Processes (DPMGP) model [86], [70].

Let $\mathbf{x} \in R^{N \times D}$ be the observed inputs and $\mathbf{y} \in R^N$ the corresponding regres-

sion values, where N is the number of instances, and D the number of features. We model our data as generated by a mixture of components associated with an infinite set of regression functions $\{f_j\}_{j=1}^{\infty}$. It is convenient to introduce the latent indicator matrix $\mathbf{z} \in R^{N \times \infty}$ which ties the data pairs (x_i, y_i) with the regression functions f_j via the equation $y_i = f_j(x_i)$ (i.e., $\mathbf{z}(i,j) = 1$ and each row contains a single non-zero entry). Each of the regression functions is drawn from a Gaussian process:

$$f_i \sim GP(0, K_i)$$

The model choice for our domain knowledge is based on the fact that our experts provided a subset of features that characterizes the disease subgroups and we know a priori the total number of clusters. We employ a k-means clustering algorithm on these features to obtain the subgroups.

3.4.2 Interaction between the two Clustering Processes

During training, we train the DPMGP and k-means models concurrently. At each iteration, each instance x_i has two membership indicators: one from the k-means algorithm (we denote as c_i) and another one from the DPMGP algorithm (we denote it z_i). c_i and z_i are indicator vectors with only one element with a value of one indicating membership to that cluster and zero for the other elements.

For notational convenience, let us define a function $C(i,j)$ on any pair of k-means membership association vectors:

$$C(i,j) = \begin{cases} 1 & \text{if } \langle c_i, c_j \rangle = 1 \\ 0 & \text{otherwise} \end{cases}$$

where $C(i,j) = 1$ implies that x_i, x_j belong to the same k-means cluster. Similarly, we define $U(i,j)$ on the DPMGP membership association vectors:

$$U(i,j) = \begin{cases} 1 & \text{if } \langle z_i, z_j \rangle = 1 \\ 0 & \text{otherwise} \end{cases}$$

where $U(i,j) = 1$ implies that x_i, x_j belong to the same DPMGP cluster.

In the training process, we will penalize the violations when x_i, x_j belong to different k-mean clusters but are in the same DPMGP cluster. The total number of violations in the entire dataset can be calculated as follows:

$$\begin{aligned} S &= \sum_{1 \leq i \leq j \leq N} (1 - C(i,j)) * U(i,j) \\ &= \sum_{1 \leq i \leq j \leq N} \widetilde{C}_{i,j} * U(i,j) \end{aligned} \tag{3.3}$$

where $\widetilde{C}_{i,j} = 1 - C(i,j)$.

Note that calculation of the penalty term in the summation in (3.3) is one directional, i.e., it only penalizes the scenario that two instances are not in the same k-means cluster ($C(i,j) = 0$), but are put together in the same DPMGP cluster ($U(i,j) = 1$). This is because the k-means clusters are level-1 clusters derived from our domain knowledge, and we would like our modified DPMGP clustering process to respect the experts' opinion reflected in the level-1 clusters (i.e., instances belonging to different level-1 clusters should not belong to the same level-2 cluster). Similarly, as we present in Section 3.4.4, the k-means clustering algorithm is also modified to adjust its decisions using the same penalty term. The uni-directional hierarchically structured penalty requires us to maintain two sets of memberships (c_i's and z_i's). Consequently, we need to modify the objective functions of the DPMGP and the k-means algorithms separately to penalize the overall violations (S) and, hence, maximize the hierarchical consistency (illustrated in Figure 3.1) among the two levels of clusters. We accomplish our goal by modifying the inference of the DPMGP model and the centroid re-assignment step in the k-means algorithm, which we describe next in turn.

3.4.3 Modified DPMGP Algorithm

As discussed in Subsection 3.4.1, the model for our main prediction task is based on a Dirichlet mixture of Gaussian Processes (DPMGP) model. Since a GP can be viewed as a distribution over functions, we let the base distribution H in our DP

to be a zero mean Gaussian process [86]. Our DPMGP model can be specified as follows:

$$G \sim DP(\alpha, H) \qquad\qquad f_i|K \sim G$$
$$y_i|f_i, x_i, \sigma \sim N(f_i(x_i), \sigma^2)$$

Data generated by the above model are naturally partitioned (or clustered) by the distinct functions f drawn from H. Dirichlet process properties imply that the number of clusters (our mixing components) grows as new data are observed [77].

Inference for a DP mixture model is typically conducted using MCMC [58] or variational approximations [11] because in most cases the desired posterior distributions are analytically intractable. The variational approach can be more advantageous due to its scalability and guaranteed convergence. Here we use the mean-field variational inference outlined in [11]. We follow similar notation used in [70] and [16] but with an additional penalty term in the objective function in the optimization formulation which leads to different update rules.

Our model employs the latent variables $\mathbf{z} = \{z_1, z_2, \dots\}$ (z_i's are the rows of the indicator matrix $\mathbf{z}$ mentioned previously), $\mathbf{v} = \{v_1, v_2, \dots,\}$ (v_i's are the stick-breaking proportions [74] in a DP) and $\mathbf{f} = \{f_1, f_2, \dots\}$. Hyper-parameters are σ (the independent Gaussian noise) and $\theta_0 = \{\sigma_0^1, \sigma_0^2, l_0^1, l_0^2\}$ (the kernel parameters of H). The inference algorithm iteratively computes the values of latent variables until convergence.

The joint distribution $p(\mathbf{y}, \mathbf{f}, \mathbf{z}, \mathbf{v})$ can be factored as:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{z}, \mathbf{v}) = p(\mathbf{y}|\mathbf{f}, \mathbf{z})p(\mathbf{f})p(\mathbf{z}|\mathbf{v})p(\mathbf{v}|\alpha)$$

where

$$p(\mathbf{y}|\mathbf{f}, \mathbf{z}) = \prod_{n=1}^{N} \prod_{k=1}^{\infty} N(y_n|f_k^n, \sigma^2)^{z_{n,k}}$$

$$p(\mathbf{f}) = \prod_{k=1}^{\infty} N(f_k|0, K)$$

$$p(\mathbf{z}|\mathbf{v}) = \prod_{n=1}^{N} \prod_{k=1}^{\infty} \left(v_k \prod_{j=1}^{k-1} (1 - v_j) \right)^{z_{n,k}}$$

$$p(\mathbf{v}|\alpha) = \prod_{k=1}^{\infty} \text{Beta}(v_k|1, \alpha)$$

In variational inference, the posterior distribution $p(\mathbf{f}, \mathbf{z}, \mathbf{v}|\mathbf{y})$ is approximated by a computationally convenient distribution $q(\mathbf{f}, \mathbf{z}, \mathbf{v})$. The KL divergence $D_{KL}(q||p)$ can be computed as follows:

$$\begin{aligned} D_{KL}(q||p) &= -\int_{\Psi} q(\Psi) \log \frac{p(\Psi|\mathbf{y})}{q(\Psi)} \\ &= -\int_{\Psi} q(\Psi) \log \frac{p(\Psi, \mathbf{y})}{q(\Psi)} + \log p(\mathbf{y}) \\ &= -\mathcal{L}(q) + \log p(\mathbf{y}) \end{aligned}$$

where $\Psi = \{\mathbf{f}, \mathbf{z}, \mathbf{v}\}$. The $\mathcal{L}(q)$ term in the above equation is often referred to as the *variational lower bound*.

Because $p(\mathbf{y})$ does not depend on q, maximizing $\mathcal{L}(q)$ effectively minimizes $D_{KL}(q||p)$. Furthermore, we apply the variational mean field approach which assumes the posterior distribution q factorizes with respect to the latent variables, and that the factors have the same functional form as the factors of p. Specifically,

$$\begin{aligned} p(\mathbf{z}, \mathbf{v}, \mathbf{f}|\mathbf{y}) &\approx q(\mathbf{z}) * q(\mathbf{v}) * q(\mathbf{f}) \\ &= \prod_{n=1}^{N} \prod_{m=1}^{M} q(z_{n,m}) \prod_{m=1}^{M} q(v_m) \prod_{m=1}^{M} q(f_m) \end{aligned} \tag{3.4}$$

where $q(v_m)$ is a beta distribution, $q(f_m)$ is a GP, and $q(z_{n,m})$ is a multinomial

distribution.

In the last line of Equation (4.10) we truncated the infinite products to contain M factors. The truncation is performed on the variational distribution q only, and could be considered as an additional constraint on its form. The original model remains unaffected and retains the infinite number of components. The truncation parameter M can be adjusted according to the needs of a particular application.

Under the assumption of Equation (4.10), the variational lower bound $\mathcal{L}(q)$ can be calculated as follows:

$$\begin{aligned}\mathcal{L}(q) = &\sum_{m=1}^{M} \int q(f_m) \log \frac{p(f_m)}{q(f_m)} df_m + \sum_{m=1}^{M-1} \int q(v_m) \log \frac{p(v_m)}{q(v_m)} dv_m \\ &+ \sum_{n=1}^{N} \sum_{m=1}^{M} q(z_{n,m}) \Big\{ \int q(v) \log p(z_{n,m}|v) dv \\ &- \log q(z_{n,m}) + \int q(f_m) \log p(y_n|f_m) df_m \Big\}\end{aligned} \tag{3.5}$$

A standard DPMGP model would infer the parameters by maximizing the variational lower bound $\mathcal{L}(q)$ in (4.3). In our DI-DPMGP model, we modify the objective function by maximizing $\mathcal{L}(q)$ and minimizing S at the same time:

$$\text{Maximize} \qquad \mathcal{L}(q) - \omega * \mathrm{E}(S) \tag{3.6}$$

where S is the total number of violations defined in Equation (3.3), $\mathrm{E}(S)$ is the expected value of S and ω is a trade-off parameter between $\mathcal{L}(q)$ and the penalty term $\mathrm{E}(S)$ which can be set through a grid search on a validation set. We select the ω that minimizes the prediction error on a validation set.

Note that the second term in (3.6) is $\mathrm{E}(S)$ rather than S because we are inferring the distribution of variable $\mathbf{z}$ (i.e., $q(\mathbf{z})$) and S is a function defined on two rows of a specific realization of $\mathbf{z}$. Following the same notation as in Section 3.4 we obtain:

$$\mathrm{E}(S) = \sum_{1 \le i \le j \le N} \widetilde{C}_{i,j} * \mathrm{E}[U(i,j)]$$

where

$$\begin{aligned}\mathrm{E}[U(i,j)] &= p(\langle z_i, z_j\rangle = 1)\\ &= \sum_{m=1}^{M} p(z_i = m \ \ \& \ \ z_j = m)\\ &= \sum_{k=1}^{M} \widehat{\pi}_{i,m}\widehat{\pi}_{j,m} = \widehat{\Pi}_i\widehat{\Pi}_j^T\end{aligned}$$

and M is the total number of clusters in the variational approximation. $\widehat{\Pi}_n = (\widehat{\pi}_{n,1}, \widehat{\pi}_{n,2}, \ldots, \widehat{\pi}_{n,M})$ are the probabilities that the nth instance belongs to the mth cluster $\forall 1 \leq n \leq N$ and $1 \leq m \leq M$. Thus, (3.6) can be re-written as our final optimization formulation:

$$\text{Maximize } \ \mathcal{L}(q) - \omega \sum_{1\leq i\leq j\leq N} \widetilde{C}_{i,j} * \widehat{\Pi}_i\widehat{\Pi}_j^T \tag{3.7}$$

where N is the total number of training instances. Setting the partial derivatives to zero in (3.7) with $\mathcal{L}(q)$ defined in (4.3), we obtain the modified inference update rules for each of the variational distributions in q:

$$q^*(\mathbf{f}) = \prod_{m=1}^{M} \mathcal{N}(f_m|\mu_m, \Sigma_m)) \tag{3.8}$$

$$\text{where} \qquad \Sigma_m = (K_m(\mathbf{x},\mathbf{x})^{-1} + B_m)^{-1}, \ \ \mu_m = \Sigma_m B_m \mathbf{y}$$

$$B_m = \frac{1}{\sigma^2}\begin{bmatrix} \widehat{\pi}_{1,m} & 0 & \ldots & 0 \\ 0 & \widehat{\pi}_{2,m} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \widehat{\pi}_{N,m} \end{bmatrix}$$

$$q^*(\mathbf{v}) = \prod_{m=1}^{M} \text{Beta}\left(v_m|1 + \sum_{n=1}^{N}\widehat{\pi}_{n,m}, \alpha + \sum_{j=m+1}^{M}\sum_{n=1}^{N}\widehat{\pi}_{n,j}\right) \tag{3.9}$$

$$q^*(\mathbf{z}) = \prod_{n=1}^{N}\prod_{m=1}^{M} \widehat{\pi}_{n,m}^{z_{n,m}} \tag{3.10}$$

where $\widehat{\pi}_{n,m} = p(z_{n,m} = 1) = \rho_{n,m} / \sum_{m=1}^{M} \rho_{n,m}$

$$\ln \rho_{n,m} = \Bigg(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \Big((y_n - [\mu_m]_n)^2 + [\Sigma_m]_{n,n} \Big) + \mathrm{E}_{\mathbf{v}}(\ln v_m) + \sum_{j=1}^{m-1} \mathrm{E}_{\mathbf{v}}(\ln(1 - v_j)) \Bigg) + \omega \sum_{i=1}^{N} \widetilde{C}_{n,i} \widehat{\pi}_{i,m}$$

We update (3.8), (3.9), (3.10) iteratively until the algorithm converges or reaches a maximum number of iterations.

Having calculated the latent variables, we can forecast the regression value y_* on a new input point x_* and provide a measure of uncertainty around y_* as follows:

$$y_* = \sum_{m=1}^{M} (\pi_m * f_m(x_*)))$$

$$\sigma^2_{y_*} = \sum_{m=1}^{M} (\pi^2_m * (\sigma^m_*)^2)$$

3.4.4 Modified k-means Algorithm

In order for the k-means algorithm to adjust its clustering decisions with respect to the signals from the DPMGP process, we modify the centroid re-assignment step to penalize the total number of violations defined in Equation (3.3). In particular, an instance x is assigned to a centroid μ if the sum of Euclidean distance from x to μ and the total number of violations x incurs while associated with μ is minimal compared to assignments to other centroids. Thus, given centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$, cluster J_i $(i = 1, 2., \ldots, k)$ at each iteration t is defined as follows:

$$J^t_i = \{x_p : ||x_p - \mu^t_i||^2 + \gamma \sum_{x_q \in S^t_i} \widetilde{C}_{p,q} * U(p,q) \leq ||x_p - \mu^t_j||^2 + \gamma \sum_{x_q \in S^t_j} \widetilde{C}_{p,q} * U(p,q) \quad \forall q, 1 \leq q \leq k\}$$

where $\widetilde{C}_{p,q} = 1 - C(p,q)$ and γ is the trade-off parameter between the Euclidean distance and the total number of violations, which can be set through a grid search on a validation set.

We integrate the modified k-means clustering process as an essential step in

	Initial EDSS score	# Instances
G1	< 2	370
G2	≥ 2 and <4	159
G3	≥ 4	45

Table 3.1: Experts' estimate of disease progression subgroups of MS data

	Age	Gender	# Instances
G1	≤ 65	M	2097
G2	≤ 65	F	885
G3	> 65	M	1911
G4	> 65	F	982

Table 3.2: Experts' estimate of disease progression subgroups of Parkinson's data

the execution of the modified DPMGP algorithm. In particular, a complete modified k-means clustering is performed within each iteration of the DPMGP model. The two algorithms provide updated feedback to each other via the total violations defined in (3.3).

3.5 Experimental Results

In this section, we first describe our motivating task of predicting MS disease progression. We then illustrate how domain knowledge leads DI-DPMGP to more accurate prediction in two medical datasets.

3.5.1 Predicting Multiple Sclerosis Disease Progression

As discussed in Section 3.1, our task is to predict the actual disability level of MS patients at the five year mark using their first two years' longitudinal data so that patients who have a likelihood of severe disability in five years can be treated more aggressively and monitored more closely.

Our data includes 574 patients enrolled in the CLIMB study [30] who have been monitored for five years. Each patient's data include demographic (e.g., age and gender) and radiologic data (e.g., lesion volume and brain parenchymal fraction). In addition, patients have a clinical visit every six months, which includes a complete neurological exam including a measurement of the subject's disability based on the expanded disability status scale (EDSS). As discussed in Section 2.2, the EDSS is known to have important interrater variability that can hinder analysis [38].

3.5.2 Experimental Method

All experiments were conducted by running five independent 10-fold cross-validations [13], and the calculated accuracies are the averages of the five executions of the program. The performance is measured using Mean Absolute Error (MAE) which is the average of all absolute differences between the predicted and true values. The hyper-parameters in this study were selected via grid search with lowest MAE on a validation set. In particular, the candidates for the trade-off parameter are $\omega \in [0.2, 0.5, 0.8]$ and for GP kernel parameters are $\sigma, l \in [1, 5, 10]$. We set the variational approximation parameter to $M = 20$.

We compare our method DI-DPMGP, which learns the level-1 clusters and level-2 prediction model simultaneously, against two baselines: DPMGP, which is a one-level non-parametric mixture of GPs model and Individual, which first obtains the level-1 clusters by partitioning the data using physician's estimate and then learn the level-2 prediction model by applying an individual DPMGP model separately to each level-1 cluster.

3.5.3 Multiple Sclerosis Dataset

We first applied the DI-DPMGP model to our motivating domain in predicting MS disease progression. The subgrouping criterion of MS disease was given by our experts using patients' initial EDSS scores. The total number of instances in each group is presented in Table 3.1. In the left plot of Figure 3.2, the two left bars (blue and red) are the MAE for the entire dataset and a weighted (by the # of patients) MAE calculated from the three subgroups. Lower MAE indicate better performance. We observe that partitioning the data according to our domain knowledge can improve performance over DPMGP. We then applied our DI-DPMGP model to the entire dataset and plotted its MAE as the rightmost bar (green) in the same graph. Our model offered 11.4% improvement over DPMGP (leftmost). Another notable observation is that the DI-DPMGP model had superior performance to the average of individual models (middle bar). This suggests that, by maximizing the

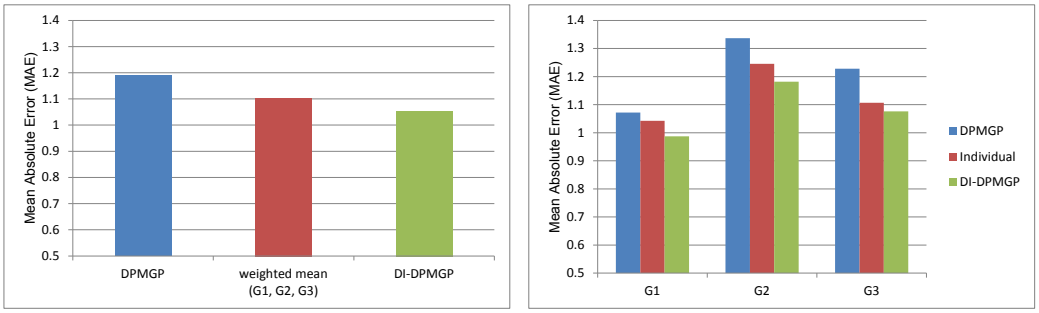


Figure 3.2: Comparison of DPMGP, Individual DPMGP, and DI-DPMGP approaches on MS data

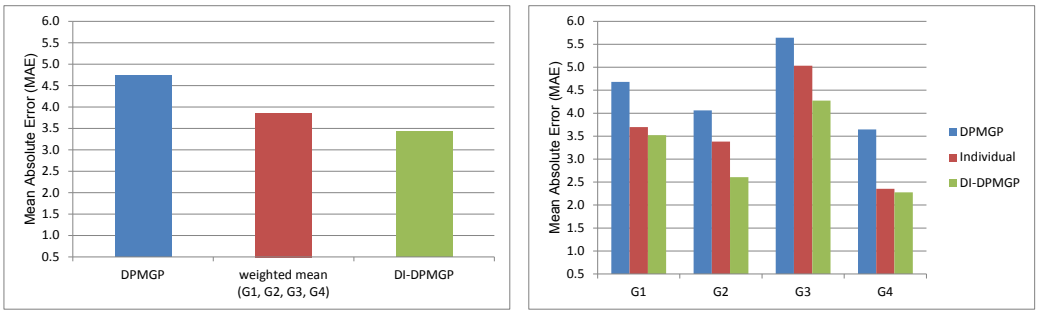


Figure 3.3: Comparison of DPMGP, Individual DPMGP, and DI-DPMGP approaches on Parkinson's data – motor UPDRS.

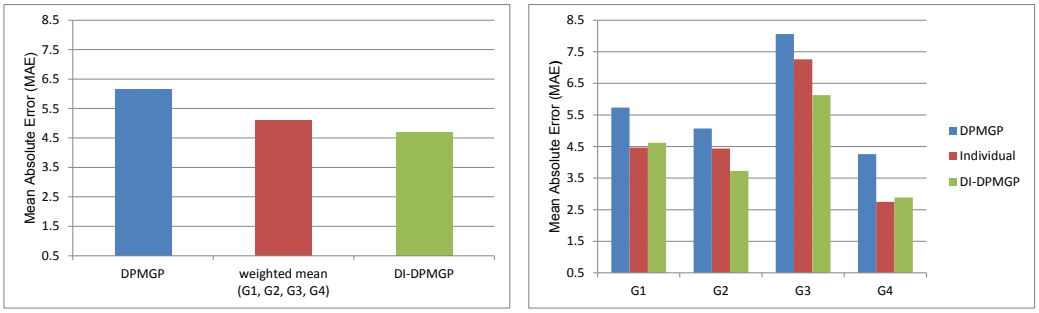


Figure 3.4: Comparison of DPMGP, Individual DPMGP, and DI-DPMGP approaches on Parkinson's data – total UPDRS.

consistency of clustering signals from both the data and the domain knowledge, our approach can realize a better partition of the data than using them separately.

To ensure the overall performance gain was not due to a particular subgroup, we investigated the behavior of each individual group across the models. To this end, we divided the MAE of an entire dataset into MAEs for G1, G2 and G3 and compared them to the MAE of the corresponding subgroups. The right graph in

Figure 3.2 presents the performance comparison for each group. In each group, the leftmost (blue) and the rightmost (green) bars are the breakdown MAEs from the baseline and DI-DPMGP models respectively; the middle bar (red) is the MAE from the DPMGP model trained by the group data only. Our first observation is that individual DPMGP (middle bars) consistently outperformed the corresponding DPMGP applied to all samples (leftmost bars), which confirms that our domain knowledge helps. Secondly, DI-DPMGP (rightmost bars) consistently outperformed the corresponding DPMGP on all data, offering 7.9%, 11.6% and 12.4% improvement for G1, G2 and G3 respectively. Furthermore, our model consistently outperformed the corresponding individual DPMGP.

3.5.4 Parkinson's Disease Dataset

Next, we evaluate our approach on a real world dataset from the UCI machine learning database [52]. We chose the Parkinson's disease dataset [81] because it contains similar physician subjectivity and patient bias as the MS dataset. This dataset consists of 5875 instances of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease recruited to a six-month trial of a telemonitoring device for remote symptom progression monitoring. The target values are the *motor UPDRS* and *total UPDRS* scores assigned by clinicians. The unified Parkinson's disease rating scale (UPDRS) is used to follow the longitudinal course of Parkinson's disease. The UPD rating scale (similar to EDSS in MS) is the most commonly used scale in the clinical study of Parkinson's disease. We tested our model in predicting both *motor UPDRS* and *total UPDRS* scores.

For Parkinson's disease, we have the domain knowledge that it has different characteristics for patients with different age and gender. Hence, we first manually divide the dataset into four subgroups shown in Table 3.2. Figures 3.3 and 3.4 present the results of predicting *motor ability* and *total ability* respectively. We observed similar conclusions as the MS dataset in predicting the target values. For overall performance (left graphs in Figure 3.3 and 3.4), DI-DPMGP (rightmost bars) offered 28% and 24% improvement over the baseline DPMGP (rightmost bars) re-

spectively. DI-DPMGP also outperformed the weighted (by # of instances) average performance (middle bars) calculated from individual DPMGP models.

For the breakdown performance of each individual group across all models, we observe in the right graphs of Figure 3.3 and 3.4 that DI-DPMGP (rightmost bars) consistently outperformed the corresponding DPMGP applied to all samples (leftmost bars). In predicting *motor UPDRS*, DI-DPMGP offered 25%, 36%, 24%, 37% improvement for G1-G4 respectively. In predicting *total UPDRS*, DI-DPMGP offered 19%, 27%, 24%, 32% improvement for G1-G4 respectively. Furthermore, for G2 and G3, our model outperformed the corresponding individual DPMGP models; whereas, for G1 and G4, it achieved comparable results.

3.6 Conclusion

We developed a domain induced Dirichlet mixture of Gaussian process model (DI-DPMGP) to incorporate domain knowledge in a non-parametric generative model. In particular, our model simultaneously addresses various types of bias and domain-specific information in medical data. For the former, in order to have the data determine the number of mixing components, we used the Dirichlet process-based clustering and estimated its parameters via variational inference. For the latter, we applied a k-means algorithm because we know the pertinent attributes and total number of subgroups. The modified DPMGP and k-means algorithms communicate via a penalty term to maximize the consistency of groupings induced by the data and by the experts' opinion resulting in a two-level hierarchical clustering structure. We evaluated our model on 1) our motivating domain of predicting disease progression in a cohort of Multiple Sclerosis patients and 2) a Parkinson's disease dataset [81]. The experiments demonstrated consistent efficacy of the new model. Our approach can be extended to other applications where domain knowledge is available to simultaneously enhance prediction and learning subgroups in a non-parametric mixture model.

In next chapter, we present our research in another medical domain: detect-

ing epileptogenic lesions in epileptic patients using their brain MRI images. Our new application exhbits a different type of patient bias which we will address using similar techniques we adopted in this chapter, i.e., the Dirichlet process-based clustering. However, our new application also brings other unique challenges.

Chapter 4

A Non-parametric Mixture of Restricted Boltzmann Machines

4.1 Introduction

In the previous two chapters, we presented our research in applying machine learning methods to predicting disease course for multiple sclerosis patients. In this chapter, we present our work in another medical domain: detecting structurally abnormal cortical regions in epileptic patients. Our algorithm, which detects lesional areas using brain MRI images, serves as a pre-surgical tool to help define resection zones for epileptic patients. We focus our task on "MRI-negative" patients whose MRI scans are deemed normal by neuroradiologists and, thus, have a low post-surgery success rate [80] attributing to imprecise resections caused by pre-surgical failure in visual identification of lesional regions. By comparing to MRI images of healthy human brains, we conjecture that machine learning algorithms may detect abnormal regions that human eyes can not find. Similar to our MS task, our dataset contains biases although, in this case, they originate from different morphometry of human brains due to different demographics or lifestyles. Because we do not have a division of primary versus auxiliary datasets in this task, we take a similar approach as in Chapter 3 and employ a Bayesian non-parametric mixture model to represent

the bias subgroup while, at the same time, addressing another unique challenge associated with this application: limited available features obtained from the MRI images.

Epilepsy is a common neurological disorder, affecting approximately 1% of the population [33]. It is characterized by profound abnormal neural activity during seizures and interictal (between seizures) periods. Uncontrolled epilepsy can have harmful effects on the brain and has increased risk of injuries and sudden death [7]. About one third of epilepsy patients remain resistant to medical treatment [48]. Our research addresses the identification of lesions in the MRI's of patients with focal cortical dysplasia (FCD), which is recognized as the most common source of pediatric epilepsy [7, 79] and the third most common source in adults having medically intractable seizures [47, 50]. Early detection and subsequent surgical removal of the FCD lesion area is the most effective and is often the last hope for these patients.

The most widely used technology in identifying the epileptic lesions is MRI coupled with intracranial EEG (iEEG). First, a panel of experienced radiologists visually inspect the patient's MRI to identify the lesion(s) and trace the intended resection zone. Then sub-dural electrodes are implanted on the cortical surface within this zone to record electrical activity [95]. A board of certified epileptologists reviews this information to determine the region that is responsible for generating the seizure, i.e., the seizure onset zone. To isolate the abnormal region, each electrode is labeled as being part of the seizure onset zone or not. The final target for surgical resection is based on both of these findings coupled with other clinical data. iEEG has been shown to be effective for localizing FCD lesions [55]. For *MRI-positive* patients, (i.e., patients with visible abnormal areas in the MRI), the placement of electrodes on the cortex is informed by the pinpointed problematic regions detected by visual inspection of the MRI. However, for *MRI-negative* patients, there is no visible lesion to guide precise electrode implantation [33], which results in sampling errors. In such cases the identified abnormal region fails to capture the lesion in its entirety in about 40% of the cases [39], resulting in poor surgical out-

comes. Consequently, the post-surgical success (i.e., seizure-freedom after surgery) ratio of MRI-positive to MRI-negative patients is 66% to 29% [80]. Unfortunately, 45% of FCD patients are MRI-negative [87]. For this reason the surgical resection procedure remains highly underutilized as most practitioners are unwilling to operate in the absence of a visually detected lesion [78].

Our task is to detect the lesional region(s) in MRI-negative patients [96]. Specifically, our model identifies the abnormal areas in a patient's brain which in turn serve as a focus of attention mechanism for the neuroradiologists in placing the iEEG sensors on the patient's cortex. The work presented in this chapter adopts a new Bayesian non-parametric approach as compared to the previous logistic regression (LR)-based model [3]. The LR model has been in use at NYU's Comprehensive Epilepsy Center since 2013. The new approach, as we present in Section 4.5, achieves improved performance compared to the LR model.

The training data comes from the 3D-MRI images of MRI-negative FCD patients who underwent resective brain surgery at NYU and were seizure free after surgery. Furthermore, the resected tissue was histopathologically verified to contain FCD. We also have access to the MRI's of healthy controls who underwent the same MRI protocol. One challenge in applying machine learning to this dataset is the paucity of features available from the MRI that are predictive of lesional tissue (see Section 4.2.1 for a complete list of the available features). To address this issue, we employ the Restricted Boltzmann Machines (RBMs) to learn a new set of nonlinear features [76], [94]. An RBM is an undirected graphical model [43] which can be interpreted as a stochastic neural network. Through training, an RBM learns a closed-form distribution of the input data. The units in the bottom layer of the network correspond to the observable attributes and the top layer consists of hidden units that can be viewed as nonlinear feature detectors [37]. Thus, an RBM is often employed to extract more meaningful features from input data [71].

Another challenge while learning with this dataset is the impact of inter-patient variability, i.e., each human brain has its own characteristics depending on the severity of the disease, age, gender, left/right handedness, lifestyle and genetics

many of which are not uniformly available in our data [73]. Thus, learning from data collected from all patients does not lead to satisfactory performance. To address this issue, we partition the data into subgroups with the goal that each subgroup will contain instances with similar brain characteristics. Similar to the approach we adopted in Chapter 3, because we do not have sufficient domain knowledge to estimate the number of subgroups, we employ a Bayesian non-parametric Dirichlet process mixture model (DPM, [77]) to infer the number of clusters automatically from the data.

Our proposed approach leads to an infinite mixture of RBMs. Because a typical mixture (finite or infinite) of RBMs is computationally intractable [57], we propose a two-stage method. We first apply an RBM to the entire dataset and cluster the patients based on the hidden layer of this model using a DPM. The second stage applies a weighted version of RBM to each non-empty cluster using the weights obtained from the DPM model. We discuss the rationale behind this two-stage method in Section 4.4.1.

We evaluate our model on MRI-negative patients from NYU who were seizure free after resective surgery. Our evaluation includes a successful detection indicator (Y or N) within the resected region and the performance (true negative rate, TNR) outside the resection zone. We need to measure the performance of the two regions separately because we not only want to locate the lesion but also don't want to misclassify non-lesional benign tissue. Note that, instead of using the true positive rate (TPR), we use an indication to measure whether there is any lesion detected inside the resection zone. This is because in the absence of any visual information to locate the lesion precisely, very generous margins are employed during the surgery to ensure that the patient is seizure free afterwards. As a result, the TPR is not a meaningful evaluation metric because the resection zone is much larger than the actual lesion and contains benign areas. Instead, we are interested in assessing whether there is any detection that can potentially guide the placement of iEEG sensors with the ultimate goal of reducing the size of the resection zone, which will lead to a much safer yet effective surgery. In our experiments, our model has

accuracies of greater than 99.4% in the benign region and a successful detection in the resection region in seven out of twelve (i.e., 58%) patients (see details in Section 4.5). This is in contrast to neuroradiologists' visual MRI inspections that have a detection rate of zero out twelve.

We first, in Section 4.2, describe our method of constructing training data from 3D-MRI images of human brains using surface morphometry [22, 28]. We then present a brief introduction to RBM in Section 4.3. For an introduction to the DPM model, please refer to Section 3.4.3 in Chapter 3. In Section 4.4, we outline our approach which integrates the RBM and DPM algorithms. We present and discuss our results in Section 4.5 and conclude in Section 4.6.

4.2 Surface-based Morphometry

Surface-based morphometry (SBM) provides the means to characterize and analyze the human brain by explicitly modeling the cortex using a suitable geometric model [22]. The cortical surface represents the outer layer of the brain modeled as a folded two-dimensional surface. It is extracted by delineating the boundary between the gray and white matter using T1-weighted MRI images [22]. The reconstructed surface is represented as a triangulated surface [22], and at each vertex on the surface different morphological features such as cortical thickness, curvature, sulcal depth, etc., can be calculated to characterize the cortex. Similarly, different morphological transforms can be applied to register the cortical surface to a standard surface also known as a group-atlas. Registration is achieved by aligning specific sulcal and gyral landmarks across the *reconstructed* cortical surfaces allowing for a precise comparison of individual cortical structures across subjects [27]. SBM has been used successfully for analyzing and detecting neurological abnormalities in various neurological disorders such as schizophrenia [69], autism [60], and epilepsy [79, 39].

4.2.1 Feature Extraction

In this work we use five features to represent each vertex on the reconstructed surface:

1. *Cortical thickness* represents the thickness of the cortex which is defined as the distance between the gray/white matter boundary and the outermost surface of the gray matter (pial surface). It is calculated at each vertex using an average of two measurements [27]: (a) the shortest distance from the white matter surface to the pial surface; and (b) the shortest distance from the pial surface at each point to the white matter surface.

2. *Gray/white-matter contrast (GWC)* represents the degree of blurring at the gray/white-matter boundary. GWC is estimated by calculating the non-normalized T1 image intensity contrast at 0.5mm above and below the gray/white boundary with trilinear interpolation of the images. The range of GWC values lies in $[-1, 0]$, with values near zero indicating a higher degree of blurring of the gray/white boundary.

3. *Sulcal depth* characterizes the folded structure of the cortex. It is estimated by calculating the dot product of the movement vectors with the surface normal [28], and results in the calculation of the depth/height of each point above the average surface. The values of sulcal depth lie in the range $[-2, 2]$ with lower values indicating a location in the sulcus whereas higher values indicate a location on the gyral crown.

4. *Curvature* is measured as $\frac{1}{r}$, where r is the radius of an inscribed circle and mean curvature represents the average of two principal curvatures with a unit of 1/mm [66]. Mean curvature quantifies the sharpness of cortical folding at the gyral crown or within the sulcus, and can be used to assess the folding of small secondary and tertiary folds in the cortical surface.

5. *Jacobian distortion* measures the magnitude of the nonlinear transform needed to wrap each vertex on the subject's brain to a target vetex on the average

surface, as part of the registration process. This nonlinear transform is needed to align the gyral and sulcal landmarks between the source and the target brain. Jacobian distortion is a measure of global brain deformation, and has been used to characterize the cortex for analyzing a number of neurological disorders [32].

4.2.2 Automated FCD Lesion Detection

The machine learning task is to develop a classifier that can distinguish between normal and abnormal cortical tissue using the patient's MRI data. SBM has been used in conjunction with machine learning and statistical techniques to identify lesions in FCD patients. Besson et al. [8] use texture, GWC and a number of morphological features including cortical thickness to represent each vertex on the reconstructed cortical surface. They then train a neural network to classify each vertex as being normal or lesional. Similarly, Thesen et al. [79] use a univariate z-score-based thresholding approach on registered SBM data to classify each vertex as being lesional or normal. Recently, Hong et al. [39] developed a two-stage Fisher linear discriminant analysis (LDA) [9] classifier to detect FCD lesions in MRI-negative patients. Initially they train a vertex-level classifier that classifies each vertex on the reconstructed cortical surface as being lesional or non-lesional for both controls and patients. These detections are further refined using another LDA classifier that is trained to distinguish between actual FCD lesions (detections made inside the manually refined resection zones of patients) and false lesional detections made on controls.

One of the major confounding factors inhibiting the development of an effective classifier for detecting FCD lesions is *inter-patient variability.* The morphology of the human brain such as its thickness, curvature and the overall structure in general are affected by different factors such as age, gender, handedness, etc. [73]. This causes a co-variate shift in data as the data from different patients is pooled to learn a common classifier. Similarly, the distribution of pathological features that define an FCD lesion differs across FCD subtypes. For example in addition to

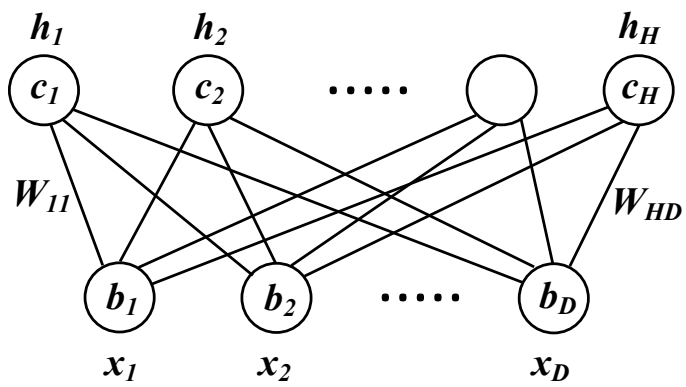


Figure 4.1: An RBM with D visible units and H hidden units

causing other morphological abnormalities, FCD type I lesions appear on MRI as abnormally thin regions of cortex, while FCD type II is characterized by abnormally thick regions. This heterogeneity of feature distributions that define the target concept (FCD lesion) for the learner must be taken into account to develop an effective supervised lesion detection scheme.

To counter the effects of differences in feature distributions across FCD subtypes, Hong et al. deal only with FCD type-II [39], while Ahmed et al. [3], stratify the training data into thick and thin lesions based on manual inspection of data, while other schemes bypass this training bias by posing lesion detection as an outlier/anomaly detection problem [79, 4]. However, none of the lesion detection schemes cited previously explicitly address the co-variate shift arising from inter-patient variability. To overcome this co-variate shift in the underlying data distributions, we train an ensemble of classifiers on a training set consisting of an equal number of patients from each of the three FCD subtypes, and control data taken from fifty neurotypical controls. In order to discover subgroups in data, that align with meaningful combinations of patient and FCD subtype characteristics we use a Dirichlet process-based mixture of RBMs.

4.3 Restricted Boltzmann Machines

Before describing our method in detail, we first give an introduction to restricted Boltzmann machine.

4.3.1 Definition

A restricted Boltzmann machine (RBM, [76, 94]), illustrated in Figure 4.1, is an undirected graphic model that consists of two layers: a visible layer $\mathbf{x} = \{x_1, x_2, \ldots, x_D\}$, which represents the attributes of the input data, and a hidden layer $\mathbf{h} = \{h_1, h_2, \ldots, h_H\}$. Units across different layers are fully connected. However, connections within the same layer are restricted. The goal of an RBM is to model the distribution of the observations $\mathbf{x}$ with the help of hidden units $\mathbf{h}$.

We define the energy of an RBM as:

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{h}^{\mathbf{T}}\mathbf{W}\mathbf{x} - \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{h} \tag{4.1}$$

where $\mathbf{W} \in R^{H \times D}$ represents the weights connecting hidden and visible units and $\mathbf{b} \in R^D$, and $\mathbf{c} \in R^H$ are the offsets of the visible and hidden layers respectively. By marginalizing out the variable h in $E(\mathbf{x}, \mathbf{h})$ defined in (4.1), we can obtain the following formulation for the probability density of $\mathbf{x}$:

$$p(\mathbf{x}) = e^{-F(\mathbf{x})}/\mathcal{Z} \tag{4.2}$$

where

$$F(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \sum_{j=1}^{H} \ln\left(1 + e^{(b_j + \mathbf{W}_j\mathbf{x})}\right) \tag{4.3}$$

$F(\mathbf{x})$ is often referred to as the free energy. H is the total number of hidden units and $\mathcal{Z} = \sum_{\mathbf{x}} e^{-F(\mathbf{x})}$ is the partition function. It can be shown that the conditional probability of h_i's given $\mathbf{x}$ is independent, i.e.,

$$p(\mathbf{h}|\mathbf{x}) = \prod_j p(h_j|\mathbf{x}) \tag{4.4}$$

and furthermore,

$$p(h_j = 1|\mathbf{x}) = \frac{1}{1 + exp(-(b_j + \mathbf{W}_j\mathbf{x}))} = \sigma(b_j + \mathbf{W}_j\mathbf{x}) \tag{4.5}$$

where $\sigma(\cdot)$ is the sigmoid function. Similarly, we have

$$p(\mathbf{x}|\mathbf{h}) = \prod_k p(x_k|\mathbf{h}) \tag{4.6}$$

$$p(x_k = 1|\mathbf{h}) = \frac{1}{1 + exp(-(c_k + \mathbf{h}^T\mathbf{W}_k))} = \sigma(c_k + \mathbf{h}^T\mathbf{W}_k) \tag{4.7}$$

Equations (4.4) - (4.7) allow us to make approximations to the inference algorithm described next.

4.3.2 Inference

Given the training data, the learning objective of an RBM is to adjust parameters $\{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ such that the free energy defined in (4.3) is minimized, which is equivalent to maximizing $p(\mathbf{x})$. A standard approach is to use stochastic gradient descent to minimize the average negative log-likelihood $\frac{1}{T}\sum_{t=1}^{T} -\ln p(\mathbf{x}^t)$. Thus, we calculate the partial derivatives with respect to each parameter and set them to zero to obtain a set of update rules. A learning algorithm iteratively updates the parameters until convergence. Specifically,

$$\begin{aligned} -\frac{\partial \ln p(\mathbf{x})}{\partial \theta} &= \frac{\partial F(\mathbf{x})}{\partial \theta} + \frac{1}{\mathcal{Z}}\frac{\partial \mathcal{Z}}{\partial \theta} \\ &= \frac{\partial F(\mathbf{x})}{\partial \theta} + \frac{1}{\mathcal{Z}}\sum_{\mathbf{x}} e^{-F(\mathbf{x})}\frac{\partial(-F(\mathbf{x}))}{\partial \theta} \\ &= \frac{\partial F(\mathbf{x})}{\partial \theta} - \sum_{\mathbf{x}} p(\mathbf{x})\frac{\partial F(\mathbf{x})}{\partial \theta} \end{aligned} \tag{4.8}$$

where $\theta \in \{\mathbf{c}, \mathbf{b}, \mathbf{W}\}$. The computational difficulty in (4.8) is the second term which is an expectation over an exponential number of configurations of the input $\mathbf{x}$

under $p(\mathbf{x})$. Hinton introduced the Contrastive Divergence (CD) learning algorithm [36] which makes this computation tractable by estimating the expectation using a sample, $\tilde{\mathbf{x}}$ from the model. This sample is obtained using k-steps of Gibbs sampling. Thus, (4.8) can be simplified to:

$$-\frac{\partial \ln p(\mathbf{x})}{\partial \theta} \approx \frac{\partial F(\mathbf{x})}{\partial \theta} - \frac{\partial F(\tilde{\mathbf{x}})}{\partial \theta} \tag{4.9}$$

For RBMs, because the hidden and visible units are conditionally independent (Equation (4.4), (4.6)), we can perform block Gibbs sampling, i.e., we sample all hidden (visible) units simultaneously given the values of the visible (hidden) units. In particular, starting with a given visible observation $\mathbf{x}^{(n)}$, we have:

$$\mathbf{h}^{(n+1)} \sim \sigma(\mathbf{W}'\mathbf{x}^{(n)} + \mathbf{c})$$

$$\mathbf{x}^{(n+1)} \sim \sigma(\mathbf{W}\mathbf{h}^{(n+1)} + \mathbf{b})$$

Consequently, the Contrastive Divergence learning algorithm with k steps of Gibbs sampling (CD-k) can be summarized as follows:

1. For each training example $\mathbf{x}^{(t)}$
   (a) Starting at $\mathbf{x}^{(t)}$, generate a sample $\tilde{\mathbf{x}}$ using k steps of Gibbs sampling.
   (b) Update parameters

$$\begin{aligned}
\mathbf{W} &\Longleftarrow \mathbf{W} + \alpha \left(h(\mathbf{x}^{(t)})\mathbf{x}^{(t)^T} - h(\tilde{\mathbf{x}})\tilde{\mathbf{x}}^T) \right) \\
\mathbf{b} &\Longleftarrow \mathbf{b} + \alpha \left(h(\mathbf{x}^{(t)}) - h(\tilde{\mathbf{x}})) \right) \\
\mathbf{c} &\Longleftarrow \mathbf{c} + \alpha \left(\mathbf{x}^{(t)} - \tilde{\mathbf{x}}) \right)
\end{aligned}$$

   where $h(\mathbf{x}) = \sigma(\mathbf{b} + \mathbf{W}\mathbf{x})$ and α is the learning rate.

2. Go to Step 1 until the stopping criterion is met.

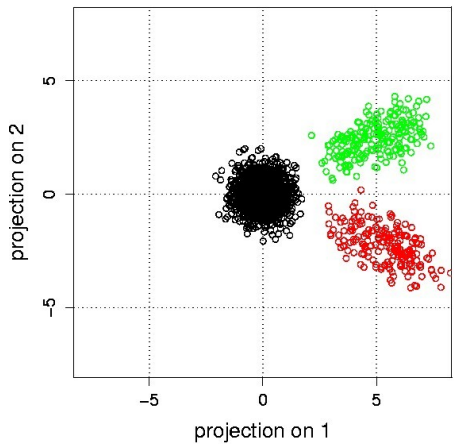


Figure 4.2: Projection of Clusters

We set $k = 1$ in our model because it has been shown empirically that even when k is not large (e.g., $k = 1$), the CD-k algorithms often gives good results [14].

4.4 RBM-DPM Model

In this section, we describe how to combine the RBM and DPM algorithms for our classification task, including a modified weighted RBM training algorithm.

4.4.1 Integrating RBM and DPM

The inference of a typical mixture (finite or infinite) of RBMs is intractable due to the difficulty of estimating the partition function $\mathcal{Z}$ in $p(\mathbf{x})$ defined in Equation (4.2). We propose to take a two-stage approach. We first employ a DPM model to partition the data into k clusters. We then apply a modified version of RBM training algorithm to each non-empty cluster using the weights obtained from the DPM model.

For stage one, we note that it is the top layer hidden units (i.e., the non-linear features extracted by the RBM) that serve as the input to our classifier. Thus, directly clustering in the input $\mathbf{x}$ may not be effective for a task for which the inputs are the $\mathbf{h}$'s. For example, in image processing, we can interpret the $\mathbf{h}$'s as a projection of $\mathbf{x}$'s on some other dimension. In Figure 4.2, the three clusters in the original

input space are no longer valid when projected to either of the two dimensions. Input data, therefore, should be partitioned according to the characteristics of the projections (i.e., the top layer hidden units) rather than the original observations.

We propose applying an infinite mixture model to cluster the data at the top layer of the neural network, i.e., the $\mathbf{h}$'s in an RBM, and then learn k classifiers for those partitions produced by the clustering algorithm. To predict the label for a new instance, we will feed the instance to each classifier and take a majority vote from their predictions. Our RBM-DPM model can be outlined as follows:

RBM-DPM Classification Model

INPUT: Instances $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

Labels $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_n\}$

1. Train an RBM R_0 $(\mathbf{W_0}, \mathbf{b_0}, \mathbf{c_0})$ using the entire input dataset $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$.

2. Compute the set of hidden units values, $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$, where $\mathbf{h_i} = \mathbf{W}\mathbf{x_i}+\mathbf{b}$ and $\mathbf{W}, \mathbf{b}$ are learned parameters from R_0.

3. Train a DPM mixture model (see Section 4.4.2) on $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$, producing k non-empty clusters.

4. Train k weighted RBMs (see Section 4.4.3) $R_1, R_2, \ldots R_k$ for each non-empty cluster from Step 3.

5. Transform each R_i $(i = 1, 2, \ldots, k)$ into a classifier by augmenting an output layer of units. Adjust parameters $(\mathbf{W}_i, \mathbf{b}_i, \mathbf{c}_i)$ using labels $\mathbf{L}$ and the backpropagation algorithm.

OUTPUT: Classifiers $R_1, R_2, \ldots, R_k$ on input domain $\mathbf{x}$.

4.4.2 Mixture of Hidden Units

In this section, we give the formulation of the DP mixture model of the hidden units $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$. It corresponds to Step 3 of the RBM-DPM model. We assume instances in the kth cluster follow a multivariate Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$. Our model employs the latent variables $\mathbf{z} = \{z_1, z_2, \ldots\}$ (z_i's are the rows of the indicator matrix $\mathbf{z}$ mentioned previously), $\mathbf{v} = \{v_1, v_2, \ldots, \}$ (v_i's are the stick-breaking proportions [74] in a DP) and $\Theta = \{\mu_1, \mu_2, \ldots, \Sigma_1, \Sigma_2, \ldots\}$. Hyper-parameters are α, μ_0 and Σ_0.

The joint distribution $p(\mathbf{h}, \mathbf{z}, \mathbf{v}, \Theta)$ can be factored as follows:

$$p(\mathbf{h}, \mathbf{z}, \mathbf{v}, \Theta) = p(\mathbf{h}|\mathbf{z}, \Theta)p(\Theta)p(\mathbf{z}|\mathbf{v})p(\mathbf{v}|\alpha) \tag{4.10}$$

where

$$\begin{aligned}
p(\mathbf{v}|\alpha) &= \prod_{k=1}^{\infty} \text{Beta}(v_k|1, \alpha) \\
p(\mathbf{z}|\mathbf{v}) &= \prod_{n=1}^{N} \prod_{k=1}^{\infty} \left(v_k \prod_{j=1}^{k-1} (1 - v_j) \right)^{z_{n,k}} \\
p(\Theta) &= \prod_{k=1}^{\infty} \mathcal{N}(\Theta_k|\mu_0, \Sigma_0) \\
p(\mathbf{h}|\mathbf{z}, \Theta) &= \prod_{n=1}^{N} \prod_{k=1}^{\infty} \mathcal{N}(\mathbf{h}_n|\mu_k, \Sigma_k)^{z_{n,k}}
\end{aligned}$$

This is a standard DPM of Gaussian distributions, for which we use the variational inference method [10, 11] to estimate the parameters. The inference algorithm iteratively computes the values of latent variables until convergence. In particular, the mixture model produces a soft mixture of $\mathbf{h}$'s as follows:

$$\begin{bmatrix} \pi_{1,1} \cdot \mathbf{h}_1 & \pi_{1,2} \cdot \mathbf{h}_1 & \dots & \pi_{1,k} \cdot \mathbf{h}_1 & \dots \\ \pi_{2,1} \cdot \mathbf{h}_2 & \pi_{2,2} \cdot \mathbf{h}_2 & \dots & \pi_{2,k} \cdot \mathbf{h}_2 & \dots \\ \vdots & \vdots & \ddots & \vdots & \\ \pi_{n,1} \cdot \mathbf{h}_n & \pi_{n,2} \cdot \mathbf{h}_n & \dots & \pi_{n,k} \cdot \mathbf{h}_n & \dots \end{bmatrix}$$

where each column is a cluster and $\sum_j \pi_{i,j} = 1$ for $i = 1, 2, \dots, n$.

We next proceed to the second stage, i.e., Step 4 of the RBM-DPM model, and train a weighted RBM (described below) for each non-empty cluster. The weights for cluster k are defined by the $\pi_{i,k}$'s ($i = 1, 2, \dots, n$).

4.4.3 Weighted RBM

The mixing proportions for the ith cluster are:

$$[\pi_{1,i}, \pi_{2,i}, \dots, \pi_{n,i}]^T$$

Because the $\mathbf{h}_i$'s are obtained from the corresponding $\mathbf{x}_i$'s, we modify the energy function using weighted $\mathbf{x}_i$'s:

$$F(\mathbf{x}_j) = \pi_{j,i} * \mathbf{c}^T \mathbf{x}_j + \sum_{j=1}^{H} \ln\left(1 + e^{(b_j + \pi_{j,i} * \mathbf{W}_j \mathbf{x}_j)}\right) \qquad \forall \mathbf{x}_j \in \mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$$

Consequently, we can modify the CD-k learning algorithm for cluster i as follows:

Weighted CD-k Learning Algorithm

1. For each training example $\mathbf{x}_j^{(t)}$ in cluster i

   (a) Starting at $\mathbf{x}_j^{(t)}$, generate a sample $\tilde{\mathbf{x}}_j$ using k steps of Gibbs sampling.

   (b) Update parameters

$$\mathbf{W} \Longleftarrow \mathbf{W} + \alpha * \pi_{j,i} * \left(h(\mathbf{x}_j^{(t)}) \mathbf{x}_j^{(t)^T} - h(\tilde{\mathbf{x}}_j) \tilde{\mathbf{x}}_j^T) \right)$$
$$\mathbf{b} \Longleftarrow \mathbf{b} + \alpha * \pi_{j,i} * \left(h(\mathbf{x}_j^{(t)}) - h(\tilde{\mathbf{x}}_j)) \right)$$
$$\mathbf{c} \Longleftarrow \mathbf{c} + \alpha * \pi_{j,i} * \left(\mathbf{x}_j^{(t)} - \tilde{\mathbf{x}}_j) \right)$$

   where $h(\mathbf{x}_j) = \sigma(\mathbf{b} + \mathbf{W}\mathbf{x}_j)$, α is the learning rate.

2. Go to Step 1 until stopping criterion is met.

4.5 Experimental Results

In this section we describe in detail the construction of our dataset, the machine learning techniques used to address domain-specific issues related to our task, and the performance evaluation of our proposed RBM-DPM model.

4.5.1 Patient and Data Description

As described in Section 4.1, we had access to the MRI data of MRI-negative patients from NYU's Comprehensive Epilepsy Center who underwent resective surgery and were completely seizure free after surgery. Furthermore, their resected cortical tissue was histopathologically verified to contain FCD. These patients are further categorized into three subtypes (type I, II and III) [12]. Six representative patients were selected from each subtype group resulting in a total of eighteen patients available to our research. This might seem like a small collection of patients, however it

Table 4.1: Epilepsy Type and Data Instances Contributed from Each Patient

Patient	Type	Positive Instances	Negative Instances
NY143	2	629	629 * 50
NY148	1	6,569	6,569 * 50
NY149	2	7,035	7,035 * 50
NY159	1	5,111	5,111 * 50
NY186	3	5,869	5,869 * 50
NY294	3	14,107	14,107 * 50
NY226	1	6,485	6,485 * 50
NY255	2	14,394	14,394 * 50
NY259	1	6,792	6,792 * 50
NY315	2	3,434	3,434 * 50
NY338	3	9,046	9,064 * 50
NY343	3	10,463	10,463 * 50
NY351	1	3,522	3,522 * 50
NY371	2	6,915	6,915 * 50
NY394	2	14,197	14,197 * 50
NY46	1	18,972	18,972 * 50
NY67	3	14,932	14,932 * 50
NY72	3	15,448	15,448 * 50
Total		163,920	8,196,000

should be noted that only a few MRI-negative patients proceed to surgical resection and out of these few only a third achieve complete post-surgical seizure freedom. *Indeed, the six type II patients in our collection represent the entire population of FCD type-II MRI-negative patients treated at NYU during the past three years.*

In terms of the actual data instances, each patient contributes a different number of positive instances to our dataset, depending on the size of his/her resection area. All the vertices within the resected region are labeled as positive (lesional, label = 1). The negative (non-lesional, label = 0) instances for our dataset are extracted from the MRI scans of fifty neurotypical healthy controls who underwent the same MRI protocol. In particular, for each patient, we extract data from each healthy image from the same location as the patient's resection region. As a re-

sult, if a patient contributes n lesional samples to our dataset, we will have fifty corresponding sets of non-lesional samples with n instances in each set (i.e., a total of $50 * n$ instances). Table 1 presents the subgroup type, total number of positive and negative instances associated with each patient. We have a total of 163,920 and 8,196,000 positive and negative instances respectively. Note that the negative instances in our dataset are taken from the healthy controls instead of from non-lesional areas outside the patient resection zones. This approach encourages our model to learn a more accurate representation of normal human brains, which is essential to our task of abnormality detection.

4.5.2 Constructing Training Set

The majority of the FCD patients in our dataset have temporal lobe resections, which is the most prevalent localization of FCD in adults [45]. Training on all patients would therefore be biased toward a specific cortical region limiting its generalization to differentiate between lesional and non-lesional vertices in other cortical regions. Training on all patients (and performing evaluation via leave-one-patient out cross validation) would result in low accuracy in regions other than the temporal lobe. Furthermore, it is desirable to have a balanced training set of patients from the three different FCD subtypes to give a good distribution over different FCD lesion types. Under these two constraints, we selected two patients from each FCD subtype (i.e., six patients in total) as our training patients such that their resected regions optimize coverage of the cortex beyond the temporal lobe.

4.5.3 Selective Ensemble of Classifiers

There is a fifty to one ratio between negative and positive instances in our data. In order to overcome this class imbalance, we apply bagging with under-sampling [85]. Each bag includes all positive instances from our training patients (i.e., all minority instances). At the same time, we randomly pick one control for each patient and include all the corresponding negative instances from the selected control. We repeat this process fifty times and create fifty balanced training sets. We learn a

classifier for each training set resulting in fifty independent classifiers. A standard ensemble method in machine learning will perform a majority vote among all fifty classifiers when making a prediction. In our case, however, we want learners that boast high TNRs on the training data. This is because, although we are aiming at a high lesion detection rate, our detection is only meaningful if we don't sacrifice the performance on non-lesional areas. To address this preference issue specific to our task, we establish an *Ensemble Threshold* and discard classifiers whose TNRs fall below this threshold *on the training data*. The goal is to weed out classifiers that fail to capture the characteristics of healthy cortical tissue.

4.5.4 Evaluation Method

We have twelve test patients (excluding the six training patients) and we present our model's performance on each test patient's entire brain i.e., vertices from both inside and outside the resected region.We measure the performance on the lesional and non-lesional instances separately to ensure that we are not only detecting the lesions, but also are not misclassifying normal cortical vertices. As discussed in Section 4.1, we use an indication flag to signal a successful detection within the resection zone and a TNR to measure our performance on non-lesional instances. Note that the benign instances of the patients are not part of our dataset constructed in Section 4.5.1. Thus, a good performance on these instances (i.e., a high TNR) demonstrates the efficacy of our model in recognizing new healthy brain structures.

We compare the performance of our RBM-DPM model to two baseline models. The first is our recently reported logistic regression (LR)-based approach [3] which deals exclusively with MRI-negative patients. This model has been in use at NYU's Comprehensive Epilepsy Center since 2013 and entails a number of pre-processing steps which include manual reduction of the resected region, stratifying the data based on sulcal depth, and a post-processing step that manually discards all detected clusters that fall below a surface area of $50mm^2$. The performance of this model reported in Table 4.2 which includes all the pre-processing and post-processing steps, and thus represents this method's best performance.

Table 4.2: RBM-DPM Model Compared to LR and RBM Models with Different Ensemble Thresholds

Patient	Threshold = 90						Threshold = 95					
	LR		RBM		RBM+DPM		LR		RBM		RBM+DPM	
	-		# Classifiers*: 18		# Classifiers: 25		-		# Classifiers: 12		# Classifiers: 17	
	Detected	TNR	Detected	TNR	Detected	TNR	Detected	TNR	Detected	TNR	Detected	TNR
NY226	Y	96.3	Y	99.0	Y	99.6	Y	98.7	N	98.3	Y	99.9
NY255	Y	96.8	Y	97.5	Y	99.6	Y	98.6	Y	99.3	Y	99.9
NY259	N	96.3	Y	98.3	Y	99.7	N	99.4	N	97.6	Y	99.9
NY315	N	97.7	Y	97.9	N	99.6	N	98.5	N	99.0	N	99.9
NY338	Y	98.5	Y	97.6	Y	99.4	N	99.8	Y	99.3	Y	99.8
NY343	Y	97.1	Y	97.6	Y	99.6	Y	99.1	Y	98.4	Y	99.9
NY351	N	97.9	N	97.2	N	99.7	N	98.3	N	98.3	N	99.9
NY371	Y	97.5	Y	97.2	N	99.6	Y	99.5	Y	98.1	N	99.9
NY394	N	98.1	N	97.5	N	99.7	N	99.6	N	98.1	N	99.9
NY46	Y	97.4	Y	98.7	Y	99.7	Y	99.2	Y	97.9	Y	99.9
NY67	Y	97.3	Y	98.6	Y	99.6	Y	99.3	Y	99.3	Y	99.8
NY72	N	98.1	N	99.8	N	99.8	N	99.7	N	98.4	N	99.9
Mean	58%	97.4	75%	98.1	58%	99.6	50%	99.1	50%	98.6	58%	99.9

*Number of retained classifiers after the selective ensemble described in Section 4.5.3.

For the second baseline model, we choose to train an RBM over the entire dataset without applying the DPM clustering algorithm to the data. The purpose of the comparison is to verify our conjecture that DPM is able to capture the interpersonal variations arising from different morphologies among the patients.

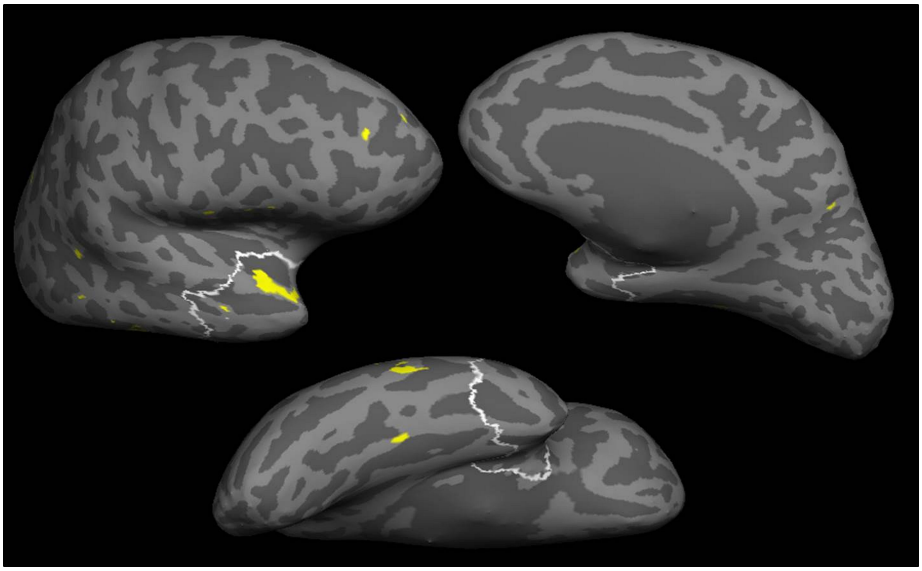
4.5.5 Discussion

Table 4.2 presents the main results of our model on the twelve test patients. We experimented with two thresholds (90 and 95, indicated by the first row) for retaining the classifiers (see details in Section 4.5.3). Lower thresholds are not interesting because they lead to lower TNR. For each threshold value, we compare the performance of our model to the two baseline models. The third row shows the number of classifiers retained in each case. Under columns "Detected", a value 'Y' indicates a successful detection. The corresponding entry in the last row shows the successful detection rate out of the twelve patients in each model.

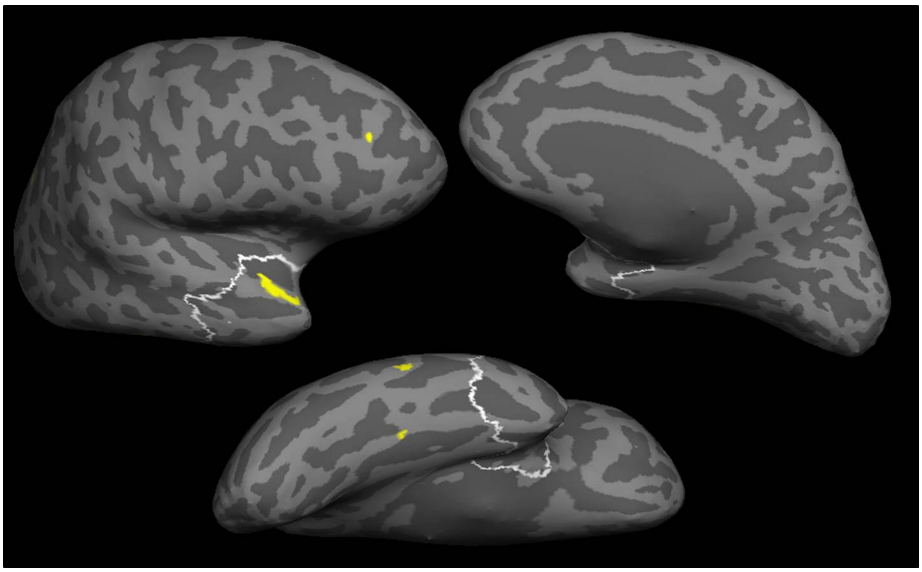
We have developed our approach keeping in mind its final use as a focus of attention mechanism for neuroradiologists. The detections made using our model would be used to inform effective electrode placement in iEEG, and constitute a source of secondary evidence for determining the resection target with an ultimate goal of reducing the resection region which leads to a safer and effective surgery. Thus, there are two performance metrics that need to be analyzed: i) performance on the non-lesional regions (i.e., TNR) and ii) detection rate on the lesional regions.

- A high true negative rate is essential in defining success in our task as it alludes to the number of false detections that a neuroradiologist needs to inspect. Indeed, a model with a TNR rate below 99% is impractical because there will be too many false positive regions. One way to increase the TNR rate is to raise the *Ensemble Threshold* to a more stringent value (e.g., from 90 to 95). As we observe in Table 4.2, the TNR rates improve across all models as we move the threshold from 90 to 95.

- The RBM-DPM approach dominates both RBM and LR models by delivering



(a) Threshold=90



(b) Threshold=95

Figure 4.3: Detection Results for NY255 using the proposed scheme, at two different thresholds of bag selection, plotted on the inflated cortical surface. The detected clusters are depicted as the solid yellow regions, while the white outlined region represents the actual resected region.

the highest TNR rates (99.6% and 99.9%) in both threshold settings. Although the RBM model has a higher detection rate at threshold 90, its corresponding TNR rate (98.1%) is too low to be effective for the neuroradiologists. The LR model achieves satisfactory TNR performance at threshold 95, which is the setting that is currently used by NYU's Comprehensive Epilepsy Center.

- Detection rate is defined as the number of patients for whom an abnormal region is detected within their resected region. Although the detection rates presented in Table 4.2 seem low (50% to 75%), it is worth noting that the detections are made from MRI images of MRI-negative patients, which means the doctors have no identification of any abnormality in these MRI images.

- We observe from Table 4.2, that RBM-DPM model delivers a more stable performance compared to the other two models at the two threshold settings. In particular, RBM-DPM maintained a detection rate of 58% (7/12 patients) when the threshold changed from 90 to a more stringent value 95. On the other hand, both RBM and LR models sufferred a drop in their detection rates in the process. For the RBM model, the detection rate changed from 75% to 50% and for the LR model, the change was from 58% to 50%. This stable performance coupled with its near perfect TNR accuracies makes RBM-DPM a desirable tool in the pre-surgical evaluation process.

- Even though resective surgery is a viable option for FCD patients, it remains underutilized as most practitioners are unwilling to perform surgery in the absence of a visually detected lesion. This limits the number of available patients whose data can be used to build automated lesion detection models. Our model has the potential to increase the number of patients who are referred to surgery by locating the lesion during the pre-surgical evaluation process. Although the current sample may seem small, the results are significant since a board of experienced neuroradiologists failed to locate the lesion for all these patients, and our approach is able to detect the lesion in 58% of the patients with a near perfect TNR.

Figures 4.3(a) and 4.3(b) plot the detection results of RBM-DPM model on a test patient for the two different threshold values using an inflated model of their cortical surface. The yellow areas are detected lesional regions from the model and the white outlined region indicates the actual resected area. Thus, yellow clusters within the resected zone are correct detections, while those outside the zone are false positives. As explained in Section 4.1, the resection zones are determined in a "generous" manner for MRI-negative patients to maximize the chances of a seizure free outcome. Indeed, we observe in Figures 4.3(a) and 4.3(b) that the detected lesional regions are considerably smaller than the size of the actual resection. Thus, the results of our model can be used to guide electrode placement during a patient's iEEG evaluation and obtain a potentially refined resection target, which leads to reduced chances of removing healthy cortical tissue.

4.6 Conclusion

We proposed a non-parametric approach to detect MRI-elusive epilepsy lesions using restricted Boltzmann machines (RBMs). In particular, we transform 3D-MRI images of human brains into a standard 2D surface using the Surface-based Morphometry methodology and extract five features that characterize human cortical surfaces. Our model addresses both issues of limited available features and inter-patient variabilities in the input data. For the former, we use an RBM as a pre-training step. For the latter, we apply a Dirichlet process-based clustering algorithm and estimate its parameters via variational inference. To accomplish our classification task, we collect multiple classifiers by training an augmented RBM for each non-empty component from the clustering algorithm and take a majority vote among all classifiers while making a prediction. We evaluated our model on brain images of twelve MRI-negative patients. Our model correctly detected abnormal regions within the resected areas in 58% of the patients, with 99.9% accuracy of correctly classifying the non-lesional vertices. Based on these findings, we are evaluating a replacement of the currently used LR model with our new approach in the clinical treatment for

epilepsy patients at NYU's Comprehensive Epilepsy Center.

In next chapter, we present our research in the educational domain: predicting potential performance of applicants in the graduate admission process. This application brings a new type of subjectivity in the data to which the previous transfer learning (Chapter 2) and Dirichlet process-based clustering (Chapters 3 and 4) techniques are not applicable. In addition, the data come in both labeled and unlabeled forms. We propose a new variant of semi-supervised SVM that addresses these challenges associated with our new task.

Chapter 5

Integrating Semi-supervised SVM with Domain Knowledge

5.1 Introduction

In this chapter, we present our research in a new application domain: a quantitative machine learning approach to master student admission at Northeastern University. In particular, we forecast an applicant's potential success in the graduate program using a model based on quantitative measures of previously admitted students. This application differs from the previous two in that it contains both labeled and unlabeled data. Indeed, we only have a limited amount of labeled data because only a small percentage of applicants attend the program each year. On the other hand, we have a large unlabeled dataset consists of applicants that are either rejected (i.e., not admitted) or declined (i.e., admitted but not enrolled). Additionally we need to address the subjectivity in the data due to change of membership of the admission committee year to year. However, unlike in the previous two applications, we do know the exact composition of data instances in each bias subgroup and, thus, the transfer learning [64] and Bayesian non-parametric clustering [77] techniques are not appropriate. To this end, we propose a new variant of semi-supervised SVM that trains a classifier within the "Learning Using Privileged Information" (LUPI) [83]

paradigm, which exploits pre-defined data subgroups to improve generalization.

Master's education is the fastest growing and largest component of the graduate enterprise in the United States. According to the 2016 joint survey conducted by the CGS (Council of Graduate Schools) and ETS (Educational Testing Service) [15], first-time enrollment in U.S. graduate programs reached a record high total of 506,927 students in Fall 2015. Because of the rise in applicants, the admissions process may become increasingly tedious and challenging. The ETS has established standardized tests (such as the GRE) to help evaluate applicants' quantitative, reading, and writing skills, but these scores alone are far from indicative of successful students. Although applicants' previous achievements can demonstrate excellence, students with high GPAs from prestigious universities do not always excel in their graduate studies.

In this work, we take a quantitative machine learning approach to predict the outlook of applicants' graduate studies based on features extracted from their application materials [99]. The training data for our model are admitted students with their empirical performance measures in the graduate program. In particular, we have a real world dataset from Northeastern University's MS in Computer Science program, consisting of MS students from 2009 to 2012. We use a student's overall GPA at Northeastern as his/her performance measure. Our model aims to identify the top 20% and bottom 20% performing students respectively (see details in Section 5.4.1).

Two challenges arise when learning with this data. First, the data involves the admission committee's (possibly subjective) evaluation. Specifically, some members of the committee may be biased in weighing a particular set of standards (e.g., GRE scores), while others may be in favor of different measures. This issue is particularly acute when the admission committee/policy changes from year to year. As a result, it can be difficult to form an accurate predictor directly from the entire dataset. Another challenge is the limitation of the training data. We have a total of 454 labeled training samples (all admitted students) from 2009 to 2012. On the other hand, we have over 2000 applications that are either rejected (i.e., not admit-

ted) or declined (i.e., admitted but not enrolled), which can serve as an unlabeled auxiliary dataset. Our conjecture is that building a semi-supervised model leveraging the large set of unlabeled data may lead to a superior performance compared to using the labeled data alone.

Our model is inspired by two existing frameworks: SVM+ [83] and S3VM [6]. SVM+ is a variant of SVM which addresses the issue of heterogeneous data. Specifically, SVM+ implicitly establishes a different hyper-plane for each data subgroup by modifying a standard SVM's objective function and constraints. S3VM is a semi-supervised version of SVM which learns a classifier using both labeled and unlabeled data. Our contribution is a new variant of SVM that unifies the advantages of both S3VM and SVM+. Our new model, which we name S3VM+, addresses the admission biases in the labeled data and utilizes unlabeled applicants' data simultaneously. S3VM+ can be applied to any domain for which the data may have clearly defined subgroups (e.g., privileged domain knowledge) and a large amount of unlabeled data.

An additional motivation of our research was to validate our hypothesis of whether we could predict student success based only on quantitative measures and, thus, remove the subjectivity of the committee reading the recommendation letters and statement of purpose. If successful, such a model will not only lead to a better selected student body, but also help to manage growing enrollments. Our experimental results (see Section 5.4 for details) demonstrate that, with our new model, we can achieve an effective yet imperfect prediction. Thus, in practice, our model could serve as a Focus of Attention (FOA) tool for the admission committee.

The rest of the chapter is organized as follows: in Section 5.2, we present the related work in predicting students' performance in the education domain. In Section 5.3, we give a brief introduction to SVM, S3VM and SVM+ and present our model S3VM+ in detail. We demonstrate the efficacy of our model in Section 5.4 by comparing its performance to those three existing models. Finally, we conclude in Section 5.5.

5.2 Related Work: Educational Data Mining

Our work is closely related to Educational Data Mining (EDM). The Educational Data Mining community website [25] defines educational data mining as follows: "Educational Data Mining is an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in." Most EDM studies focus on predicting students' academic performance after they have been admitted to the college/program. For example, Lepp et al. investigated the relationship between cell phone use and academic performance in a sample of US college students [49]. Delen applied machine learning techniques for student retention management [23]. Ioanna et al. presented a dropout prediction in e-learning courses using machine learning techniques [53]. Nevertheless, another important aspect of educational research is selecting the best fitting students at admission time, which has not been widely addressed in past literature.

The most closely related work to our project is the admissions research conducted by the University of Texas at Austin (UT Austin) for their graduate admission program [88], driven in part by their need to manage growing application numbers. In their work, the authors applied logistic regression (LR) to help the admission committee identify weak candidates who will likely be rejected and exceptionally strong candidates who will likely be admitted. Our work bears a similar mission but is different in three aspects. First, the UT Austin research includes credentials such as recommendation letters and statement of purpose, whereas our work builds a purely quantitative model relying only on non-subjective measures. Second, the recommendations made by UT Austin's algorithm are based on an applicant's likelihood of admission, whereas our model aims to predict the future performance of the applicants in the graduate program. Last, our model addresses human subjectivity in admission decisions. The contribution of our work is a quantitative machine learning model to predict a candidate's future performance.

5.3 Integrating Semi-supervised SVM with domain knowledge

We choose our model based on the characteristics of our dataset and particular challenges involved in our task. In particular, we choose SVM and two existing frameworks: S3VM [6] and SVM+ [83]), as our baseline models. Our proposed model is a new variant of SVM, which is inspired by S3VM and SVM+. We first give brief introductions to SVM, S3VM and SVM+. We then describe our new model in detail in Section 5.3.4.

5.3.1 Standard SVM

Support vector machine (SVM) was proposed by Cortes and Vapnik in 1993 [19]. In its basic form, SVM is a supervised learning algorithm which classifies the training data into two (positive and negative) classes. It seeks a linear hyper-plane that maximizes the margin between the positive and negative training instances. Formally, given a labeled dataset

$$L = \{(x_1, y_1), (x_2, y_2) \ldots, (x_n, y_n)\}$$

$$\text{where } x_i \in R^d \text{ and } y_i \in \{1, -1\} \quad \forall i = 1, \ldots, n.$$

The maximum margin classifier is found by solving the following quadratic optimization problem:

$$\min_{w,b,\eta} \quad \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{l} \eta_i \tag{5.1}$$

subject to

$$y_i(w \cdot x_i + b) + \eta_i \geq 1 \quad \eta_i \geq 0 \quad i = 1, \ldots, l$$

where C is the trade-off between maximizing the margin and total violations. A typical method to select C is to perform a grid search using the training data (see Section 5.4.2 for details). For any given new instance x^*, a SVM classifies x^* using $sign(w \cdot x^* + b)$.

5.3.2 S3VM (Semi-Supervised SVM)

S3VM is semi-supervised SVM proposed by Bennett and Demiriz [6]. The model is learned using a mixture of labeled data (the training set) and unlabeled data (the auxiliary set). The objective is to assign class labels to the auxiliary set such that the "best" support vector machine (SVM) is constructed. In particular, given a labeled dataset $L = \{x_1, x_2, \ldots, x_l\}$ and an unlabeled auxiliary dataset $U = \{x_{l+1}, x_{l+2}, \ldots, x_{l+k}\}$, S3VM learns a classifier from both L and U using overall risk minimization (ORM) posed by Vapnik [84] (Chapter 10). Starting with the standard SVM formulation, S3VM adds two constraints for each data point in the auxiliary set U. One constraint calculates the misclassification error as if the point were placed in class 1, and the other constraint calculates the misclassification error as if the point were placed in class -1. The objective function calculates the minimum of the two possible misclassification errors. The final membership assignments of the instances in U correspond to the ones that result in a minimum total sum of slacks across all instances in the training set. Specifically, we have:

$$\min_{w,b,\eta,\xi,z} \quad \frac{1}{2} \parallel w \parallel^2 + C \left[\sum_{i=1}^{l} \eta_i + \sum_{j=l+1}^{l+k} \min(\xi_j, z_j) \right] \tag{5.2}$$

subject to

$$\begin{aligned} y_i(w \cdot x_i + b) + \eta_i \geq 1 \quad & \eta_i \geq 0 \quad i = 1, \ldots, l \\ w \cdot x_j + b + \xi_j \geq 1 \quad & \xi_j \geq 0 \quad j = l+1, \ldots, l+k \\ -(w \cdot x_j + b) + z_j \geq 1 \quad & z_j \geq 0 \quad j = l+1, \ldots, l+k \end{aligned}$$

where C is the trade-off between maximizing the margin and total violations. η_i's are the slacks for the labeled data, and ξ_j's and z_j's are the slacks for the unlabeled data hypothetically assigned to the positive and negative classes respectively.

Equation (5.2) can be solved using mixed integer programming by applying the "large integer M" technique. The basic idea is to introduce a constant integer $M > 0$ and a decision variable $d_j \in \{0, 1\}$ for each point x_j in the auxiliary set U. d_j indicates the class membership of x_j. If $d_j = 1$, then the point is in class 1 and if

$d_j = 0$, then the point is in class -1. The integer M is chosen sufficiently large such that if $d_j = 0$ then $\xi_j = 0$ is feasible for any optimal w and b. Likewise if $d_j = 1$, then $z_j = 0$. In other words, ξ_j and z_j can have at most one non-zero value no matter what class x_i belongs to. Consequently, we could replace the $min(\xi_j, z_j)$ in Equation (5.2) by $(\xi_j + z_j)$. This results in the following optimization formulation:

$$\min_{w,b,\eta,\xi,z} \quad \frac{1}{2} \parallel w \parallel^2 + C \left[\sum_{i=1}^{l} \eta_i + \sum_{j=l+1}^{l+k} (\xi_j + z_j) \right] \tag{5.3}$$

subject to

$$\begin{aligned}
& y_i(w \cdot x_i + b) + \eta_i \geq 1 \\
& \eta_i \geq 0, \quad i = 1, \ldots, l \\
& w \cdot x_j + b + \xi_j + M(1 - d_j) \geq 1 \\
& -(w \cdot x_j + b) + z_j + M d_j \geq 1 \\
& \xi_j \geq 0, \quad z_j \geq 0, \\
& j = l+1, \ldots, l+k, \quad d_j \in \{0, 1\}
\end{aligned}$$

The solution to Equation (5.3) can be found using mixed integer programming products. In our experiment, we used CVX [1] and Gurobi [2] optimizers. Same as a standard SVM, S3VM classifies a new instance x^* using $sign(w \cdot x^* + b)$.

5.3.3 SVM+

Vapnik and Vashist [83] introduced SVM+, which is a variant of SVM that addresses the issue of learning with heterogeneous data. In their model, the authors developed a new paradigm to learn using privileged information (LUPI). The objective of SVM+ is to take advantage of additional domain knowledge, and in particular data subgroups that may arise from different sources or due to labeling biases.

Suppose the training data are the union of $t > 1$ groups. We follow the notation in [51] and denote the indices of group r by

$$T_r = \{i_{n_1}, \ldots, i_{n_r}\}, \quad r = 1, \ldots, t$$

All training samples can then be represented as:

$$\{\{X_r, Y_r\},\ \ r = 1, \ldots, t\}$$

where $\{X_r, Y_r\} = \{(x_{r_1}, y_{r_1}), \ldots, (x_{r_{n_r}}, y_{r_{n_r}})\}$. To incorporate the group information while training the model, SVM+ defines the slacks inside each group by a unique *correcting function*:

$$\xi_i = \xi_r(x_i) = \phi_r(x_i, w_r), \qquad i \in T_r,\ \ r = 1, \ldots, t$$

Specifically, the correcting functions are defined as:

$$\xi_r(x_i) = w_r \cdot x_i + d_r, \qquad i \in T_r,\ \ r = 1, \ldots, t$$

Compared to a standard SVM, SVM+ uses slack variables that are restricted by the correcting functions, and the correcting functions capture additional information about the data. Note that all of the data is used to construct the decision hyperplane. The group information is only used to fine tune the slack variables. Formally, the objective function for $SVM+$ is formulated as follows:

$$\min_{w,b,w_1,w_2,w_r,d_1,d_2,d_r} \quad \frac{1}{2} \parallel w \parallel^2 \ + \ \frac{\gamma}{2} \sum_{r=1}^{t} \parallel w_r \parallel^2 \ + \ C \sum_{r=1}^{t} \sum_{i \in T_r} \xi_i^r \tag{5.4}$$

subject to:

$$y_i(w \cdot x_i + b) + \xi_i^r \geq 1$$

$$\xi_i^r(x_i) = w_r \cdot x_i + d_r$$

$$\xi_i^r \geq 0, \quad i \in T_r, \quad r = 1, \ldots, t$$

Parameter γ adjusts the relative weight between $\parallel w \parallel^2$ and the $\parallel w_r \parallel^2$'s. C is the trade-off between maximizing the margin and total violations.

Liang and Cherkassky [51] further extended the SVM+ approach to multi-task learning. In the SVM+MTL [51] framework, the data is partitioned into groups using privileged information similar to the SVM+ model. However, instead of mak-

ing a correcting function for the slack variables, their model establishes a unique correcting function (i.e., a hyper-plane) for each group in addition to a shared common hyper-plane. In other words, the decision function for group $r = 1, \ldots, t$ is as follows:

$$f_r(x) = (w \cdot x + b) + (w_r \cdot x + d_r)$$

where w, b are the parameters for the common hyper-plane and w_r, d_r are the parameters for the correcting function for group r. The corresponding formulation of the quadratic optimization problem is as follows:

$$\min_{w,b,w_1,w_2,w_r,d_1,d_2,d_r} \quad \frac{1}{2} \parallel w \parallel^2 \; + \; \frac{\gamma}{2} \sum_{r=1}^{t} \parallel w_r \parallel^2 \; + \; C \sum_{r=1}^{t} \sum_{i \in T_r} \xi_i^r \tag{5.5}$$

subject to

$$y_i[(w \cdot x_i + b) + (w_r \cdot x_i + d_r)] + \xi_i^r \geq 1$$

$$\xi_i^r \geq 0 \quad i \in T, \quad r = 1, \ldots, t$$

SVM+MTL is an adaptation of SVM+ for solving MTL problems. In our experiment, we applied the SVM+MTL framework because it provides more flexibility to learn a different decision plane for each year's student data.

For SVM+MTL, predicting the class label for a new given instance x^* is not straightforward because its decision function requires a group-dependent correcting function, and we do not know the group membership of test instances. To resolve this problem, we predict the label for x^* in each group and perform a majority vote over all predicted labels. Specifically, a test instance x^* will be predicted in each group as follows:

$$f_r(x^*) = sign[(w \cdot x^* + b) + (w_r \cdot x^* + d_r)]$$

where $r = 1, \ldots, t$ are the bias groups, and w, b, w_r's and d_r's are learned model parameters. The class membership for x^* is determined by a majority vote over $f_r(x^*)$'s.

5.3.4 S3VM+

Our new model, S3VM+ leverages the unlabeled data and addresses the biases in the training data simultaneously. In particular, we train our model with a labeled dataset and an unlabeled auxiliary dataset. Furthermore, our data is partitioned into yearly groups because of the admissions committee changes from year to year and thus may have different biases. For the labeled dataset, we incorporate the grouping information by establishing a correcting function for each group (constraints (a) and (b) in Equation (5.7)).

For the unlabeled data, we introduce two slack variables ξ_i and z_i for each data point x_i representing the error of placing x_i in the positive class and negative classes respectively. The objective function for S3VM+ takes the minimum of the two slacks for each unlabeled instance and minimizes the total sum of error across all training instances.

Formally, the optimization problem for S3VM+ is formulated as follows:

$$\min_{w,b,w_1,w_2,w_r,d_1,d_2,d_r} \frac{1}{2} \parallel w \parallel^2 + \frac{\gamma}{2} \sum_{r=1}^{t} \parallel w_r \parallel^2 + C \left[\sum_{i=1}^{l} \eta_i^r + \sum_{j=l+1}^{l+k} \min(\xi_j^r, z_j^r) \right] \tag{5.6}$$

subject to

$$\begin{aligned}
&(a)\ \ y_i[(w \cdot x_i + b) + (w_r \cdot x_i + d_r)] + \eta_i^r \geq 1 && (labeled)\\
&(b)\ \ \eta_i^r \geq 0 \quad i = 1, \ldots, l\\
&(c)\ \ [w \cdot x_j + b + (w_r \cdot x_j + d_r)] + \xi_j^r \geq 1 && (unlabeled)\\
&(d)\ \ \xi_j^r \geq 0 \quad j = l+1, \ldots, l+k, \quad d_j \in \{0, 1\}\\
&(e)\ \ -[(w \cdot x_j + b) + (w_r \cdot x_j + d_r)] + z_j^r \geq 1 && (unlabeled)\\
&(f)\ \ z_j \geq 0 \quad j = l+1, \ldots, l+k \quad d_j \in \{0, 1\}
\end{aligned}$$

where C is the trade-off between maximizing the margin and total violations, and γ is the trade-off parameter between $\parallel w \parallel^2$ and the $\parallel w_r \parallel^2$'s.

We apply the "large integer M" technique (see Section 5.3.2 for details)

and convert the constraint with a minimization function to two constraints over linear functions. Because both labeled and unlabeled data are grouped by academic year, we apply the same correcting functions used for the labeled data to each corresponding annual group of unlabeled data (constraints (c) to (f) in Equation (5.7))

$$\min_{w,b,w_1,w_2,w_r,d_1,d_2,d_r} \frac{1}{2} \parallel w \parallel^2 + \frac{\gamma}{2} \sum_{r=1}^{t} \parallel w_r \parallel^2 + C \left[\sum_{i=1}^{l} \eta_i^r + \sum_{j=l+1}^{l+k} (\xi_j^r + z_j^r) \right] \tag{5.7}$$

subject to

$$\begin{aligned}
&(a)\ \ y_i[(w \cdot x_i + b) + (w_r \cdot x_i + d_r)] + \eta_i^r \geq 1 && (labeled)\\
&(b)\ \ \eta_i^r \geq 0 \quad i = 1, \ldots, l\\
&(c)\ \ [w \cdot x_j + b + (w_r \cdot x_j + d_r)] + \xi_j^r + M(1 - d_j) \geq 1 && (unlabeled)\\
&(d)\ \ \xi_j^r \geq 0 \quad j = l+1, \ldots, l+k, \quad d_j \in \{0, 1\}\\
&(e)\ \ -[(w \cdot x_j + b) + (w_r \cdot x_j + d_r)] + z_j^r + M d_j \geq 1 && (unlabeled)\\
&(f)\ \ z_j \geq 0 \quad j = l+1, \ldots, l+k \quad d_j \in \{0, 1\}
\end{aligned}$$

where C is the trade-off between maximizing the margin and total violations, and γ is the trade-off parameter between $\parallel w \parallel^2$ and the $\parallel w_r \parallel^2$'s.

To classify a new instance x^*, we follow the same approach as SVM+, which is to take a majority vote on class labels predicted by each group.

5.4 Experimental Results

In this section, we first describe the process of constructing our training and testing dataset. We then discuss the methods we used to conduct our experiments in Section 5.4.2. We present our analysis of our experiments in Section 5.4.3.

5.4.1 Constructing the Training and Test Data

We have a real world dataset consisting of MS students from the Master's in CS program at Northeastern University. Table 5.1 presents the features we collected from students' applications for our experiment. Feature 1 contains the students' undergraduate GPAs adjusted according to each individual university's grading scale. For example, a 3.5 out of 5 and a 7 out of 10 would result in the same value. Feature 10 contains self-reported values representing the maximum number of lines of programming written by the student prior to joining the MS program. Feature 12 contains the rankings of the undergraduate institutions where the students obtained their bachelor's degrees. We classified the rankings into four categories with one being the most prestigious and four being the least. The classification was performed manually according to the Best Global Universities list published by US News and World Report. The rest of the features are standardized test scores. Both the GRE and TOEFL had two versions of tests during 2009 - 2012 which use different scoring scales. Both of these tests are converted to their new versions of scoring scales using conversion tables provided by the ETS [15].

As mentioned in Section 5.1, our task is to identify successful candidates at the point of admission. One measure of success is MS-GPA in the MS program (as distinct from the input feature 1 "Undergraduate GPA"). Indeed, a cumulative MS-GPA is the most widely used measure for students' academic performance [75]. The labels in our training data are determined by the training instances' percentiles in the overall MS-GPAs. Specifically, the top and bottom 20% students are labeled with class 1 and -1 respectively. The number 20% was intuitively chosen as an measure which sets the individuals apart from the average students.

Note that we did not use a midpoint MS-GPA as a cutoff to separate the positive and negative classes, in order to reduce the label noise. In particular, instances close to the average GPA are harder to categorize as good or bad students. Another intuitive approach is to define two hard MS-GPA thresholds for good versus bad performances, i.e., to have a MS-GPA above an upper threshold (e.g., > 3.8)

for good students, and below a bottom threshold (e.g., < 3.0) for bad students. A further investigation reveals that this approach is less effective for the following reason: different instructors have different grading policies due to the nature of the courses. For some fundamental courses, an 'A' means you are in the top 30% of a class, while for some other advanced courses, an 'A' means you are in the top 10% of a class. Even for the same course in the same year with different sections, instructors may choose to cooperate on exams/grading or not. Because students have different instructors and/or take different courses, hard cutoffs are not an accurate reflection of a student's abilities.

Having stated this, on the other hand, if a student performs consistently in the top 20% in each class, this student will be among the top 20th percentile of the entire MS-GPA spectrum. The same can be said for those that perform consistently in the bottom 20th percentile. Identifying the factors that lead to this consistent success or underperformance are of greatest interest to this research. Therefore, we used relative measures to label our positive and negative training samples. For comparison purposes, we report our experimental results on both relative and hard cutoffs in Tables 5.5 and 5.6 respectively.

Table 5.2 summarizes the distribution of students from 2009 to 2012 using relative thresholds. Column "Total" is the total number of students enrolled in the corresponding year. Columns "Top 20% MS-GPA" and "Bottom 20% MS-GPA" are the total number of students in the top and bottom 20th percentile among their peers measured by the cumulative MS-GPAs. There is not an equal number of positive and negative instances for each year because there are multiple students with the same MS-GPA.

Both SVM+ and our model S3VM+ make use of an unlabeled auxiliary dataset. We collect the application data of rejected (i.e., not admitted) and declined (i.e., admitted but not enrolled) applicants as the auxiliary data. These data contain the same features as the labeled data, and the size distribution of auxiliary data from 2009 to 2012 is presented in the last column of Table 5.2. Our training data are all labeled and unlabeled instances from 2009 to 2011, and our test data are labeled

Table 5.1: Features Collected for Training

1	Undergraduate GPA
2	GRE Verbal
3	GRE Quantitative
4	GRE Analytical Writing
5	TOEFL Total
6	TOEFL Reading
7	TOEFL Listening
8	TOEFL Speaking
9	TOEFL Writing
10	Max # of Lines of Code Written
11	Bachelor's Degree in EECS (Yes/No)
12	Undergraduate School Ranking

Table 5.2: Student Data Statistics

Year	Total	Top 20%	Bottom 20%	Auxiliary Data
2009	37	7	7	431
2010	89	18	17	503
2011	132	28	27	705
2012	196	51	42	948

Table 5.3: Prediction Using 1Y Data

Train	Test	Top 20% MS-GPA	Bottom 20% MS-GPA	Overall
2009	2010	0.72	0.59	0.66
2010	2011	0.64	0.70	0.67
2011	2012	0.65	0.76	0.70

instances from 2012.

5.4.2 Experimental Method

We are interested in identifying the top and bottom 20% of candidates from an application pool based on the performance of the admitted students. Our first

Table 5.4: Predicting Using 10-fold Cross Validation

	Test Accuracy			Traing Accuracy		
Dataset	Top 20% MS-GPA	Bottom 20% MS-GPA	Overall	Top 20% MS-GPA	Bottom 20% MS-GPA	Overall
2009 - 2011	0.70	0.71	0.71	0.79	0.79	0.79
2009 - 2012	0.74	0.72	0.73	0.84	0.75	0.79

Table 5.5: Performance Comparison of Four Models Using Relative Cutoffs

	Test Accuracy			Traing Accuracy		
Model	Top 20% MS-GPA	Bottom 20% MS-GPA	Overall	Top 20% MS-GPA	Bottom 20% MS-GPA	Overall
SVM	0.73	0.71	0.72	0.79	0.80	0.79
S3VM	0.75	0.74	0.74	0.81	0.82	0.81
SVM+	0.77	0.70	0.74	0.92	0.84	0.88
S3VM+	**0.82**	**0.72**	**0.77**	**0.95**	**0.89**	**0.92**

Table 5.6: Performance Comparison of Four Models Using Hard Cutoffs

	Test Accuracy			Traing Accuracy		
Model	> 3.8 MS-GPA	< 3.4 MS-GPA	Overall	> 3.8 MS-GPA	< 3.4 MS-GPA	Overall
SVM	0.65	0.69	0.66	0.73	0.75	0.74
S3VM	0.72	0.65	0.70	0.83	0.70	0.77
SVM+	0.75	0.64	0.71	0.92	0.75	0.84
S3VM+	**0.77**	**0.67**	**0.74**	**0.93**	**0.80**	**0.87**

goal is to confirm our conjecture that there are biases in admission decisions from year to year. To this end, we conducted two experiments. The first experiment is to use the previous year's data to predict the current year's performance using a standard SVM. For example, we would use class 2009's data to predict class 2010's performance, and class 2010's data to predict class 2011's performance. Table 5.3 presents the prediction accuracies for each year. We observe that, for 2010, the top 20% of students are easier to predict than the bottom 20%, whereas for 2011 to 2012, the situation is reversed. This lack of consistency and the low overall accuracies (up to 70%) suggest that there is no strong correlation of predicative patterns from year to year. Our second experiment is to apply a standard 10-fold cross validation on two datasets: all data from 2009 to 2011 and all data from 2009 to 2012. Because 2012 added a significant amount (89%) of instances, we would expect a noticeable increase in both the training and test accuracies if the data across different years conform to the same distribution. Table 5.4 summarizes the results of this experiment. We observe only a marginal improvement in overall test accuracy after adding instances from 2012 and, more importantly, the overall fit of the data remains the same (79%). From these two experiments, we conclude that data across different academic years have different distributions. We believe this year to year bias is due to the change in the membership of the admission committee.

In light of above learned information, we partitioned the data by academic year and use them as the privileged groups in SVM+ and S3VM+. We take the union of labeled data from 2009 to 2011 as our labeled training data. The auxiliary dataset is formed as the union of the corresponding auxiliary data from 2009 to 2011. We test and compare the performance of the four models (SVM, S3VM, SVM+, S3VM+) in predicting labeled instances in 2012.

The hyper-parameters are the trade-off constant C for all four models and γ for SVM+ and S3VM+. We perform 10-fold cross validation and grid search on the training data to select the hyper-parameters. We first use a coarse grid $\{0.01, 10, 1000\}$ for C and refine the candidates after the initial search. The final list for C is $\{1, 10, 100\}$. Following a similar procedure, our final search list

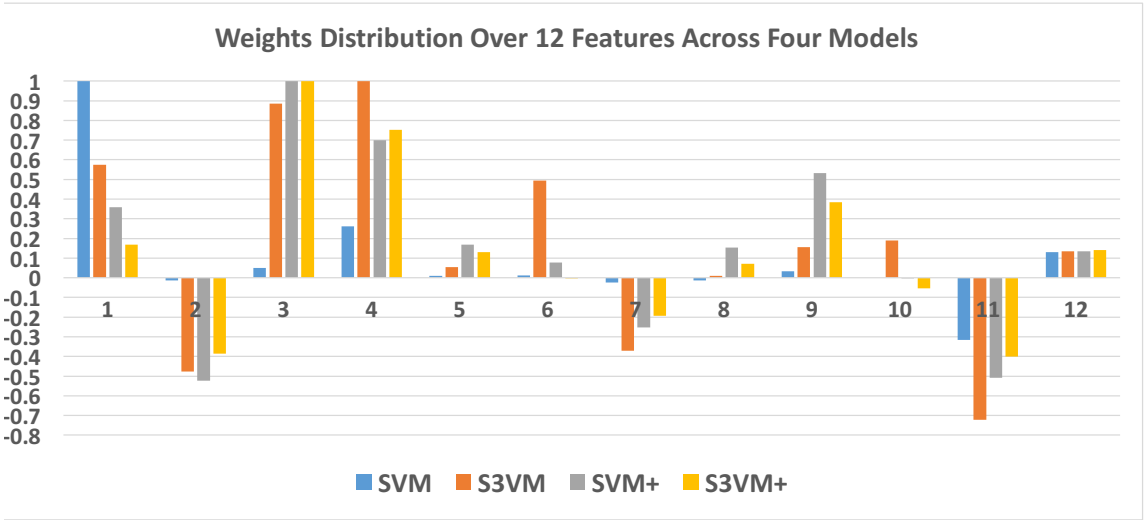


Figure 5.1: Plot of individual feature weights $w1$ to $w12$ across four models. The weights are scaled with the maximum absolute value of w_i's in each model, i.e., $w = \frac{w}{\max\limits_{1 \leq i \leq 12} \{|w_i|\}}$)

for γ is $\{0.01, 1, 100\}$. After the best hyper-parameters are selected, we train the corresponding model one more time using the entire training data and then apply the learned model to the test data and measure its performance. We report both training and test accuracies in Table 5.5.

5.4.3 Analysis of Performance

Table 5.5 displays the main results of our experiment. First, we observe that the test accuracies for SVM on the positive and negative classes are more balanced compared to the results in Table 5.3. There is also an improvement in the overall performance for SVM. This can be explained by the increased amount of training data (three years versus only one year of data) used in our Table 5.5 experiment.

Second, we conclude that all three variants of SVM (S3VM, SVM+, S3VM+) are superior to standard SVM. Using SVM as a baseline measure:

- S3VM improved slightly on the accuracies of both positive and negative classes, which suggests that using auxiliary data has a positive impact on identifying

Table 5.7: Weights Ranking Comparison of Four Models

Model	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
SVM	w1	w11	w4	w12	w3	w9
S3VM	w4	w3	w11	w1	w6	w2
SVM+	w3	w4	w9	w2	w11	w1
S3VM+	w3	w4	w11	w2	w9	w7
	Rank 7	Rank 8	Rank 9	Rank 10	Rank 11	Rank 12
SVM	w7	w2	w6	w8	w5	w10
S3VM	w7	w10	w9	w12	w5	w8
SVM+	w7	w5	w8	w12	w6	w10
S3VM+	w1	w12	w5	w8	w10	w6

both the good and bad students. This is consistent with the fact that the auxiliary data contain both declined (i.e., admitted but not enrolled) and rejected (i.e., not admitted) applicants, which could improve the accuracy of positive and negative classes respectively.

- SVM+ demonstrated improvement on the positive side only, which indicates that the partition of bias groups by academic year is most effective in identifying the top students. One explanation for this could be that the top 20% of students are inherently different from year to year, while the bottom 20% of students remain similar, or that a particular admissions committee has biases about how to recognize a strong student.

- Our model S3VM+ has a noticeable advantage among all models in predicting the positive class: 82% versus 73% (SVM), 75% (S3VM) and 77% (SVM+). In light of the construction of S3VM+, one could conclude that adding auxiliary data to each partition group further enhances the power of identifying top students. On the other hand, because grouping does not have a significant impact on identifying bottom students (as demonstrated by SVM+), S3VM+ would only result in a limited gain for the negative class.

Lastly, from the training accuracies presented in Table 5.5, we observe a

significantly better fit of the training data using our model S3VM+. In particular, 95% versus 92% (SVM+), 81% (S3VM), 79% (SVM) accuracies for the positive class and 89% versus 84% (SVM+), 82% (S3VM) and 80% (SVM) accuracies for the negative class. Compared to the standard SVM, S3VM improved training accuracies evenly on both classes, and SVM+ and S3VM+ demonstrated more significant gains on the positive class, which is consistent with what we observed in the test data.

5.4.4 Labeling Strategy: Relative versus Absolute

Recall that in Section 5.4.1, we discussed our choice of labeling the top 20% and bottom 20% of students with respect to their MS-GPAs as our two classes. We explained our rationale of using relative rather than hard cutoffs to label our data. We confirm this conjecture in Table 5.6, where we show the results of an experiment using MS-GPA > 3.8 for the top students and MS-GPA < 3.4 for the other. In the table we see that for all four methods, the overall accuracies are lower than in Table 5.5.

5.4.5 Analysis of Weight Vectors

Because we utilized a linear SVM and its variants, we found it interesting to investigate the ranking and magnitude of each individual feature in the weight vectors produced by each model. The ranking indicates the importance of a feature, and the magnitude helps to identify weights that are insignificant (e.g., close to zero).

Table 5.7 presents the ranking of w_i's in the weight vectors (w's) of four models. Figure 5.1 displays the weights of individual features across four models using their magnitudes. In order to make a meaningful comparison, each weight vector $w = \{w_1, w_2, \ldots, w_{12}\}$ is scaled by the maximum absolute value of its components, i.e.,

$$w = \frac{w}{\max\limits_{1 \leq i \leq 12} \{|w_i|\}}$$

Thus, the weight for the most important feature is either 1 or -1. Note that, for SVM+ and S3VM+, we display the shared hyper-plane vector w without the cor-

recting functions for each group.

From Figure 5.1, we conclude that the most important features are 1 ("Undergraduate GPA"), 2 ("GRE Verbal"), 3 ("GRE Quantitative"), 4 ("GRE Analytical Writing") and 11 ("Bachelor's Degree in EECS (Yes/No)"). A closer examination reveals that SVM relies mostly on three features (1, 4, and 11). S3VM has significantly large weights on two additional features, 6 ("TOEFL Reading") and 7 ("TOEFL Listening"), on top of the five features listed above. SVM+ and S3VM+ made use of one additional feature which is 9 ("TOEFL Writing").

From Table 5.7, we observe that all models except standard SVM suggest the same top two features: "GRE Quantitative" and "GRE Analytic Writing" scores. Furthermore, SVM+ and S3VM+ overlapped in their top five features but with a different ranking order.

5.4.6 Practical Value of our Method for Admission

Although the performance of our approach is not perfect, our experimental results demonstrate an effective predictive model that could serve as a Focus of Attention (FOA) tool for an admission committee. Future work will be to determine how to look at students' transcripts to understand undergraduate GPA more deeply (we conjecture that high scores for computer science and math classes are more predictive than overall GPA). In addition, the letters and personal statement likely have some signal as well as to which students will be most successful and thus we will incorporate text mining methods.

5.5 Conclusion

We applied a quantitative machine learning approach to predict candidates' potential academic performance based on information from their applications. We built our model using admitted students with their cumulative MS-GPA as the performance measure (i.e., label) and tested our model's efficacy for the incoming students. Throughout our experiments, we found a unique challenge associated with our task,

which is different data distributions across the academic years due to subjectivity arising from changing membership of the admissions committee. We addressed this issue with the "Learning Using Privileged Information" (LUPI) framework [83]. We further handled the limited training data issue by employing a semi-supervised version of SVM to utilize the large amount of unlabeled data (i.e., the rejected/declined applications). Our resulting model, S3VM+, is a novel variant of SVM that addresses subjectivity and lack of labeled data simultaneously. Our experimental results demonstrate a significant gain in performance of our model as compared to three existing models (i.e., SVM, S3VM, and SVM+). Although we based our work on a two-year master's program, our model is easily extensible to similar tasks such as college or pre-school admissions. Our model can also be applied to other real world situations in which data may have clearly defined subgroups (i.e., privileged information) and a large amount of unlabeled data.

Chapter 6

Conclusion and Future Work

This thesis focuses on investigating machine learning methods that can be used to enhance performance while learning with datasets containing bias and human subjectivity. The undertaking of this research is valuable because it addresses violations to one of the fundamental principle in machine learning which assumes data are drawn from the same distribution (i.e., data points are i.i.d). In particular, the existence of bias and subjectivity in data leads to multiple subgroups each of which has its own unique characteristics. Consequently, learning algorithms which do not take into account these distinctions are insufficient and can result in unsatisfactory performance. In this thesis, we illustrated our findings in three motivating domains each of which is associated with its unique challenges. We conclude from our experiments that addressing data bias and subjectivity in the learning process can achieve superior performance over traditional machine learning approaches.

In our multiple sclerosis (MS) research, we collaborated with doctors from Harvard medical school to estimate the long term prognosis of multiple sclerosis patients utilizing their short term clinical observations. The dataset consists of patients collected from nineteen doctors in the CLIMB study [30] at Brigham and Women's Hospital. Patients' preferences for their own doctors and doctors' individualized interpretations on patients' clinical test results lead to heterogeneous data distributions for different practitioners. Our transfer learning [64] algorithm built a model for each doctor using patients under his/her practice (the primary dataset)

plus a carefully selected subset of patients from other practitioners (the auxiliary dataset). In particular, we provided a classification model to predict disease course of MS patients with a preferred high performance in the "progressive" cases. We demonstrated the efficacy of this model in Section 2.6 by comparing it to other existing approaches such as SVM [20], TrAdaboost [21], and KNN [90].

Considering the constraint of a minimum size requirement a transfer learning algorithm imposes on the primary dataset and the exclusion of some training samples from the auxiliary dataset (Section 2.7), we employed Bayesian non-parametric mixture model to address bias subgroups in our regression task of forecasting the actual MS disability scores for MS patients. Specifically, each component in the mixture model represents data instances of similar bias characteristics where the number of components dependents on the data, i.e., inferred from the data as part of the training process. Furthermore, we resorted to Gaussian Process [68] to model the complex relationship between the predictors and target values and a k-means algorithm to model the domain knowledge from the experts. Our final model is a domain-induced Dirichlet mixture of Gaussian processes model (DI-DPMGP, Chapter 3) which simultaneously addresses various types of bias and the domain specific information in the medical data. We applied our regression model to the MS dataset and an early Parkinson's patients dataset from the UCI machine learning database [52] where, in both cases, our model showed significant improvements (Section 3.5) over other existing approaches.

In our epilepsy research, we collaborated with doctor's from New York University Comprehensive Epilepsy Center to detect cortical lesions using epileptic patients' brain MRI images. The bias in this dataset originates from the inter-patient variability, i.e., each human brain has its own characteristics depending on the severity of the disease, demographics, etc. Noting that the transfer learning framework is not applicable in this case due to an absence of a meaningful partition of primary and auxiliary datasets, we adopted (similar to our DI-DPMGP approach) a Bayesian non-parametric clustering technique to induce bias subgroups from the data. We further applied restricted Boltzmann machines (RBMs) [76, 94] to address the issue

of feature scarcity from a brain MRI image. As a result, our model is an infinite mixture of RBMs (Chapter 4) and it achieved a 56% lesion detection rate among eighteen "MRI-negative" patients at NYU Comprehensive Epilepsy Center (Section 4.5). The results are significant because a board of experienced neuroradiologists failed to locate any lesion for all eighteen patients.

We extended our research to the educational domain in which we collaborated with the admission committee of College of Computer and Information Science at Northeastern University to create a classifier to help select applicants for their masters in CS program. Our model demonstrated a quantitative approach to predict an applicant's potential success in the program, where success is measured by their predicted GPA in the MS program. (Chapter 5). In this case, the subjectivity stems from the change of committee memberships over time which leads to different distributions of students across multiple academic years. What distinguishes this application from previous two is that we know the exact composition of data in each bias subgroup, which makes our previous two approaches (i.e., transfer learning and Dirichlet process-based clustering) inappropriate. In addition, this task has a shortage in labeled data (enrolled students) and abundance in unlabeled data (rejected or declined students). We introduced a new variant of SVM [20] which uses both labeled and unlabeled data in a semi-supervised learning approach and addresses the subjectivity within the "Learning Using Privileged Information" (LUPI) paradigm [83]. Our model achieved a 77% overall accuracy in predicting the top 20% and bottom 20% of students in the program (Section 5.4). Although the performance of our approach is not perfect, it can serve as an effective Focus of Attention (FOA) tool for the admission committee to help reduce their increasingly heavy workload due to the rise in the number of applications.

Technically, our research ranges from transfer learning [64], learning using privileged information (LUPI) [83], to Bayesian non-parametric frameworks [11]. Various existing algorithms/concepts, such as the SVM [20], Gaussian Processes [68], Dirichlet Processes [77], restricted Boltzmann machines (RBM [37]), k-means [54] etc., are employed to serve as building blocks for our new models. Our approaches

are extensible to a variety of other real world applications. For example, the transfer learning (Chapter 2) and domain induced Dirichlet mixture of Gaussian Processes model (Chapter 3) are applicable to any clinical dataset compiled from multiple physicians in which some part of the data collected involves human judgement. Our DPM-RBM model (Chapter 4) can be used for biased dataset with a limited number of predictors and our S3VM+ (Chapter 5) model can be applied to a semi-supervised learning task with pre-defined data subgroups.

We foresee other research areas that could bring fruitful results. The first one would be addressing assumptions made by our models. In particular, our transfer learning approach assumes each patient belongs to just one doctor. This assumption is often violated when patients see residents rather than a primary doctor. Because longitudinal data is needed to predict disease progression in MS, in our experiments we included only those patients who saw the same doctor for the majority of their visits. Thus how to predict outcome in patients who see different doctors at different visits remains an open problem.

The second research area would be to explore other approaches in comparison to ours within the same application domain. For example, our DI-DPMGP model (Chapter 3) introduced hierarchical constraints to a non-parametric model in the form of data subgroups. Another approach could be to augment the generative DPMGP model and learn the two levels of clusters in one probabilistic framework. The inter-cluster constraints can be achieved using techniques such as do-not-link pairs as described in [70] and [65].

Lastly, we propose to develop models to better understand and detect biased data. A common characteristic of our applications is that, although we can not pinpoint the distribution of bias and subjectivity in the data, we are aware of the their existence and causation. Thus, we can take advantage of this knowledge to design approaches that better fit the data. In other situations, however, we may not know the cause, or even the existence, of biases in the data. Developing effective tools to discover and understand bias and subjectivity in data collected for a learning task would be a valuable research topic in machine learning.

Bibliography

[1] *CVX Research Inc.* www.cvxr.com.

[2] *Gurobi Optimizer.* www.gurobi.com.

[3] B. Ahmed, C. E. Brodley, K. E. Blackmon, R. Kuzniecky, G. Barash, C. Carlson, B. T. Quinn, W. Doyle, J. French, O. Devinsky, and T. Thesen. Cortical feature analysis and machine learning improves detection of mri-negative focal cortical dysplasia. *Epilepsy & Behavior*, 48:21 – 28, 2015.

[4] B. Ahmed, T. Thesen, K. Blackmon, Y. Zhao, O. Devinsky, R. Kuzniecky, and C. Brodley. Hierarchical conditional random fields for outlier detection: An application to detecting epileptogenic cortical malformations. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1080–1088, 2014.

[5] M. Bell, S. Rao, E. So, et al. Epilepsy surgery outcomes in temporal lobe epilepsy with a normal MRI. *Epilepsia*, 50(9):2053–2060, 2009.

[6] K. Bennett and A. Demiriz. Semi-supervised support vector machines. *NIPS*, 11:368–374, 1998.

[7] A. Bernasconi, N. Bernasconi, B. Bernhardt, and D. Schrader. Advances in mri for 'cryptogenic' epilepsies. *Nat Rev Neurol.*, 7(2):99–108, 2011.

[8] P. Besson, N. Bernasconi, O. Colliot, et al. Surface-based texture and morphological analysis detects subtle cortical dysplasia. In *MICCAI*, pages 645–652, 2008.

[9] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., 2006.

[10] D. M. Blei and M. I. Jordan. Variational methods for the dirichlet process. *Proceedings of the twenty-first international conference on Machine learning*, page 12, 2004.

[11] D. M. Blei and M. I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1.1:121–143, 2006.

[12] I. Blumcke, M. Thom, E. Aronica, et al. The clinicopathologic spectrum of focal cortical dysplasias: A consensus classification proposed by an ad hoc task force of the ILAE diagnostic methods commission. *Epilepsia*, 52(1):158–174, 2011.

[13] J. P. Bradford and C. E. Brodley. The effect of instance-space partition on significance. *Machine Learning 42.3*, pages 269–286, 2001.

[14] M. A. Carreira-Perpinan and G. Hinton. On contrastive divergence learning. *AISTATS*, 10:33–40, 2005.

[15] CGS and ETS. Graduate enrollmentand degrees: 2005 to 2015. *http://cgsnet.org/reports*, 2016.

[16] S. P. Chatzis and Y. Demiris. Nonparametric mixtures of gaussian processes with power-law behavior, ieee transactions on 23.12. *Neural Networks and Learning Systems*, pages 1862–1871, 2012.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[18] A. Compston and A. Coles. Multiple sclerosis. *Lancet 372 (9648): 150217. doi:10.1016/S0140-6736(08)61620-7. PMID 18970977*, 2008.

[19] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20.3:273–297, 1995.

[20] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines*. Cambridge University Press, 2000.

[21] W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for transfer learning. *Proc. 24th International Conference on Machine Learning.*, pages 193–200, 2007.

[22] A. Dale, B. Fischl, and M. Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *NeuroImage*, 9(2):179–194, 1999.

[23] D. Delen. A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, 49.4:498–506, 2010.

[24] H. B. Demuth, M. H. Beale, O. D. Jess, and M. T. Hagan. *Neural Network Design*. Martin Hagan, 2014.

[25] EDM. International educational data mining society. *http://www.educationaldatamining.org/*, 2016.

[26] M. Fang, E. McCarthy, and D. Singer. Are patients more likely to see physicians of the same sex? recent national trends in primary care medicine. *Am J Med. Volume 117, Issue 8*, pages 575–581, 2004.

[27] B. Fischl and A. Dale. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *PNAS*, 97(20):11050–11055, 2000.

[28] B. Fischl, M. Sereno, and A. Dale. Cortical surface-based analysis: II: Inflation, flattening, and a surface-based coordinate system. *NeuroImage*, 9(2):195–207, 1999.

[29] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[30] S. Gauthier, B. Glanz, M. Mandel, and H. W. HL. A model for the comprehensive investigation of a chronic autoimmune disease: The multiple sclerosis climb study. *Autoimmun Rev.*, 5(8):532–536, 2006.

[31] T. Gholipour, B. Healy, N. Baruch, H. Weiner, and T. Chitnis. Demographic and clinical characteristics of malignant multiple sclerosis. *Neurology*, 76(23):1996–2001, 2011.

[32] C. D. Good, J. Ashburner, and R. Frackowiak. Computational neuroanatomy: new perspectives for neuroradiology. *Revue Neurologique*, 157:685–700, 2001.

[33] W. A. Hauser and D. C. Hesdorffer. *Epilepsy: frequency, causes and consequences*. Epilepsy Foundation of America, 1990.

[34] B. Healy, I. Degano, A. Schreck, D. Rintell, H. Weiner, T. Chitnis, and B. Glanz. The impact of a recent relapse on patient-reported outcomes in subjects with multiple sclerosis. *Quality of Life Research*, 21(10):1677–1684, 2012.

[35] B. Healy, D. Engler, H. W. T. Gholipour, R. Bakshi, and T. Chitnis. Accounting for disease modifying therapy in models of clinical progression in multiple sclerosis. *Journal of the Neurological Sciences 303 (1-2): 109-113. doi:10.1016/j.jns.2010.12.024. PMID 21251671*, 2011.

[36] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14.8:1771–1800, 2002.

[37] G. E. Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007.

[38] J. Hobart, J. Freeman, and A. Thompson. Kurtzke scales revisited: the application of psychometric methods to clinical intuition. *Brain 123*, no. 5:1027–1040, 2000.

[39] S. J. Hong, H. Kim, D. Schrader, N. Bernasconi, B. C. Bernhardt, and A. Bernasconi. Automated detection of cortical dysplasia type II in MRI-negative epilepsy. *Neurology*, 83(1):48–55, 2014.

[40] L. Hviid, B. Healy, D. Rintell, T. Chitnis, H. Weiner, and B. Glanz. Patient reported outcomes in benign multiple sclerosis. *Multiple Sclerosis*, 17(7):876–884, 2011.

[41] K. Johnson, B. Brooks, J. Cohen, C. Ford, J. Goldstein, R. Lisak, L. Meyers, H. Panitch, J. Rose, and R. Schiffer. Copolymer 1 reduces relapse rate and improves disability in relapsing-remitting multiple sclerosis: results of a phase iii multicenter, double-blind placebo-controlled trial. the copolymer 1 multiple sclerosis study group. *Neurology*, 45(7):1286–1276, 1995.

[42] L. Kappos, M. Freedman, C. Polman, G. Edan, H. Hartung, D. Miller, X. Montalban, F. Barkhof, E. Radu, and L. Bauer. Effect of early versus delayed interferon beta-1b treatment on disability after a first clinical event suggestive of

multiple sclerosis: a 3-year follow-up analysis of the BENEFIT studys. *Lancet*, 370(9585):389–397, 2007.

[43] D. Killer and N. Friedman. Probabilistic graphical models: Principles and techniques. *MIT Press*, 2009.

[44] C. Krishnan, A. Kaplin, R. Brodsky, D. Drachman, R. Jones, D. Pham, N. Richert, C. Pardo, D. Yousem, and E. Hammond. Reduction of disease activity and disability with high-dose cyclophosphamide in patients with aggressive multiple sclerosis. *Arch Neurol.*, 65(8):1044–1051, 2008.

[45] P. Krsek, B. Maton, B. Korman, E. Pacheco-Jacome, P. Jayakar, C. Dunoyer, et al. Different features of histopathological subtypes of pediatric focal cortical dysplasia. *Annals of Neurology*, 63(6):758–769, 2008.

[46] J. Kurtzke. Rating neurologic impairment in multiple sclerosis: An expanded disability status scale (edss). *Neurology 33 (11)*, pages 1444–1452, 1983.

[47] R. I. Kuzniecky and A. Barkovich. Malformations of cortical development and epilepsy. *Brain and Development*, 23(1):2 – 11, 2001.

[48] P. Kwan and M. J. Brodie. Early identification of refractory epilepsy. *New England Journal Of Medicine*, 342(5):314–319, 2000.

[49] A. Lepp, J. E. Barkley, and A. C. Karpinski. The relationship between cell phone use and academic performance in a sample of us college students. *Sage Open*, 5.1:2158244015573169, 2015.

[50] J. T. Lerner et al. Assessment and surgical outcomes for mild type I and severe type II cortical dysplasia: a critical review and the UCLA experience. *Epilepsia*, 50(6):1310–1335, 2009.

[51] L. Liang and V. Cherkassky. Connection between svm+ and multi-task learning. *IEEE World Congress on Computational Intelligence*, pages 2048–2054, 2008.

[52] M. Lichman. UCI machine learning repository, 2013.

[53] I. G. V. N. G. M. Lykourentzou, Ioanna and V. Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers and Education*, 53, no. 3:950–965, 2009.

[54] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1, no. 14:281–297, 1967.

[55] K. J. Miller, M. denNijs, P. Shenoy, J. W. Miller, R. P. N. Rao, and J. G. Ojemann. Real-time functional brain mapping using electrocorticography. *NeuroImage*, 37(2):504 ? 507, 2007.

[56] E. Mowry, M. Pesic, B. Grimes, S. Deen, P. Bacchetti, and E. Waubant. Demyelinating events in early multiple sclerosis have inherent severity and recovery. *Neurology*, 72(7):602–608, 2009.

[57] V. Nair and G. E. Hinton. Implicit mixtures of restricted boltzmann machines. *dvances in neural information processing systems*, pages 1145–1152, 2009.

[58] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9.2:249–265, 2000.

[59] Q. Nguyen, H. Valizadegan, and M. Hanskrecht. Learning classification with auxiliary probabilistic information. *IEEE 11th International Conference on Data Mining (ICDM)*, pages 477–486, 2011.

[60] C. Nordahl, D. Dierker, I. Mostafavi, et al. Cortical folding abnormalities in autism revealed by surface-based morphometry. *J Neurosci.*, 27(43):11725–11735, 2007.

[61] J. Noseworthy, M. Vandervoort, C. Wong, and G. Ebers. Interrater variability with the expanded disability status scale (edss) and functional systems (fs) in a multiple sclerosis clinical trial. *Neurology 40.6*, page 971, 1990.

[62] P. P. of Relapses and D. by Interferon beta-1a Subcutaneously in Multiple Sclerosis) Study Group. Randomised double-blind placebo-controlled study of interferon beta-1a in relapsing/remitting multiple sclerosis. *Lancet*, 352(9139):1498–1504, 1998.

[63] J. Paisley, C. Wang, D. Blei, and M. I. Jordan. A nested hdp for hierarchical topic models. *arXiv preprint arXiv:1301.3570*, 2013.

[64] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE transactions on Knowledge and Data Engineering*, 22.10:1345–1359, 2010.

[65] D. Preston, C. Brodley, R. Khardon, D. Sulla-Menashe, and M. Friedl. Redefining class definition using constraint-based clustering:an application to remote sensing of the earth's surfaces. *KDD*, pages 823–832, 2010.

[66] R. P. R, B. Fischl, V. C. V, N. Makris, and P. E. Grant. A methodology for analyzing curvature in the developing brain from preterm to adult. *International Journal of Imaging Systems and Technology*, 18(1):42–68, 2008.

[67] S. R., M. A, and H. G. Restricted boltzmann machines for collaborative filtering. *Proceedings of the 24th international conference on Machine learning*, pages 791–798, 2007.

[68] C. E. Rasmussen. Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*, pages 63–71, 2004.

[69] L. Rimol, R. Nesvg, D. Hagler Jr., et al. Cortical volume, surface area, and thickness in schizophrenia and bipolar disorder. *Biological Psychiatry*, 71(6):552–560, 2012.

[70] J. C. Ross and J. G. Dy. Nonparametric mixture of gaussian processes with constraints. *Proceeding of the 30th International Conference on Machine Learning*, 2013.

[71] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1:318–362, 1986.

[72] P. Rzezak, F. L. D. P. Squarzoni, T. de Toledo Ferraz Alves, J. Tamashiro-Duran, S. R. C. M. Bottino, P. A. Lotufo, P. R. Menezes, M. Scazufca, and G. F. Busatto. Relationship between brain age-related reduction in gray matter and educational attainment. *PLoS ONE*, 10(10):1–15, 2015.

[73] D. H. Salat, R. L. Buckner, A. Snyder, et al. Thinning of the cerebral cortex in aging. *Cerebral Cortex*, 14(7):721–730, 2004.

[74] J. Sethuraman. A constructive definition of dirichlet priors. *FLORIDA STATE UNIV TALLAHASSEE DEPT OF STATISTICS*, No. FSU-TR-M-843, 1991.

[75] A. M. Shahiri and W. Husain. A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, 72:414–422, 2015.

[76] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Computing: Explorations in the Microstructure of Cognition*, 1, 1986.

[77] Y. W. Teh. Dirichlet process. *Encyclopedia of machine learning. Springer US*, pages 280–287, 2010.

[78] J. F. Tellez-Zenteno, R. Dhar, and S. Wiebe. Long-term seizure outcomes following epilepsy surgery: a systematic review and meta-analysis. *Brain*, 128(5):1188–1198, 2005.

[79] T. Thesen, B. Quinn, C. Carlson, et al. Detection of epileptogenic cortical malformations with surface-based MRI morphometry. *PLoS ONE*, 6(2):e16430, 2011.

[80] J. F. Tllez-Zenteno, F. M.-A. L. H. Ronquillo, and S. Wiebe. Surgical outcomes in lesional and non-lesional epilepsy: A systematic review and meta-analysis. *Epilepsy Research*, 89(23):310–318, 2010.

[81] A. Tsanas, M. Little, P. E. McSharry, and L. O. Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *Biomedical Engineering, IEEE Transactions on 57*, no. 4:884–893, 2010.

[82] M. J. Tullman. Overview of the epidemiology, diagnosis, and disease progression associated with multiple sclerosis. *Am J Manag Care*, 19 (2 Suppl):S15–20, 2013.

[83] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22.5:544–557, 2009.

[84] V. N. Vapnik. *Estimation of dependences based on empirical data.* New York: Springer-Verlag, 1982.

[85] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. Class imbalance, redux. *In Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 754–763, 2011.

[86] Y. Wang, R. Khardon, and P. Protopapas. Shift-invariant grouped multi-task learning for gaussian processes. *Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg*, pages 418–434, 2010.

[87] Z. I. Wang, A. V. Alexopoulos, S. E. Jones, Z. Jaisani, I. M. Najm, and R. A. Prayson. The pathology of magnetic-resonance-imaging-negative epilepsy. *Mod Pathol*, 26(8):1051–1058, 2013.

[88] A. Waters and R. Miikkulainen. Grade: Machine learning support for graduate admissions. *AI Magazine*, 35.1:64, 2014.

[89] H. Weiner. A shift from adaptive to innate immunity: a potential mechanism of disease progression in multiple sclerosis. *Journal of Neurology*, 255:3–11, 2008.

[90] P. Wu and T. Dietterich. Improving svm accuracy by training on auxiliary data sources. *Proceedings of the 21st International Conference on Machine Learning*, page 110, 2004.

[91] E. P. Xing, M. I. Jordan, and R. Sharan. Bayesian haplotype inference via the dirichlet process. *Journal of Computational Biology*, 14.3:267–284, 2007.

[92] C. Xu, T. Dacheng, and C. Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[93] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:35–63, 2007.

[94] F. Y. and D. Haussler. Unsupervised learning of distributions on binary vectors using 2-layer networks. *NIPS*, pages 912–919, 1992.

[95] J. Yuan, Y. Chen, and E. Hirsch. Intracranial electrodes in the presurgical evaluation of epilepsy. *Neurological Sciences*, 33(4):723?729, 2012.

[96] Y. Zhao, B. Ahmed, T. Thesen, K. Blackmon, J. G. Dy, and C. E. Brodley. A non-parametric approach to detect epileptogenic lesions using restricted boltzmann machines. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 373–382, 2016.

[97] Y. Zhao, C. Brodley, T. Chitnis, and B. C. Healy. Addressing human subjectivity via transfer learning: An application to predicting disease outcome in multiple sclerosis patients. *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 965–973, 2014.

[98] Y. Zhao, T. Chitnis, B. C. Healy, J. G. Dy, and C. E. Brodley. Domain induced dirichlet mixture of gaussian processes: An application to predicting disease progression in multiple sclerosis patients. *IEEE Internaltional Conference on Data Mining (ICDM)*, 16:1129–1134, 2015.

[99] Y. Zhao, B. Lackaye, J. G. Dy, and C. E. Brodley. A quantitative machine learning approach to master students admission for professional institutions. *Under review*, 2017.