Predictive Learning with Heterogeneity in Populations

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Anuj Karpatne

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Prof. Vipin Kumar, Advisor

October, 2017

Acknowledgements

The joyous journey of my PhD couldn't have been possible without the support of a number of people in my life, whom I am grateful towards. Foremost, I am deeply thankful to my advisor, Prof. Vipin Kumar, who has greatly shaped me who I am as a thinker and researcher today. His passion for research is infectious and every interaction with him has not only enriched my understanding of data mining concepts, but also the ability to look at a research problem with an open mind from multiple perspectives, and to venture on directions that are most rewarding. Among his many qualities, the one I am thankful for the most is the immense trust he had in my ability to do research at every stage of my PhD, from the day I joined his group to my date of defense. This really pushed me to explore my own abilities with a "can do" attitude that I cherish the most. He will continue to be a source of inspiration and if I am able to carry even 1% of his qualities in the future, I'll consider it success.

During graduate school, I had the opportunity to learn from a number of great educators, Professors Vipin Kumar, Arindam Banerjee, Ravi Janardan, and John Carlis, through coursework or otherwise. Learning from them was a joyride that has instilled in me a passion for life-long learning and teaching. I would also like to thank Professors Rui Kuang, Snigdhansu Chatterjee, and Arindam Banerjee for taking the time to serve on my thesis proposal and final exam committees.

Working in Kumar Research Group, I have had the pleasure of interacting with a diverse group of mentors, collaborators, and colleagues, which has substantially helped me to grow as a researcher. I would like to thank Shyam Boriah, Michael Steinbach, Gowtham Atluri, Arindam Banerjee, and Pang-Ning Tan for several insightful discussions we have had in the course of many years. I would also like to thank Miriam

Marlier, Dennis Lettenmaier, and Imme Ebert-Uphoff for working together on exciting research collaborations. I am grateful to my fellow lab-mates and friends, Ankush Khandelwal, Xi Chen, Varun Mithal, Saurabh Agrawal, Guruprasad Nayak, Xiaowei Jia, Ivan Brugere, Yashu Chamber, and Lydia Manikonda, for creating a conducive work atmosphere. I have also had the opportunity to interact with some really bright undergraduate students who have worked as interns on our research projects: Stryker Thompson, Vishnu Arun, Robert Leunberger, Yizheng Ding, Nyssa Capman, Mace Blank, and Reid Anderson. They have been amazing to work with.

Nothing that I have ever accomplished could have been possible without the love and support of my parents and my brother and his wife, who have been anchors in my life. I am also deeply grateful to my cousins and my extended family who have been very close to me. Finally, I would like to thank all my friends in graduate school, Akash Agrawal, Rahul Saladi, Naveen Elangovan, Arpan Bandyopadhyay, Yokesh Kumar, Pradeep Mohan, Narendra Singh, and Priya Mohan, who have made my PhD journey a memorable experience.

Dedication

To the Universe, of which we are minuscule dots.

Abstract

Predictive learning forms the backbone of several data-driven systems powering scientific as well as commercial applications, e.g., filtering spam messages, detecting faces in images, forecasting health risks, and mapping ecological resources. However, one of the major challenges in applying standard predictive learning methods in real-world applications is the heterogeneity in populations of data instances, i.e., different groups (or populations) of data instances show different nature of predictive relationships. For example, different populations of human subjects may show different risks for a disease even if they have similar diagnosis reports, depending on their ethnic profiles, medical history, and lifestyle choices. In the presence of population heterogeneity, a central challenge is that the training data comprises of instances belonging from multiple populations, and the instances in the test set may be from a different population than that of the training instances. This limits the effectiveness of standard predictive learning frameworks that are based on the assumption that the instances are independent and identically distributed ($i.i.d$), which are ideally true only in simplistic settings.

This thesis introduces several ways of learning predictive models with heterogeneity in populations, by incorporating information about the *context* of every data instance, which is available in varying types and formats in different application settings. It introduces a novel multi-task learning framework for problems where we have access to some ancillary variables that can be grouped to produce homogeneous partitions of data instances, thus addressing the heterogeneity in populations. This thesis also introduces a novel strategy for constructing mode-specific ensembles in binary classification settings, where each class shows multi-modal distribution due to the heterogeneity in their populations. When the context of data instances is implicitly defined such that the test data is known to comprise of contextually similar groups, this thesis presents a novel framework for adapting classification decisions using the group-level properties of test instances. This thesis also builds the foundations of a novel paradigm of scientific discovery, termed as theory-guided data science, that seeks to explore the full potential of data science methods but without ignoring the treasure of knowledge contained in scientific theories and principles.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

From satellites in space to wearable computing devices and from credit card transactions to electronic health-care records, the deluge of data [1–3] has pervaded every walk of life. Our ability to collect, store, and access large volumes of information is accelerating at unprecedented rates with better sensor technologies, more powerful computing platforms, and greater on-line connectivity. With the growing size of data, there has been a simultaneous revolution in the computational and statistical methods for processing and analyzing data, collectively referred to as the field of data science. These advances have made long-lasting impacts on the way we sense, communicate, and make decisions [4], a trend that is only expected to grow in the foreseeable future. Indeed, the start of twenty-first century may well be remembered in history as the "golden age of data science."

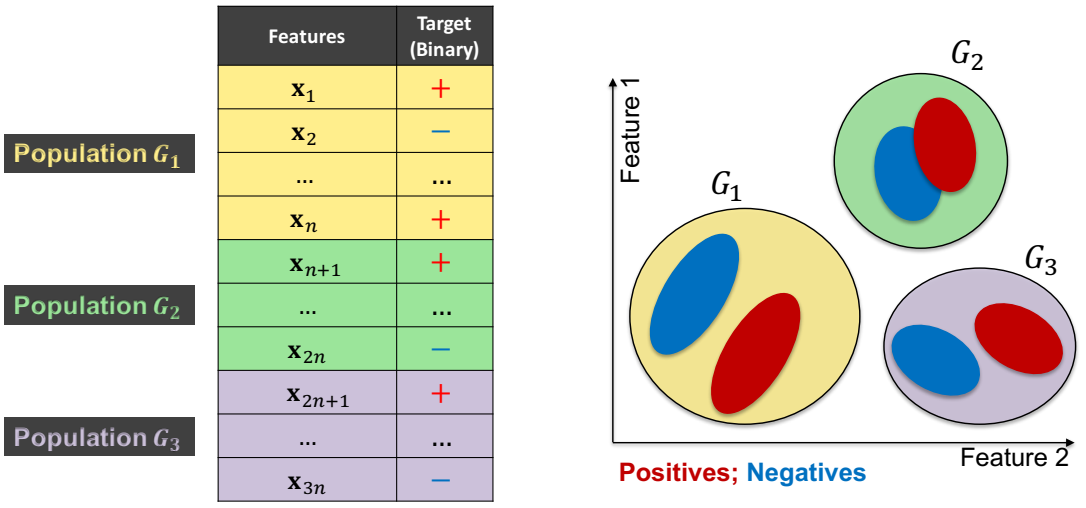
A unique ability of data science methods is to automatically extract patterns and models from large volumes of data, using a variety of methods and modeling paradigms. One of the paradigms of data science that has found great success in several applications is the paradigm of *predictive learning*. The basic goal in predictive learning is to estimate the value of a target variable, Y, (also referred to as the output or the response variable), using observations of other input variables, X, referred to as features, attributes, or explanatory variables. For example, given information about the age,

medical history, disease symptoms, and diagnosis reports of a person (treated as attributes), we can predict their chances of being infected by a certain disease (treated as output). Predictive learning forms the backbone of several data-driven systems powering scientific as well as commercial applications, e.g., filtering spam messages, detecting faces in images, forecasting health risks, and mapping ecological resources. When the target variable is categorical in nature and only takes discrete values (e.g., $Y \in \{+1, 1\}$ or $Y \in \{1, 2, \ldots, K\}$), the predictive learning problem is called *classification*. Otherwise, when the target variable is allowed to take continuous values ($Y \in \mathbb{R}$), we call it *regression*.

The general framework for predictive learning involves finding predictive relationships between input and output variables by sifting through several examples of input-output pairs, termed as training data. Formally, given a training data set $\mathcal{D} = \{\mathbf{x}, y\}_1^n$, we aim to learn a mapping, $f : X \rightarrow Y$, such that $f(.)$ can be applied on any unseen test instance, $\mathbf{x}$, to predict the value of its target variable $y = f(\mathbf{x})$. A variety of approaches have been developed to learn predictive models from training data, ranging from simple solutions such as perceptrons and decision trees to advanced algorithms such as support vector machines and deep neural networks [5]. Many of these methods are based on strong statistical foundations that ensure that models trained over a training set are *generalizable* over unseen instances encountered during testing.

One of the underlying assumptions in standard frameworks for predictive learning is that the data instances in the training set are identical to each other, and belong to a common yet unknown population. Hence, the training instances are generally considered independent and identically distributed, commonly referred to as the '$i.i.d$' assumption. Furthermore, it is assumed that instances in the training set are fairly reflective of the distribution of unseen test instances encountered in the future. In other words, the training and test sets are assumed to contain instances belonging to a single common population, thus sharing identical (or homogeneous) relationships between input and output variables.

Although standard predictive learning frameworks work well under these assumptions of homogeneity, they are routinely violated in a number of real-world applications. This is because most real-world systems are composed of a plurality of data populations, with varying properties of predictive relationships in every population. For example,



(a) List of data instances

(b) Distribution of instances in feature space

Figure 1.1: Illustration of population heterogeneity and its impact on predictive learning. Figure 1.1(a) shows a tabular view of data instances for a binary classification problem from three different populations, G_1 to G_3. Figure 1.1(b) shows the distribution of the classes in the feature space for each population.

in the problem of predicting a patient's risk for a certain disease (output) given their healthcare records (input variables), different populations of subjects from diverse ethnic backgrounds and living conditions can show huge variations in the relationship between their healthcare records and disease risks. In fact, two patients with the same healthcare record (input variables) can be associated with different risks for a disease (output) depending on the population that they belong.

We refer to the scenario where: (a) the training data is comprised of instances from multiple populations, and (b) the test set belongs to a population different from that of the training set as *population heterogeneity*. Figure 1.1 illustrates a toy example of population heterogeneity for a binary classification problem (involving two classes: $+$ and $-$). We can see in Figure 1.1(a) that the data set comprises of instances from three different populations, G_1 to G_3, each consisting of n instances. Figure 1.1(b) shows that the positive and negative classes of each population (shown using colored circles) have widely varying distributions in the feature space. While the classes may be separable in some populations (e.g., G_1 and G_3), they may be more difficult to separate in other populations (e.g., G_2). Some of the real-world problems that are impacted by

population heterogeneity include:

- Predicting the risk of a disease (output) given the healthcare records (input variables) of a patient. In this problem, different populations of subjects may require different models of disease risk given the input variables.
- Estimation of geoscience variables such as health of vegetation or presence of surface water (output) using remote sensing data (input variables) observed at every location on the Earth at every time-step. It is well-known that the characteristics of predictive relationships for geoscience variables vary widely across geographic space and time, due to changes in geography, topography, types of soil, climatic conditions, and seasonal cycles [6].
- Recommending posts on social networking websites (output) using information about users such as their age or level of education (input variables). In this case, depending on their social affiliations and usage history, the preferences of two users may be different even if they have similar age or education.

1.2 Challenges and Objective

There are a number of challenges in learning predictive models with heterogeneity in populations of data instances.

- First, the training data comprises of instances from not just one distribution but several distributions juxtaposed together. For example, in classification problems, every class may appear as multiple sub-categories or modes (shown as red and blue regions in the illustration shown in Figure 1.1(b)). In the presence of multimodality within the classes, there may be imbalance among the distribution of different modes in the training set. Hence, some of the modes may be underrepresented during training, resulting in poor performance on those modes during the testing stage. This may be critical in the presence of anomalous modes that are rare but important to detect, e.g., ecosystem disturbances and weather extremes.
- Second, while some of the modes of a particular class may be easy to distinguish from modes of the other class, there may be modes that participate in class confusion, i.e., reside in regions of feature space that overlap with instances from other

classes. The presence of such overlapping modes can degrade the learning of any classification model trained across all modes of every class.

- Third, even if we are able to learn a predictive model that shows reasonable performance on the training set, the test set may have a completely different distribution of data instances than the training set, as the populations of training and test sets can be different. Hence, the training performance can be quite misleading as it may not always be reflective of the performance on test instances.

These challenges severely restrict the applicability of standard predictive learning frameworks when applied to scenarios involving heterogeneity in populations. In such settings, it is evident that the set of feature values observed at a data instance are not sufficient for estimating the value of its target variable without ambiguity. This is because along with the observed feature values, the population that the data instance belongs plays a decisive role in making predictions of the target variable. Hence, what is needed is a way to incorporate knowledge about the *context* in which a data instance is observed, e.g., using ancillary variables other than its features that can help in inferring its population. The goal of this thesis is to address the challenges associated with population heterogeneity in predictive learning by incorporating information about the context of data instances, which are available in varying forms in different application settings.

1.3 Thesis Contributions and Organization

This thesis presents several approaches for predictive learning with heterogeneity in populations, that incorporate information about the context of data instances in predictive learning frameworks. Following are the main contributions of this work and the organization of the remainder of this thesis.

- Chapter 2 presents a brief review of the landscape of methods for handling the presence of population heterogeneity in different predictive learning settings. It introduces the concepts of explicit context (where ancillary variables can be directly used to estimate the nature of predictive relationships at every instance) and implicit context (where ancillary variables are absent or their influence on predictive

relationships is latent), which are used to provide a systematic categorization of related approaches relevant to the contributions presented in this thesis.

- Chapter 3 presents a novel approach for handling population heterogeneity when contextual information of training and test instances are explicitly available as ancillary variables, that can be grouped using clustering methods to form homogeneous partitions of the data. The proposed approach uses a multi-task learning formulation to jointly address the challenge of population heterogeneity as well as paucity of labeled data, common in several real-world problems [7].

- Chapter 4 presents a novel ensemble learning framework for incorporating the multi-modal structure of classes in binary classification settings, when both classes show heterogeneity in populations. The proposed framework, termed as Bipartite One-vs-One (BOVO) [8], uses mode-specific information to provide superior predictive performance than traditional ensemble learning methods. It further offers interpretability of results by providing additional information about the mode affiliations of every test instance.

- Chapter 5 presents a novel scheme for adapting the classification responses of mode-specific ensembles using group-specific information of test instances [9]. By inferring the implicit context in a group of test instances by observing their distribution in the feature space, it is able to appropriately select ensemble classifiers that are most relevant in the context of the test group.

- An underlying theme of research in learning with population heterogeneity is a systematic way of incorporating domain (or scientific) knowledge in predictive learning frameworks, for inferring the relevant context of data instances. Chapter 6 presents a broader paradigm developed as part of this thesis on combining the strengths of scientific knowledge with data science methods, termed as theory-guided data science [10]. This chapter presents a review of this emerging paradigm of research and discusses several research themes under this paradigm that is being pursued in varied scientific and engineering disciplines.

- Chapter 7 presents concluding remarks and discusses future directions of work.

Chapter 2

Background

This chapter presents a landscape of predictive learning methods for handling the challenge of heterogeneity in populations. A sufficient requirement for incorporating the effects of population heterogeneity in predictive learning is to exactly known which population every instance belongs. Unfortunately, this information is seldom available at the required level of detail in most practical settings. This is because the total number of populations present in a real-world system, let alone their distributions, is often an unknown quantity. In fact, populations can be defined at varying levels of granularity depending on the requirements of the application and availability of data instances. For example, in order to predict housing costs based on affinity to economic assets, we can build predict models at the level of counties, districts, states, or countries, each resulting in populations at different spatial scales. Hence, information about data populations and the affiliation of instances to populations is mostly hidden and needs to be inferred from the data, often with the help of domain or background knowledge.

The background of every data instance is often captured in the form of *ancillary variables*, Z, that are recorded along with the attributes at every observation. Some examples of ancillary variables in spatio-temporal settings include the spatial identifiers of the location at which the observation was taken (e.g., coordinates such as latitude and longitude of the location), or the time-stamp of observation. Other examples of ancillary variables include the history of observations at every instance (e.g., past medical records of every subject), the genetic profile of subjects, or the structure of relationships of every instance with respect to other instances (e.g., in social networks). It is

evident that the nature and formats of ancillary variables widely vary across different application settings. If used appropriately, they can provide the necessary information about the *context* of every data instance that can help in addressing the heterogeneity in populations. In the following, we describe two basic types of contexts defined by ancillary variables, that can be used in predictive learning problems in a number of ways.

- *Explicit Context:* In some cases, the values of ancillary variables, Z, are directly related to the nature of predictive relationships at every instance. For example, in the problem of predicting traits of a plant such as its leaf area and seed mass (output) given environmental factors (input variables), we can consider the species or any other phylogenetic information of the plant as its ancillary variable. Since plants belonging to the same species are expected to show similar behavior of plant traits given environmental conditions, we can use the species affiliation of every plant as an explicit context for incorporating the effects of population heterogeneity. In other cases, the dependence of predictive relationships on values of ancillary variables may not be in absolute terms, but in relation to the values of other instances. For example, consider the problem of predicting a spatial target variable such as land cover using observations at every location. While the absolute value of the spatial coordinates may not provide information on how the predictive relationships behave over space, we know that nearby locations mostly have similar target values due to the spatial auto-correlation in the data. Hence, the relative values of spatial coordinates contain useful information for learning predictive relationships in spatial settings, and thus can be treated as ancillary variables providing explicit context.

- *Implicit Context:* In most predictive learning problems, we do not know the right ancillary variables influencing heterogeneity in populations, as they are often unobserved and implicitly defined. Moreover, even if we have access to some ancillary variables about every observation, we may not know the nature of dependence between the values of ancillary variables and the properties of predictive relationships. For example, consider the problem of predicting a time-varying quantity such as the number of Web queries containing a certain keyword, e.g., "Donald

Trump," using input variables such as media posts on other topics. In such complex problems, if we treat time as an ancillary variable, we may be able to enforce smoothness in predictions at nearby time-steps, but long-term trends in the nature of predictive relationships may not be fully understood. This requires more ingenious ways of using the ancillary variables (if at all available) for handling the challenge of heterogeneity in populations.

A variety of approaches have been explored for incorporating both these types of context in predictive learning problems, using different ancillary variables in various applications. Figure 2.1 provides a basic taxonomy of these approaches, which are briefly reviewed in the following sections.

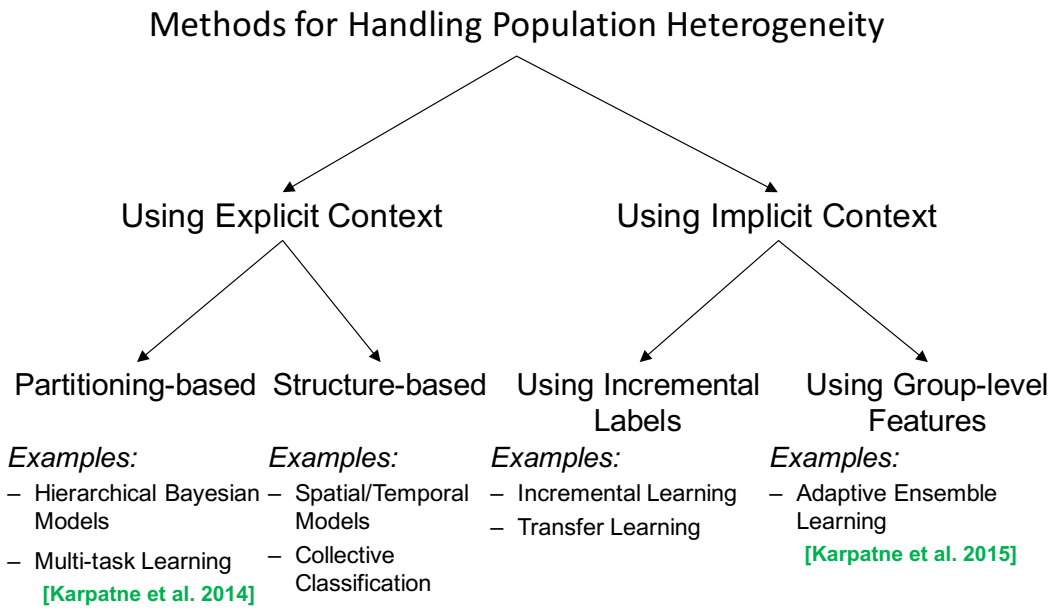


Figure 2.1: Taxonomy of approaches for handling population heterogeneity in predictive learning problems.

2.1 Using Explicit Context

In the presence of ancillary variables that are directly relevant for inferring the effects of population heterogeneity, one of the simplest ways of including them in predictive modeling frameworks is to concatenate them with the set of features used as input variables. Although, in theory, the expanded set of features and ancillary variables may be sufficient for an adequate learning algorithm to extract the necessary predictive relationships when supplied with ample training instances, in practice, treating ancillary variables as just other input variables may not be the most effective strategy for addressing population heterogeneity when training data is limited in size. This is especially true in applications where the ancillary variables have widely varying types and formats than the explanatory variables, e.g., in the form of networks or time-series [11], and a simple concatenation may not be useful or even possible.

Indeed, there are several ways of making better use of ancillary variables in predictive learning frameworks than simple concatenation with features, where the ancillary variables are treated separately to extract the right context for learning predictive relationships. There are two broad categories of approaches for using the explicit context of ancillary variables in predictive learning frameworks: (a) methods that partition the data instances into groups with same (or similar) ancillary variables, and then learn different predictive models for every partition, and (b) methods that utilize the structure among the instances based on the values of their ancillary variables, to constrain the predictions at instances relative to other instances. We describe relevant methods in both these categories of approaches below.

2.1.1 Partioning-based Methods

In this category of approaches, the primary objective is to construct partitions of data instances such that every partition is homogeneous in nature and includes instances only from a single population. For example, we can cluster Z to construct homogeneous partitions of instances, under the assumption that instances with similar ancillary variables belong to the same population. The learning of a predictive model at every partition of the data can then be considered as a separate task. The use of multi-task learning formulations to jointly learn predictive models at all partitions, while sharing the learning

across related tasks, has been explored in Karpatne et al. [7]. Another partitioning-based approach for handling the heterogeneity in populations is to use hierarchical Bayesian models, where the effect of ancillary variables on predictive relationships is modeled using different strategies such as mixed effects models and random effects models [12].

2.1.2 Structure-based Methods

Another category of approaches for informing predictive decisions using Z is to extract structured dependencies among the data instances based on the relative values of their ancillary variables. For example, given the location coordinates of data instances, we can construct a neighborhood graph where adjacent nodes correspond to instances that are spatially close to each other. Graphical models such as Markov Random Fields can then be used to enforce spatial contiguity constraints on the predictions made at adjacent locations. Similarly, Hidden Markov Models can be used for enforcing temporal consistency in the predictions made at nearby time stamps. Another area of work that makes use of structured dependencies among instances is collective classification methods [13], where the node attributes are used together with the properties of adjacent nodes to make predictions of a target variable.

2.2 Using Implicit Context

When the influence of ancillary variables on the nature of predictive relationships is implicit in nature, we can either use incremental labels to dynamically adapt the predictions to changing populations in the test set, or develop methods that use the group-level features of test instances, e.g., their distribution in the feature space, to adapt predictions without using incremental labels. Both these categories of methods are discussed in the following.

2.2.1 Using Incremental Labels

If it is possible to collect new labeled samples from current testing scenarios in an on-line fashion, we can adapt the predictive model learned from the original training set to the dynamic needs of changing populations in the test set. The framework of incremental learning [14], which attempts to capture the notion of concept drift

(changing characteristics of class distributions over time) using incremental labels, is directly relevant in this category. Another relevant area of work is that of transfer learning [15], where the learning from a source task with ample availability of labeled data can be transferred to a target task with limited availability of labeled data. In our problem setting, the source task corresponds to the learning of a predictive model over the original training set, and the target task corresponds to learning predictive models on future testing scenarios with limited availability of labeled samples.

2.2.2 Using Group-level Properties of Unlabeled Instances

In the absence of incremental labels from future testing scenarios, we can address the challenge of population heterogeneity by observing the distribution of a group of unlabeled test instances and identifying the population of training instances that it closely resembles. In this way, we can adapt predictive models to future testing scenarios without using incremental labels, which are challenging to obtain in most real-world applications. In a recent work by Karpatne et al. [9], group-level properties of test instances were extracted using a mixture of Gaussian models learned from the training data. These group-level properties were then used to adapt the decisions of an ensemble of predictive models to the specific requirements of a given group of test instances.

Chapter 3

Multi-task Learning using Ancillary Variables

3.1 Introduction

In order to learn predictive relationships in the presence of population heterogeneity, one approach can be to first divide the entire data set into *homogeneous* partitions by grouping instances based on the values of their ancillary variables. This would result in groups of instances with similar values of ancillary variables, which are likely to share common predictive relationships between explanatory and target variables. If sufficient training data is available for every such data partition, we can conveniently learn a predictive model for every partition independently of the other partitions. However, in a number of real-world problems, training data is often limited because obtaining ground-truth labels is time-consuming, labor-intensive, and expensive. This when coupled with the challenge of population heterogeneity makes the learning of independent predictive models at every data partition prone to over-fitting, leading to poor generalization performance. This is especially true for data partitions that suffer from paucity of training data, which are insufficient for learning suitably complex predictive models required for the problem. Hence, there exists a trade-off between increasing the amount of heterogeneity explained by the model and reducing the model complexity. This motivates the need for an approach that that can utilize the structure in the data instances and their partitions for regularized learning of heterogeneous relationships.

There are various forms of structure that exist in real-world datasets. As an example, remote sensing datasets show a strong structure in space and time, and the presence of multiple types of vegetation on land dictates a structured similarity among locations belonging to similar vegetation (land cover) types. Social network datasets on the other hand express the structure among users (data instances) using graph-based network representations. The structure among the data instances can be leveraged for reducing the model complexity, by constraining the model search space. As an example, we can penalize the learning of widely dissimilar relationships at structurally similar partitions of the data, leading to a lower model complexity as opposed to learning a model at each partition independently.

In this chapter, we propose a multi-task learning framework for learning predictive relationships in the presence of data heterogeneity and insufficient training data, which utilizes the structure among data partitions for robust predictive learning. Specifically, the proposed framework comprises of three key steps: (a) partitioning the heterogeneous data into relatively homogeneous data partitions, (b) extracting the structure among the data partitions, and (c) utilizing the structure among the partitions for regularizing the learning of a predictive model at each data partition. By performing a series of experiments to evaluate our performance in comparison with the baseline approaches, we show that this proposed method: (a) captures meaningful information about the heterogeneity in the data, (b) improves the prediction performance in the presence of data heterogeneity, (c) is robust to over-fitting in scenarios with limited training data, and (d) is robust to the choice of the number of partitions used to represent the heterogeneity in the data.

The remainder of this chapter is organized as follows: Section 3.2 provides a brief overview of related work. Section 3.3 describes the proposed approach. Section 3.4 discusses the data. Section 3.5 discusses the evaluation setup. Section 3.6 provides experimental results. Section 3.7 includes concluding remarks and discusses directions for future work.

3.2 Related Work

Existing methods that utilize structure in the data can be broadly classified into the following three categories: (i) methods that utilize structure among the explanatory variables, (ii) methods that utilize structure among the response variables, and (iii) methods that utilize structure among the data instances. In this section, we briefly review the literature pertaining to each of the three categories above. Out of these three categories, methods that utilize structure among the data instances for addressing heterogeneity are most related to this chapter.

Methods that utilize structure among the explanatory variables aim at extracting discriminative features from explanatory variables which are useful in predictive learning. In this context, dimensionality reduction and subspace monitoring techniques have been explored for high-dimensional predictor datasets [16]. Further, shrinkage estimators encompass a broad family of methods that aim at regularizing ill-posed problems by introducing additional information, such as the desired structural properties of explanatory variables [17]. Methods that have utilized structure among multivariate response variables include structured output regression techniques, that have been mainly explored for localization and image restoration applications in computer vision and image processing [18]. Multi-label learning has been proposed for classification scenarios where the classes are not mutually exclusive and there is a structure among the classes [19].

The family of methods that is closest to the problem being addressed in this chapter includes those that incorporate structure among data instances or their partitions. Methods that perform semi-supervised learning utilize information about the structure in unlabeled data, which can then be used to assist a supervised learning task [20]. However, they do not explore the heterogeneity in relationships between explanatory and response variables, which requires learning a different model for each partition of the data. On the other hand, transfer learning and multi-task learning aim at utilizing the knowledge learned in a source task for its application in a target task [15] or for sharing the learning among multiple related tasks [21]. For instance, the similarity among related tasks can be represented in the form of a graph which can then be used for regularizing the learning over each individual task [22, 23]. Further, task clustering has been used for representing task similarities in multi-task learning [24]. However, these

approaches need explicit knowledge about task definitions and prior information about the number of tasks and their structure. Obtaining information about task divisions can be difficult in real-world scenarios where the inherent heterogeneity is implicit and needs to be extracted.

3.3 Proposed Approach

We first present a generic formulation of the proposed multi-task learning framework in section 3.3.1, and then subsequently provide a specific instantiation of the proposed framework for its application in estimating forest cover using remote sensing datasets in section 3.3.2.

3.3.1 Generic Formulation

Let $y \in \mathbb{R}$ be the response variable that needs to be predicted using $\boldsymbol{x} \in \mathbb{R}^d$, which is a d-dimensional vector comprising of d explanatory variables. Let $\mathcal{X} = \{\boldsymbol{x_i}\}_1^N$ and $\mathcal{Y} = \{y_i\}_1^N$ be the set of explanatory variables and response variables over N data instances, respectively. Let there exist a heterogeneity among the N data instances, implying that different segments of $(\mathcal{X}, \mathcal{Y})$ share different relationships between $\boldsymbol{x}$ and y. Furthermore, let each data instance, $(\boldsymbol{x_i}, y_i)$, be associated with an additional set of structural variables, $\boldsymbol{z_i}$, that capture information about the structural dependencies of $(\boldsymbol{x_i}, y_i)$ with other data instances. The structural variables, $\mathcal{Z} = \{\boldsymbol{z_i}\}_1^N$, thus account for the heterogeneity in the data, and can take different forms depending on the source of heterogeneity being experienced in the application domain.

We consider the scenario where both $\mathcal{X}$ and $\mathcal{Z}$ are available over all N data instances during the training phase, but supervised information about y is available only over a few n data instances, where $n \ll N$. Let $\mathcal{Y}_{tr} = \{y_i\}_1^n$ denote the set of response variables that are available during the training phase. Our objective is to utilize the information in $\mathcal{X}$, $\mathcal{Z}$, and $\mathcal{Y}_{tr}$ for learning relationships between $\boldsymbol{x}$ and y, and use the learning to predict y_i for each $\boldsymbol{x_i} \in \mathcal{X}$.

We present a framework for learning predictive relationships in the presence of heterogeneity and limited training data, which comprises of the following three steps: (a)

partitioning the overall data into homogeneous partitions (whose instances share a common relationship between $\boldsymbol{x}$ and y), (b) learning the structure among the data partitions, and (c) using the structure among the partitions for regularizing the learning of a relationship at each partition of the data. We next provide a brief description of each of the three steps of the generic framework.

In order to group data instances into homogeneous partitions, we make use of the structural variables, $\boldsymbol{z_i} \in \mathcal{Z}$, for assigning every $\boldsymbol{x_i} \in \mathcal{X}$ to a homogeneous data partition comprising of structurally similar data instances (with similar $\boldsymbol{z_i}$ values). With the assumption that structurally similar data instances share similar relationships between $\boldsymbol{x}$ and y, we can cluster $\mathcal{Z}$ into m clusters, $\{\mathcal{Z}_k\}_1^m$, thus partitioning $\mathcal{X}$ into m partitions, $\{\mathcal{X}_k\}_1^m$.

For each data partition, $\mathcal{X}_k$, let $\boldsymbol{Y_k} = \{y_i\}_1^{n_k}$ denote the set of response variables for some n_k instances in $\mathcal{X}_k$, for which training data is available. Let $\boldsymbol{X_k}$ denote the set of explanatory variables for the same n_k instances in $\mathcal{X}_k$, where $\sum_{k=1}^m n_k = n$. We consider learning a generalized linear model [25] at each data partition, $\mathcal{X}_k$, for predicting $\boldsymbol{Y_k}$ given $\boldsymbol{X_k}$.

Let the linear predictor at $\mathcal{X}_k$ be given by:

$$\boldsymbol{\eta_k} = \boldsymbol{X_k}\boldsymbol{\beta_k} \tag{3.1}$$

The expected value of the set of response variables, $\boldsymbol{\mu_k} = \mathrm{E}[\boldsymbol{Y_k}]$, can be written as a function of the linear predictor using a link function, g, in the following fashion:

$$\boldsymbol{\mu_k} = g^{-1}(\eta_k) \tag{3.2}$$

The model parameter can then be estimated by minimizing the negative log-likelihood function of $\boldsymbol{\beta_k}$.

$$\hat{\boldsymbol{\beta_k}} = \min_{\boldsymbol{\beta_k}} -\log P(\boldsymbol{Y_k}|\boldsymbol{\beta_k}) \tag{3.3}$$

However, in scenarios where m is large and n_k is small, learning a unique $\boldsymbol{\beta_k}$ independently at each data partition is prone to over-fitting. Instead, we can make use of the structure among the data partitions, $\{\mathcal{X}_k\}_1^m$, for regularizing our learning of $\boldsymbol{\beta} = \{\boldsymbol{\beta_k}\}_1^m$. Let the structure among the data partitions be represented as an undirected graph,

$\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where the vertices of the graph, $\boldsymbol{V}$, denote the m data partitions, and the edges of the graph, $\boldsymbol{E}$, denote similarities among the data partitions, learned using similarities in the structural variables of the partitions, $\{\mathcal{Z}_k\}_1^m$.

We include the structure among the data partitions, expressed using $\boldsymbol{G}$, as a regularization term in our objective function of minimizing the negative log-likelihood of $\boldsymbol{\beta}$. In particular, we intend to penalize the learning of model parameters, $\boldsymbol{\beta_i}$ and $\boldsymbol{\beta_j}$, if i and j are neighboring data partitions in G but $\boldsymbol{\beta_i}$ is widely different from $\boldsymbol{\beta_j}$. This can be achieved by introducing the squared L_2 distance between $\boldsymbol{\beta_i}$ and $\boldsymbol{\beta_j}$ in our objective function as follows:

$$\min_{\boldsymbol{\beta}} \; -\sum_{k=1}^{m} \log P(\boldsymbol{Y_k}|g^{-1}(\boldsymbol{X_k}\boldsymbol{\beta_k})) + \; \lambda \sum_{(i,j) \,\in\, E} \Big\|(\boldsymbol{\beta_i} - \boldsymbol{\beta_j})\Big\|_2^2 \tag{3.4}$$

where λ is a regularization trade-off parameter. It can be observed that the regularization term in equation 3.4 can be succinctly written as $\boldsymbol{\beta}^T \tilde{\boldsymbol{L}} \boldsymbol{\beta}$, where $\tilde{\boldsymbol{L}}$ is the component-wise unnormalized graph Laplacian of $\boldsymbol{G}$ [26], over each dimension in $\boldsymbol{\beta_k}$ from 1 to d. $\tilde{\boldsymbol{L}}$ can thus be written as

$$\tilde{\boldsymbol{L}} = \boldsymbol{L} \otimes \boldsymbol{I_d} \tag{3.5}$$

where, L is the unnormalized graph Laplacian of $\boldsymbol{G}$, $\boldsymbol{I_d}$ is an identity matrix of dimension d, and $a \otimes b$ denotes the Kronecker product between a and b. Let $\mathbf{Y} = (\mathbf{Y_1^T} \ldots \mathbf{Y_m^T})^\mathbf{T}$ be an $n \times 1$ stacked vector of response values over all data partitions, and $\boldsymbol{X}$ be the design matrix of size: $n \times md$ over all data partitions, represented as

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X_1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{X_2} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{X_m} \end{bmatrix} \tag{3.6}$$

where $\mathbf{0}$ denotes a zero matrix of appropriate dimensions. Equation 3.4 can then be rewritten using matrix notations involving $\boldsymbol{\beta}$, as

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} - \log P(\boldsymbol{Y}|g^{-1}(\boldsymbol{X}\boldsymbol{\beta})) + \lambda \boldsymbol{\beta}^T \tilde{\boldsymbol{L}} \boldsymbol{\beta} \tag{3.7}$$

The solution to equation 3.7 can be found by using gradient descent techniques or the Newton-Raphson method [27].

3.3.2 Specific Formulation

The generic formulation described in section 3.3.1 comprises of three essential steps. In this section, we present specific approaches for realizing each of the three steps for the purpose of forest cover estimation in the remote sensing domain.

For the problem of forest cover estimation, the response variable, $y_{l,t}$, is the amount of forest cover (`FC`) at a location l in year t, where forest cover denotes the proportion of pixel area covered by forests at a given location in a year ($y_{l,t} \in [0, 1]$). The explanatory variable, $\boldsymbol{x_{l,t}}$, consists of land surface temperature (`LST`) observations at a location l in year t.

Due to the presence of multiple land cover types, different regions on land show different relationships between `LST` and `FC`, leading to the presence of data heterogeneity. Since information about land cover types is not known explicitly, we are tasked at learning the partitioning of locations into homogeneous regions (whose locations share a common relationship between `LST` and `FC`). To achieve this, we look at the temporal behavior of locations in remote sensing datasets over the first few years, represented as a time-series at every location. With the assumption that locations that behave similarly in time (having similar time-series characteristics) belong to the same data partition, and thus share similar relationships between `LST` and `FC`, we can extract the data heterogeneity due to the presence of land cover types. We use normalized difference vegetation index (`NDVI`) time-series during the first few years as our structural variable, since `NDVI` has been shown to contain discriminatory information about land cover types in a recent study [28].

Partitioning the data

We employ unsupervised clustering approaches on $\mathcal{Z}$ (`NDVI`) for partitioning locations into homogeneous groups, each belonging to a different land cover type. By clustering the `NDVI` time-series during the first few years ($\boldsymbol{z_l}$), we are able to group locations that show similar trend in the `NDVI` time-series, which is indicative of their belonging to the

same land cover type. The choice of the clustering method would be more evident in the subsequent discussion in section 3.3.2 on learning the structure among data partitions.

Learning structure among data partitions

There exists multiple techniques for learning the structure among data partitions which have been obtained by clustering `NDVI` time series ($\boldsymbol{z_l}$). If the partitions have been discovered using a partitional clustering approach, such as the k-means algorithm, we can use the similarity between cluster representatives (centroids) for learning the structure among the partitions as a weighted complete graph among the m partitions. As an alternative approach, relationships between data partitions can be learned by employing hierarchical clustering techniques such as the bisecting k-means algorithm [29], and using the parent-child associations obtained in the clustering process as the structure among partitions. The presence of aggregated groups discovered by bisecting k-means is intuitive for our target application, since land cover types exhibit a hierarchical structure among themselves, e.g. broadleaf and needleleaf forests can be grouped into evergreen forests, which can be grouped with deciduous forests to form dense forests. It should be noted that the aggregated groups discovered as internal nodes act as dummy clusters that induce a structure among the leaf clusters. However, the final partitioning of the data is obtained only using the leaf nodes.

Using structure in predictive learning

Since the values of $y_{l,t}$ vary between $[0, 1]$, we consider logistic regression as our preferred regression algorithm. Logistic regression can be viewed as a generalized linear model, which uses the logit link function between the expected values of the response variable, $\boldsymbol{\mu_k}$, and the linear predictors, $\boldsymbol{\eta_k}$, given by:

$$\boldsymbol{\eta_k} = \boldsymbol{X_k}\boldsymbol{\beta_k}, \quad \text{and} \quad \boldsymbol{\mu_k} = \frac{1}{1 + \exp(-\boldsymbol{\eta_k})} \tag{3.8}$$

The structure among the clusters can be represented as a graph and used for regularizing our learning in lines of equation 3.7. Furthermore, due to the presence of aggregated groups which do not directly take part in the partitioning process, we discount the log-likelihood of observations at an internal node at height i by w_i, where $w_i < w_j$ for $i > j$

(nodes with higher heights have lower weights). Let

$$\boldsymbol{W} = Diag(w_{h_1}\boldsymbol{e_{n_1}},\ w_{h_2}\boldsymbol{e_{n_2}},\ \ldots,\ w_{h_m}\boldsymbol{e_{n_m}}) \tag{3.9}$$

be a diagonal matrix of size $n \times n$, where h_i is the height of node i, and $\boldsymbol{e_{n_i}}$ is a vector of ones of length n_i. Minimizing the negative log-likelihood of β using the logit link function along with introducing a regularization term in the objective function leads to the following optimization problem:

$$\begin{aligned}\min_{\boldsymbol{\beta}} E(\boldsymbol{\beta}) = -\boldsymbol{W}\boldsymbol{Y}^T \log(\boldsymbol{\mu}) - \boldsymbol{W}(\boldsymbol{e_n} - \boldsymbol{Y})^T \log(1 - \boldsymbol{\mu}) \\ +\ \lambda\boldsymbol{\beta}^T\tilde{\boldsymbol{L}}\boldsymbol{\beta}\end{aligned} \tag{3.10}$$

where, $\boldsymbol{\mu} = (\boldsymbol{\mu_1^T} \ldots \boldsymbol{\mu_m^T})^{\mathbf{T}}$, and $\mathbf{e_n}$ is a vector of ones of length n. Taking the first and second derivatives of $E(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, we get

$$\nabla E(\boldsymbol{\beta}) = \boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\mu} - \boldsymbol{Y}) + 2\lambda\tilde{\boldsymbol{L}}\boldsymbol{\beta} \tag{3.11}$$

$$\nabla^2 E(\boldsymbol{\beta}) = \boldsymbol{X}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{X} + 2\lambda\tilde{\boldsymbol{L}} \tag{3.12}$$

where, $\boldsymbol{R} = Diag(\mu(1 - \mu))$ is a diagonal matrix of size $n \times n$. We can then use the values of $\nabla E(\beta)$ and $\nabla^2(\beta)$ in the Newton-Raphson method to get the following update equation for β:

$$\boldsymbol{\beta_{t+1}} = \boldsymbol{\beta_t} - \boldsymbol{D}^{-1}(\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{\mu} - \boldsymbol{Y}) + 2\lambda\tilde{\boldsymbol{L}}\boldsymbol{\beta_t}) \tag{3.13}$$

where $D = \boldsymbol{X}^T\boldsymbol{R}\boldsymbol{W}\boldsymbol{X} + 2\lambda\tilde{\boldsymbol{L}}$ is an $md \times md$ matrix whose inverse has to be computed at each iteration of the Newton-Raphson method. We start with an initial choice of $\boldsymbol{\beta_0}$ as the global β learned by running a single logistic regression over the entire data. We stop iterating when the difference in $\boldsymbol{\beta_{t+1}}$ and $\boldsymbol{\beta_t}$ starts diminishing and goes below a certain tolerance value (10^{-3}), which indicates that the learning has converged to the optimum solution. After learning β, we use the $\boldsymbol{\beta_i}$ at a leaf node for testing over unseen data instances that belong to partition i (leaf node).

3.4 Datasets

Both LST and NDVI are obtained from the MODIS instrument onboard NASA's Terra and Aqua satellitesThe datasets are gridded at a spatial resolution of 0.05° on the geographic climate modeling grid (CMG), and are available at a monthly temporal resolution starting from the year 2000. We provide a description of each of the datasets below:

3.4.1 Land Surface Temperature (LST)

LST is derived from thermal infrared bands and measures the land surface temperature during the day as well as the night. We only consider cloud-free observations of LST for evaluation. Using a similar treatment of LST as proposed in [30], we consider the mean difference between LST Day and LST Night during the months corresponding to the dry season at a location in a year as the explanatory variable.

3.4.2 Normalized Difference Vegetation Index (NDVI)

NDVI provide a measure of greenness at a location which is indicative of the health of the biomass at that location. We consider the monthly NDVI time-series at a location, l, over a period of five years (2000 to 2004) as our structural variable, $\mathbf{z_l}$. The choice of NDVI for discriminating different land cover types from each other has been justified in a previous work on forest cover estimation [28].

3.4.3 Forest Cover Dataset (PRODES)

To obtain supervised information about the forest cover at a given location in Brazil, we used information from the Program for the Estimation of Deforestation in the Brazilian Amazon (PRODES) [31], which provides an annual deforestation product for each state in the Brazilian Amazon, using the analysis of high-resolution Landsat Thematic Mapper (TM) images.

3.5 Evaluation Setup

3.5.1 Baseline Algorithms

We compare the performance of our approach with the following three baseline methods:

Global Model (GLOBAL)

This baseline method (proposed in [30]) relies on learning a single logistic regression over the entire data. Since the global model neglects the rich heterogeneity in remote sensing datasets due to the presence of multiple land cover types, it suffers from poor generalization performance, and suffers from under-fitting.

Unregularized Regression (UNREG)

Instead of learning a single global model of the relationship between $\mathbf{x}$ and y, this baseline method (proposed in [28]) independently learns a separate logistic regression model at each data partition discovered by clustering $\mathbf{z}$. This can be viewed as a special version of our proposed approach, where the value of the regularization parameter, λ, is equal to 0, indicating the absence of any regularization. This model suffers from high model complexity and in scenarios where the size of training data is small, it often experiences the phenomena of over-fitting, leading to poor generalization performance. Since the model is not able to perform any learning in clusters which have 0 training instances, we utilize the global model learned using the overall data at such clusters for making predictions.

Ridge Regression (RIDGE)

In order to minimize the structural risk (indicative of the complexity of the model) at a data partition independently of other partitions, we introduce the L_2-norm of β as a regularization term in the objective function of UNREG, an approach commonly used in statistics to handle multi-collinearity [32]. This can be viewed as a special case of the proposed approach where the graph consists of a completely disconnected set of nodes with no edges. This enforces complete independence among the learned model parameters at data partitions and thus is a weaker form of regularization as compared

to our proposed approach. In order to learn relationships at clusters with 0 training instances, we utilize the global model learned using the overall data.

3.5.2 Evaluation Metric

We consider prediction performance as the guiding theme for evaluating and comparing predictive learning models. Let $\{y_i\}_1^n$ denote the set of true observations for a response variable, and let $\{\widehat{y_i}\}_1^n$ be the set of predicted values of the response variable. The Coefficient of Determination (R^2), which measures the proportion of variability in the response variable explained by the regression model, can then be formally defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{3.14}$$

where $\overline{y} = \sum y_i/n$ denotes the mean. We use $(1 - R^2)$ as an evaluation metric for analyzing the performance of our approach in comparison with other baseline approaches, since the same evaluation metric has been used in existing approaches for forest cover estimation, such as [28]. A lower $(1 - R^2)$ value corresponds to a better goodness of fit of the model.

3.5.3 Experimental Design

We evaluate the performance of our proposed approach in comparison with the baseline approaches over the combined region of four states in Brazil. The names of the four states, along with their latitude and longitude boundaries, can be enlisted as: Mato Grosso ($7°$–$19°S$, $62°$–$50°W$), Pará ($3°N$–$10°S$, $59°$–$46°W$), Amapá ($5°N$–$2°S$, $55°$–$49°W$), and Roraima ($6°N$–$2°S$, $65°$–$58°W$). We consider 10 years of `LST` and `FC` data from 2000 to 2009 for the purpose of evaluation. The total number of locations in the combined region of these four states is 164,400, amounting to 1,644,000 distinct data instances. We randomly sample P number of data instances for training, $Q = 100$ number of data instances from the remaining data for validating meta-parameters, and the remainder of the data is used for testing. Each random sampling is repeated $N = 50$ times so as to obtain the mean and standard deviation statistics of the evaluation metric, $(1 - R^2)$.

3.6 Experimental Results

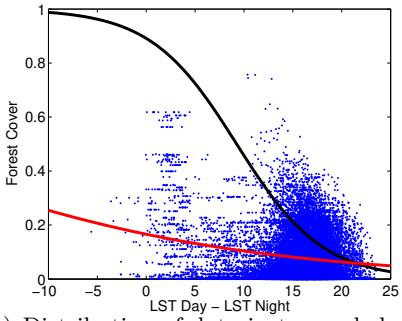
3.6.1 Visualization of clusters

We cluster the overall data into 15 partitions using the bisecting k-means algorithm, and specifically focus on two of the discovered clusters in figure 3.1. Figures 3.1(a) and 3.1(b) show the scatter plot of data instances belonging to cluster 1 and cluster 2, respectively, where the X axis corresponds to the explanatory variable, `LST`, and the Y axis corresponds to the response variable, `FC`. The black curves show the global logistic regression model learned over the entire data, whereas the red curves shows the logistic regression models learned at each cluster of the data independently. It can be seen that the global model overestimates Y in cluster 1, while it underestimates Y in cluster 2 as compared to the individual models at each cluster. This shows the importance of learning different regression models over different clusters of the data, thus accounting for data heterogeneity.

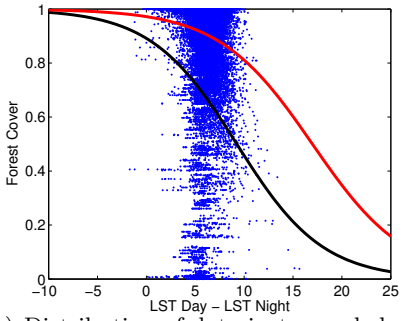
Figures 3.2(a) and 3.2(b) show the the centroid `NDVI` time series of locations belonging to cluster 1 and cluster 2, respectively. It can be seen that locations belonging to cluster 1 have a higher seasonal variance in `NDVI` and a lower annual `NDVI` mean than locations belonging to cluster 2. Furthermore, figures 3.1(c) and 3.1(d) show a sample of locations on land (marked by orange and yellow dots respectively) that belong to cluster 1 and cluster 2, respectively. It can be observed that cluster 1 corresponds to a land cover type that includes farms and barren land, while cluster 2 corresponds to densely vegetated forests. This shows that the discovered clusters correspond to land cover types and have real-world interpretability.

3.6.2 Varying the number of clusters

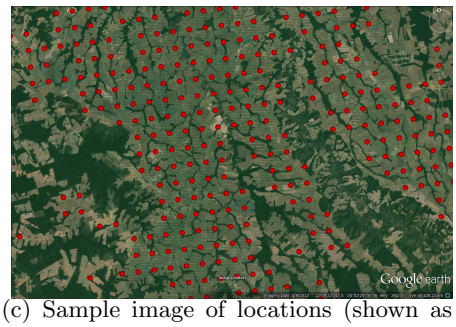
We randomly sample $P = 400$ observations for training and explore the behavior of testing errors for each competing algorithm as the number of clusters is increased from 1 to 500. Figure 3.3 shows the behavior of the mean and standard deviation of $(1 - R^2)$ values over varying number of clusters. It can be observed that the GLOBAL approach gives a constant mean $(1 - R^2)$ value of 0.70, since the GLOBAL approach is oblivious to any clustering procedure. On the other hand, UNREG, RIDGE, and the proposed approach shows an improvement in performance as the number of clusters, m,



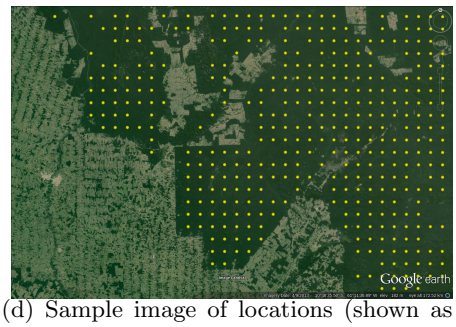
(a) Distribution of data instances belonging to cluster 1



(b) Distribution of data instances belonging to cluster 2



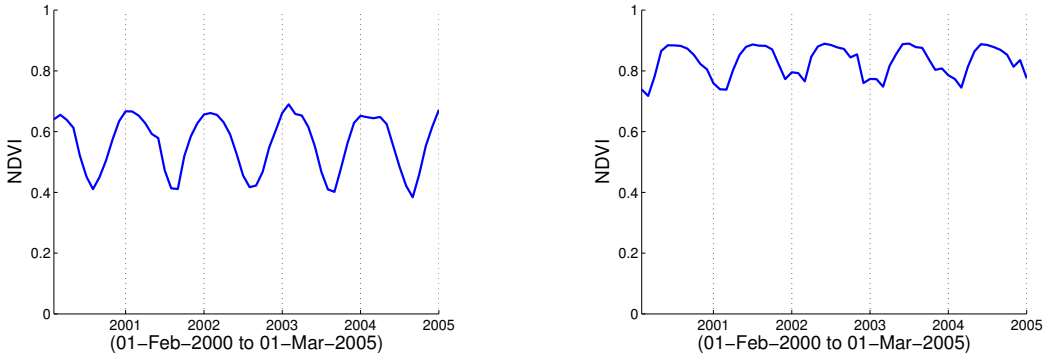
(c) Sample image of locations (shown as orange dots) belonging to cluster 1



(d) Sample image of locations (shown as yellow dots) belonging to cluster 2

Figure 3.1: Visual exploration of the partitions discovered by clustering `NDVI` time series. Figures 3.1(a) and 3.1(b) show scatter plot of data instances belonging to cluster 1 and 2, respectively. The X axis is `LST` Day - `LST` Night (explanatory variable) and the Y axis is `FC` (response variable). The black curves represents the global model, while the red curves represent individual models learned at each of the two clusters. Figures 3.1(c) and 3.1(d) show sample images of locations belonging to cluster 1 and 2, respectively.

is increased from 1 to 30. This indicates their potential in addressing data heterogeneity. However, increasing m from 30 to 500 increases the model complexity, making the learning prone to over-fitting. UNREG gradually starts over-fitting and reaches a $(1 - R^2)$ value close to that of GLOBAL at higher values of m. On the other hand, RIDGE is able to regularize the learning and maintains a constant $(1 - R^2)$ value as m is increased from 30 to 100. This shows the ability of RIDGE in avoiding over-fitting using limited training data. However, the performance of RIDGE eventually starts degrading as m increases from 50 to 500. Finally, the proposed approach consistently outperforms the three baseline methods for every value of m and is able to provide



(a) Centroid `NDVI` time series of cluster 1

(b) Centroid `NDVI` time series of cluster 2

Figure 3.2: `NDVI` time series of the centroids of cluster 1 (Figure 3.2(a)) and cluster 2 (Figure 3.1(c)).

a stronger regularization in the learning, indicated by lower $(1 - R^2)$ values even at $m = 500$. The minimum $(1 - R^2)$ value obtained by the proposed approach is 0.41, at $m = 50$.

3.6.3 Varying the size of training data

As we increase the number of observations available during training, we progress from an insufficient training data scenario to a sufficient training data scenario. In the presence of sufficient training data, algorithms with higher model complexity (such as UNREG) can be supported with lesser propensity of running into the problem of over-fitting. Figure 3.4 illustrates this effect by showing the results obtained by using $P = 1000$ observations for training. It can be observed that the $(1 - R^2)$ values for RIDGE and UNREG are relatively closer to the proposed approach, and keep on decreasing for all values of m from 1 to 50.

On the contrary, reducing the size of the training set reduces the amount of information available for addressing heterogeneity in the data, thus limiting the scope for reducing $(1 - R^2)$ values as compared to the GLOBAL approach. Furthermore, algorithms with higher model complexity would start over-fitting at lower values of m. Figure 3.5 demonstrates this phenomena using $P = 100$ observations for training. In

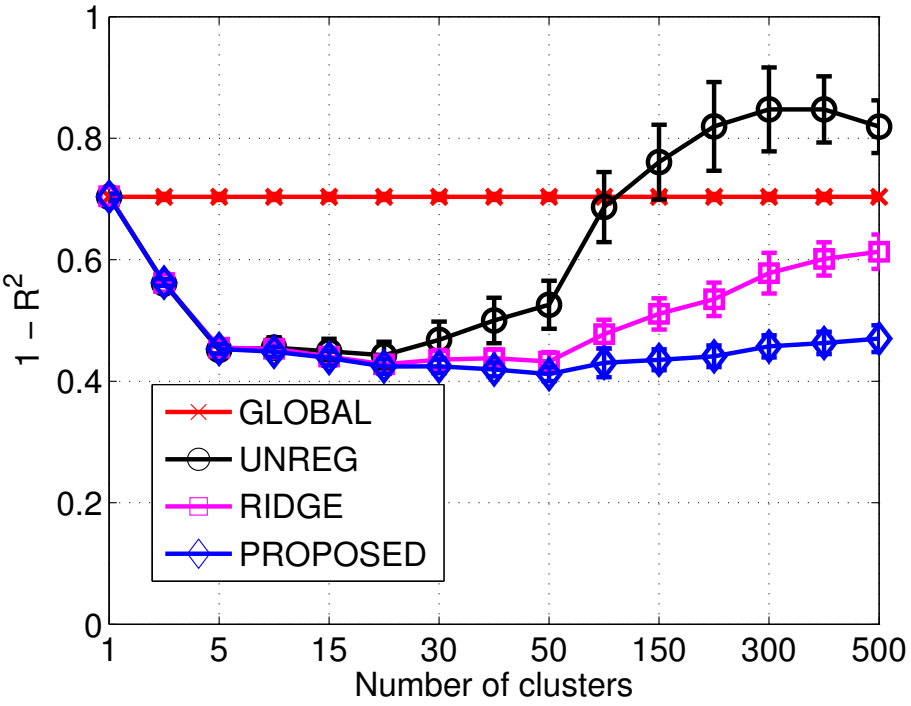


Figure 3.3: Errorbar plots of $(1 - R^2)$ at $P = 400$, as the number of clusters is increased from 1 to 500.

this case, both UNREG and RIDGE start over-fitting at $m = 5$. Also, it can be observed that the performance of UNREG deteriorates as we increase m from 1 to 30, indicating the presence of over-fitting. However, as we increase m from 30 to 500, we start encountering clusters with 0 training instances, and since the UNREG approach is not able to perform any learning in such clusters, it starts using the GLOBAL model for making predictions at such clusters. Thus, it can be observed that the performance of UNREG starts approaching the GLOBAL results at $m = 500$. On the other hand, our proposed approach consistently outperforms the baseline approaches for each value of m, since it employs a strong structural regularization scheme. The lowest $(1 - R^2)$ values obtained by the proposed approach is 0.48 at $m = 5$. It can also be observed that the performance of the proposed approach does not drastically deteriorate on increasing the m as compared to the baseline approaches.

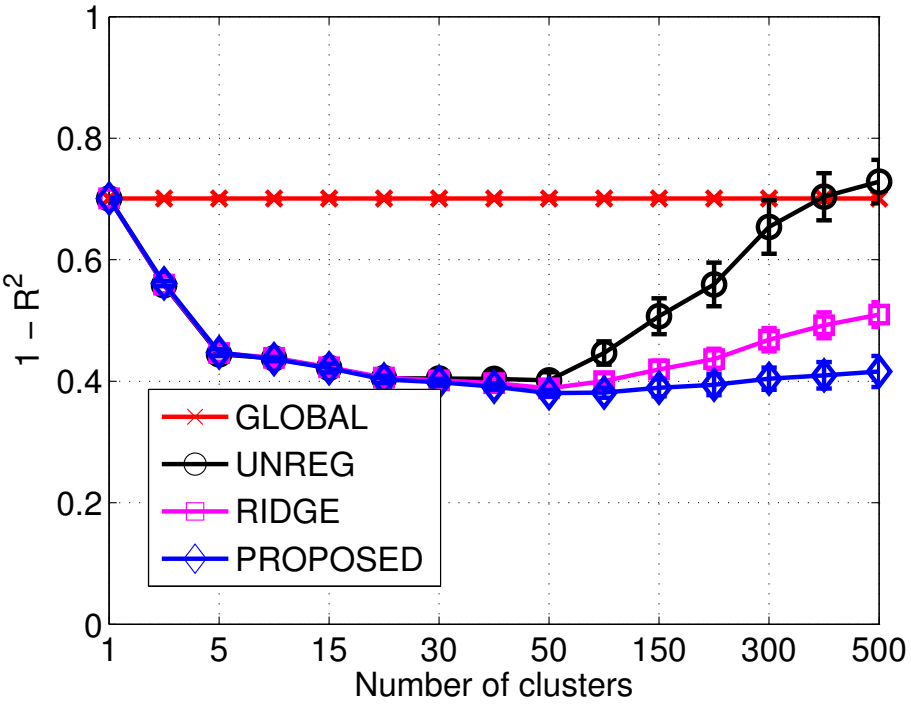


Figure 3.4: Errorbar plots of $(1-R^2)$ at $P = 1000$ as the number of clusters is increased from 1 to 500

3.6.4 Randomizing the structure in data

In order to assess the significance of using the structure among data partitions in regularizing our learning, we perform two randomization experiments, R-CLUSTER and R-EDGE, described as follows:

R-CLUSTER

Instead of assigning locations to clusters on the basis of their similarity in $\mathbf{z}$ (`NDVI` time series), we randomly assign each location to a cluster, while still preserving the structure among the clusters extracted using bisecting k-means. By randomizing the assignment of locations to clusters, we intend to construct artificial partitioning of locations which do not resemble homogeneous partitions of the data (corresponding to land cover types), but still are treated as unique entities (requiring the learning of separate

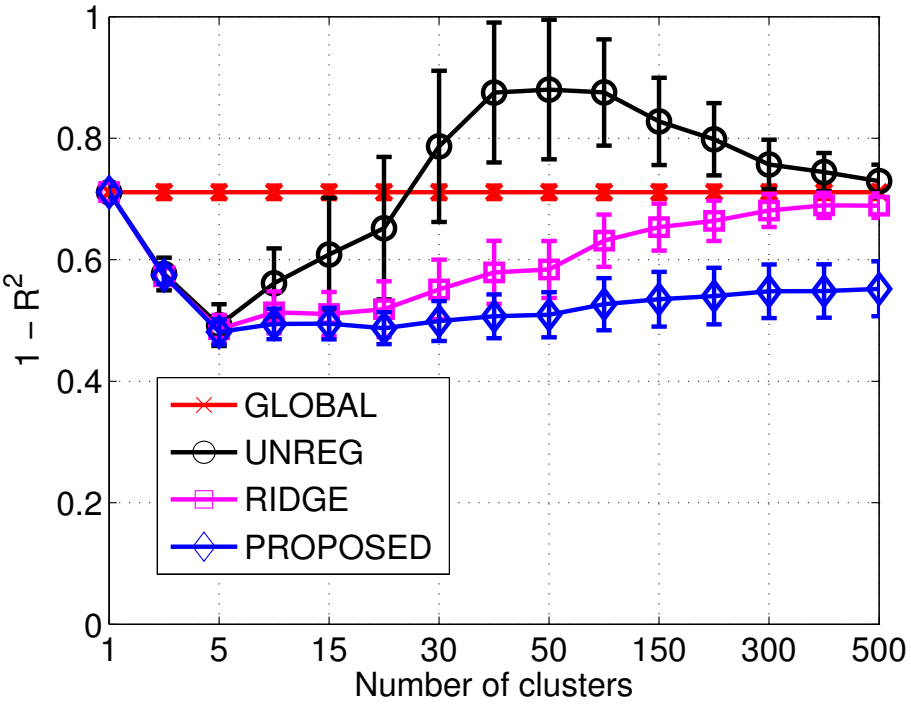


Figure 3.5: Errorbar plots of $(1 - R^2)$ at $P = 100$ as the number of clusters is increased from 1 to 500

model parameters) by the proposed approach. This would help quantitatively verify the interpretability of the discovered data partitions, obtained by clustering `NDVI` time series.

R-EDGE

We preserve the assignment of locations to clusters but randomize edges between the leaf nodes and their immediate parents in the structure among the clusters, leading to the creation of a randomized structure among clusters. The aim of this experiment is to test the significance of the structural relationships (extracted by bisecting k-means) among the data partitions, useful in regularizing our model learning and avoiding over-fitting.

Figure 3.6 summarizes the results of the randomization experiments in comparison with the results of the proposed approach and the GLOBAL model (repeated from section 3.6.2), using $P = 400$ observations for training.

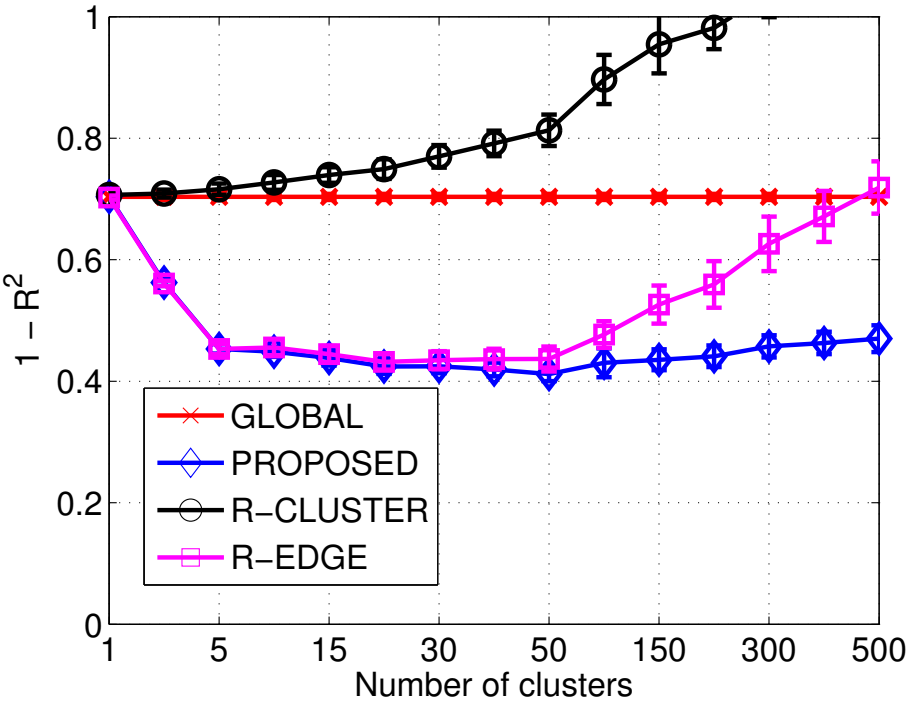


Figure 3.6: Errorbar plots of $(1 - R^2)$ at $P = 400$ after performing randomization experiments: R-CLUSTER and R-EDGE

Figure 3.6 shows that as m increases, R-CLUSTER starts showing higher $(1 - R^2)$ values than the GLOBAL model. This indicates that learning multiple model parameters (in an ensemble fashion) over random partitions of the data does not necessarily capture data heterogeneity. On the other hand, due to the increased model complexity of R-CLUSTER, the performance of R-CLUSTER starts degrading even in comparison with the GLOBAL approach.

It can be observed from figure 3.6 that R-EDGE shows similar $(1 - R^2)$ values as the proposed approach for m less than 50, after which it starts over-fitting. This can be explained by the fact that addressing heterogeneity alone is sufficient to improve the prediction performance for smaller values of m. However, as m is increased from 50 to 500, R-EDGE starts over-fitting in the presence of a randomized structure among data partitions. This indicates the existence of an underlying structure among the clusters, which is being extracted by the bisecting k-means and is being utilized in the

learning process for overcoming over-fitting. Furthermore, it can be observed that the performance of R-EDGE is very similar to that of the RIDGE model, shown in figure 3.3 and described in section 3.6.2. This correspondence can be explained by the fact that in the presence of a randomized structure among clusters, the regularization scheme effectively starts learning model parameters at each cluster independently, since sharing the model parameters in accordance with the randomized structure does not provide any gain in performance.

3.7 Conclusions and Future Work

There exists a rich population heterogeneity in a number of real-world datasets, that correspond to the presence of different relationships between explanatory and response variables over different partitions of the data. This can be conveniently exploited for improving prediction performance. In the absence of sufficient training data, addressing data heterogeneity is challenging, due to the increased model complexity in addressing heterogeneity. We proposed a framework for learning relationships in the presence of data heterogeneity and limited training data, which utilizes the structure among data partitions for regularizing the overall learning. We presented a generic formulation of our approach using generalized linear models, and further provided specific instantiations of the generic formulation for its application in estimating forest cover using remote sensing datasets. In particular, we utilized a graph-Laplacian based regularization scheme for sharing the learning of logistic regression models over data partitions (corresponding to land cover types), in the presence of limited training data. By performing a series of comparative experiments with the baseline approaches, we show that our proposed approach is both accurate and robust to over-fitting and the choice of parameters used to represent the heterogeneity in remote sensing datasets.

Future work would explore specific instantiations of each of the key steps of the proposed framework using state-of-the-art methods. In particular, we can explore extending our generic formulation using graph-based regularization to non-linear regression models. Since the guiding theme of our work is improving the prediction performance, we have omitted any discussion on the computational efficiency of our approach. Since the solution to our proposed approach requires matrix inversions at each step of an iterative

algorithm, we can explore techniques for improving the computational efficiency of the proposed approach. Further, we would be interested in learning the posterior estimates of the model parameters, thus additionally learning the confidence in our predictions of response variables. Finally, the proposed approach can be applied in other domains of study which suffer from insufficient training data, and exhibit similar forms of data heterogeneity.

Chapter 4

Learning Mode-specific Classification Ensembles

4.1 Introduction

In a number of real-world binary classification problems, there often exists a heterogeneity in the populations of instances belonging to the two classes, leading to a multi-modal distribution of both classes. As an example, different groups of locations on the Earth, belonging to either the water or the land class, show different characteristics in remote sensing datasets due to differences in geographies, topographies, climatic conditions, etc., resulting in a rich variety of land and water bodies at a global scale. As another example, different groups of human subjects, belonging to either the healthy or the diseased class, show different physiological symptoms to a certain disease, based on differences in their genetic information, living conditions, etc. To illustrate the presence of heterogeneity within the two classes, Figure 4.1 shows a toy example of a synthetic dataset where each of the two classes (positives and negatives) exhibit a multi-modal distribution in the feature space.

In the presence of a multi-modal distribution of instances within the two classes, one possibility is to learn a single non-linear classifier that discriminates between all positive and negative modes in the data. However, learning such a classifier is difficult especially in scenarios where certain pairs of positive and negative modes have higher degrees of overlap in the feature space as compared to others. The presence of such overlapping

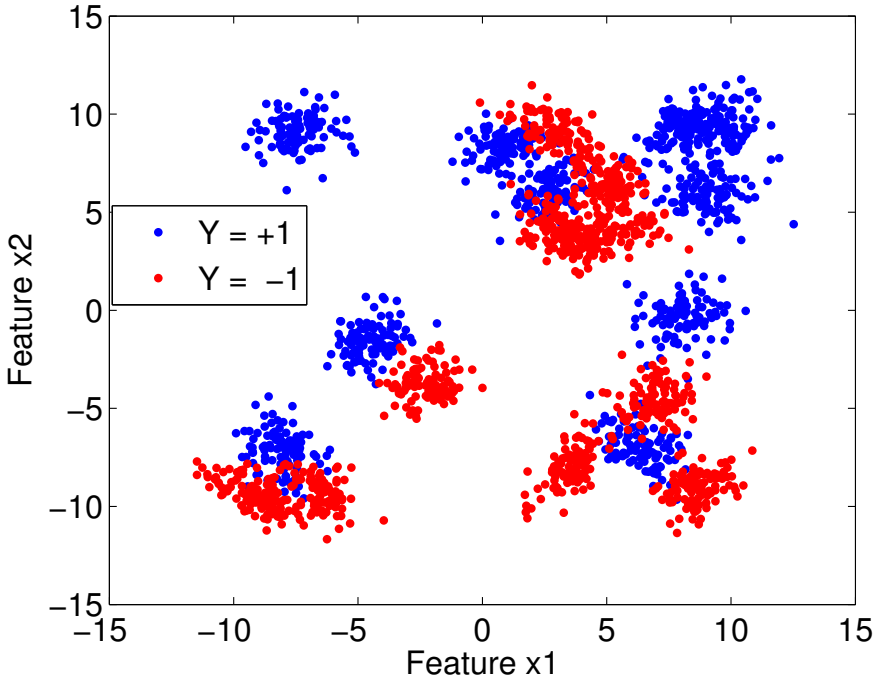


Figure 4.1: An illustrative example showing multi-modality in the distribution of the two classes.

pairs can impact the performance of the single classifier over other modes in the data that are reasonably separable in the feature space. In other words, since the properties of the desired classification boundary can vary differently across different modes of the positive and negative classes, learning a single classifier is difficult. Furthermore, the learning of a single classifier can be biased towards certain modes in the data that have been favorably represented in the training set, resulting in improper learning of the classifier over modes that have been under-represented during training. This motivates the need for decomposing the learning of a single classifier into the learning of different classifiers for different groups of positive and negative modes.

Recently, a number of machine learning techniques have been introduced for addressing various forms of heterogeneity in the data, e.g. heterogeneity among the instances (multi-instance learning) or heterogeneity among the views (multi-view learning). However, none of the existing heterogeneous machine learning approaches are suitable for dealing with multi-modality within the classes. In fact, techniques that appear closest to

handling multi-modality within the classes include multi-class classification techniques, since the problem of learning classifiers for different groups of positive and negative modes can be translated as a multi-class classification problem, where every positive or negative mode corresponds to a different sub-class. As an example, clustering-based techniques for decomposing the two classes into sub-classes, and then employing multi-class classifiers for discriminating between the different sub-classes has been explored in [33, 34]. Ensemble learning approaches for multi-class classification include the error correcting output coding (ECOC) approach presented in [35], which learns a different classifier to discriminate between different subsets of class labels. ECOC has been shown to provide improvements in both the bias as well as the variance of a base classifier [36], and a number of variants of ECOC have been proposed in the literature [37, 38]. Another ensemble learning approach for multi-class classification includes the pair-wise classifiers [39], which learns a classifier for every pair of class labels and has been shown to provide comparable performance as ECOC. A unified analysis of ECOC and the pair-wise classifiers was presented in [40].

However, existing ensemble learning methods for multi-class classification would ignore the bipartite nature of the sub-classes when used for a binary classification problem where each class constitutes of multiple sub-classes. Learning a classifier that discriminates between different sub-classes belonging to the same class is irrelevant for a binary classification problem, and the presence of such irrelevant classifiers can degrade the performance of the ensemble of classifiers. This motivates the need for devising ensemble learning methods that can take into account the bipartite nature of the positive and negative modes, which is unique to binary classification problems with multi-modality within the classes. The resulting classifier ensemble can also have the advantage of discarding classifiers for pairs of positive and negative modes that have a high degree of overlap in the feature space, differentiating it from existing ensemble learning methods for multi-class classification.

It should be noted that existing ensemble learning methods for binary classification use random partitions of the input space for learning ensemble classifiers [41, 42] and do not take into account the multi-modal structure within the two classes. As an example, bagging makes use of bootstrap samples of training instances for learning every classifier,

as opposed to performing a stratified sampling of training instances using their multi-modal structure. In contrast, we are interested in using the multi-modal structure of the two classes, as opposed to random samples, for learning ensemble classifiers. This would help in ensuring adequate representation of every mode in the learning of the classifier ensemble, along with maintaining diversity among the classifiers. Additionally, by learning classifiers for different subsets of positive and negative cluster labels, we attempt at capturing the local properties of the desired classifier in different regions of the feature space, in accordance with the multi-modality of the data.

In this thesis, we present a generic ensemble learning framework for binary classification with multi-modality within the classes. We compare the performance of the proposed methods with baseline approaches on a synthetically generated dataset and a real-world application of global lake monitoring using remote sensing datasets. We are able to demonstrate that the proposed approaches are able to provide significant improvements in classification performance as compared to learning a single classifier or using traditional ensemble learning techniques, over a broad range of base classification algorithms. The remainder of the chapter is organized as follows. Section 4.2 discusses related work that is relevant to this chapter. 4.3 describes the proposed mode-specific ensemble learning approach. Section 4.4 presents experimental results on a global surface water monitoring data set. Section 4.5 provides a discussion of the results. Section 4.6 includes concluding remarks and discusses directions for future work.

4.2 Related Work

The presence of heterogeneity within the two classes leads to differences in the characteristics of instances in different regions of the feature space. This requires the learning of classification models that can adapt themselves in different regions of the feature space, in lieu of the multi-modal distribution of the two classes. Traditional classification approaches that exhibit this property by learning different classification models in different sub-regions of the feature space include k-nearest neighbor (KNN) based classifiers, decision trees, and rule-based methods. However, KNN based approaches are susceptible to the fallibility of distance functions, especially in the presence of a large number of attributes. On the other hand, decision trees and rule-based techniques

can exhibit high model complexity and thus are prone to over-fitting. This limits the usability of existing classification approaches in scenarios that involve high degree of heterogeneity within each of the two classes.

There exists a rich body of literature on ensemble learning methods for binary classification problems [41–43]. The underlying principle of ensemble classifiers is to use a diverse set of weak learners, such that their aggregate response is closer to the true response than any of the individual responses of the weak learners. Ensemble learning methods have been shown to provide promising improvements in classification performance over a broad range of base classifiers. Popular techniques for ensemble learning include bagging [44], boosting [43], and random forests [45].

Ensemble learning approaches for multi-class classification problems, include the Error correcting output coding (ECOC) approach presented in [35], which learns a different classifier to discriminate between different subsets of class labels. ECOC has been shown to provide improvements in both the bias as well as the variance of a base classifier [36], and a number of variants of ECOC have been proposed in the literature [37, 38]. As an alternate ensemble learning approach for multi-class classification, the pair-wise classifiers [39] learns a classifier for every pair of class labels, which has been shown to provide comparable performance with ECOC. A unified analysis of ECOC and the pair-wise classifiers was presented in [40].

Supervised learning approaches that have made use of unsupervised techniques for learning the structure in the training data have been explored in [33, 34, 46, 47]. These methods involve the use of clustering-based techniques for decomposing the two classes into sub-classes, and then employ multi-class classifiers to discriminate between the different sub-classes.

4.3 Approach

The proposed ensemble learning framework comprises of the following three components: (i) extracting multi-modal structure within the two classes, (ii) constructing an ensemble of binary classifiers using the learned multi-modal structure in the data, and (iii) combining the responses from ensemble classifiers in order to assign binary labels to test instances. We discuss each of the three components in detail in the following

subsections:

4.3.1 Learning the Multi-modal Structure

Since the information about the multi-modal structure of the two classes is not explicitly known, it is important to learn the multi-modal distribution of the two classes from the training dataset. This can be achieved by clustering the training instances belonging to each of the two classes separately, as proposed in [33]. This can be formally described as follows.

Let the training dataset be represented as $\mathcal{D} = \{(\mathbf{x_i}, y_i)\}_1^n$, where $\mathbf{x_i} \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \{-1, +1\}$ is the binary response label. Let this training dataset constitute of n_P positive instances, $\mathcal{X}_\mathcal{P} = \{\mathbf{x_i} \mid y_i = +1\}_1^{n_P}$, and n_N negative instances, $\mathcal{X}_\mathcal{N} = \{\mathbf{x_i} \mid y_i = -1\}_1^{n_N}$. Using a suitable clustering strategy, we can cluster $\mathcal{X}_\mathcal{P}$ into k_P clusters such that the cluster label of a positive training instance can be given as $c_i \in \{P_1, ..., P_{k_P}\}$. Similarly, we can cluster $\mathcal{X}_\mathcal{N}$ into k_N clusters such that the cluster label of a negative training instance can be given as $c_i \in \{N_1, ..., N_{k_N}\}$.

It must be noted that the choice of the clustering technique and the number of clusters used for representing the multi-modality is dependent on the target application. We thus provide a generic framework for ensemble learning that is not tied to a specific clustering strategy but instead can be used in conjunction with any reasonable clustering strategy that captures the multi-modality within the classes. We used Gaussian mixture model (GMM) clustering as the preferred clustering strategy in this chapter.

4.3.2 Constructing Classifier Ensemble

In order to learn classifiers that discriminate between different subsets of positive and negative modes in the data, we need to selectively sample training instances that belong to the cluster labels being considered by a classifier. We present a generic framework for ensemble construction using a similar analysis presented in [40] on constructing ensembles for multi-class classification. Our objective for ensemble construction can be formulated as designing a coding matrix, $\mathbf{M} \in \{-1, 0, +1\}^{(k_P+k_N)\times m}$, which encapsulates information about the subsets of cluster labels participating in the learning of every classifier. The rows of $\mathbf{M}$ correspond to the positive and negative cluster labels,

the columns correspond to the m classifiers, and the value at $\mathbf{M}(i, j)$ denotes whether the i^{th} cluster label will be used for learning the j^{th} classifier as either the positive or the negative class. This can be formally described as follows.

Let the assignment of every positive and negative cluster label, c_i, to a row in the coding matrix, $\mathbf{M}$, be represented as $A(c_i)$, where $A(c_i) = i$, if $c_i = P_i$ and $A(c_i) = i+k_P$, if $c_i = N_i$. $A(c_i)$ thus maps the positive cluster labels to the first k_P rows of $\mathbf{M}$ and the negative cluster labels to the last k_N rows of $\mathbf{M}$. In order to learn the j^{th} classifier, f_j, we use training instances from cluster label, c_i, only if $\mathbf{M}(A(c_i), j) \neq 0$. In particular, we train f_j using a subset of training instances, $\mathcal{D}_j$, where,

$$\mathcal{D}_j = \{(\mathbf{x_i}, t_i) \mid t_i = \mathbf{M}(A(c_i), j) \text{ and } t_i \neq 0\}.$$

The j^{th} column of the coding matrix, $\mathbf{M}(:, j)$, thus helps in assigning a binary label, $t_i \in \{-1, +1\}$, to every instance belonging to a cluster label that either has a $+1$ or a -1 at its corresponding row position in $\mathbf{M}(:, j)$. f_j can then be learned using $\mathcal{D}_j$ given a base classification algorithm. We can further compute the accuracy of f_j on $\mathcal{D}_j$, denoted by Acc_j, which can be used for weighting the classifiers while combining their responses during testing as described in Section 4.3.3.

Depending on the choice of the coding matrix, $\mathbf{M}$, different strategies for ensemble construction can be developed. We present two promising coding strategies for binary classification with multi-modal data, which have their roots in multi-class classification techniques and are able to incorporate the bipartite nature of the cluster labels.

Bipartite Error Correcting Output Coding (BECOC)

Similar in essence to the error correcting output coding (ECOC) techniques used for multi-class classification problems, we propose Bipartite ECOC (BECOC) approach that exploits the bipartite nature of the cluster labels. The objective of BECOC is to design a coding matrix, $\mathbf{M}$, such that:

$$\mathbf{M}(i, j) \in \begin{cases} \{0, +1\}, & \text{if } i \leq k_P. \\ \{-1, 0\}, & \text{if } i > k_P. \end{cases}$$

This ensures that instances belonging to positive cluster labels are treated as positive

instances, while instances belonging to negative cluster labels are treated as negative instances in the learning of every classifier. Hence, no classifier discriminates between cluster labels belonging to the same class. Furthermore, it is desirable for $\mathbf{M}$ to satisfy the following two properties of ECOC for maximum error-correcting properties:

- *Column Separation* The columns of $\mathbf{M}$ should be different from each other. This is important for ensuring sufficient diversity and limited redundancy among the classifiers. It can be measured as the maximum Hamming distance between any two columns in $\mathbf{M}$.

- *Row Separation* The rows of $\mathbf{M}$ for the positive cluster labels should be different from the rows of $\mathbf{M}$ for the negative cluster labels. This is required for ensuring effective error-correcting properties of the classifier ensemble, which will be discussed in more detail in Section 4.3.3. It can be measured as the maximum Hamming distance between rows corresponding to positive and negative cluster labels in $\mathbf{M}$.

Devising an optimal coding matrix for ECOC with maximum row and column separation is an NP-complete problem, and choosing the number of columns of $\mathbf{M}$ is an open problem [48, 49]. Using the suggestions presented in [40], we chose an $\mathbf{M}$ that provided the maximum row and column separation out of 1000 randomly generated coding matrices. Furthermore, we used the suggested choice of m to be $\lceil 15 \log_2(k_P + k_N) \rceil$ in all our implementations.

Bipartite One-vs-One (BOVO)

BOVO involves the learning of a different classifier for every pair of positive and negative cluster labels. This corresponds to designing a coding matrix, $\mathbf{M}$, such that for every pair of positive and negative cluster labels, (c_i, c_j), there exists a column l in $\mathbf{M}$ such that:

$$\mathbf{M}(k,l) = \begin{cases} +1, & \text{if } k = A(c_i). \\ -1, & \text{if } k = A(c_j). \\ 0, & \text{otherwise.} \end{cases}$$

4.3.3 Combining Ensemble Responses

Having learned an ensemble of m classifiers, $\{f_1, ..., f_m\}$, we can apply the ensemble of classifiers at a test instance, $\mathbf{x}$, to obtain a vector of ensemble responses, $f(\mathbf{x}) = [f_1(\mathbf{x}), ..., f_m(\mathbf{x})]$. One possibility for combining the ensemble responses is to compute the aggregate ensemble response and use the sign of the aggregate response for predicting the class label, similar to existing ensemble learning methods for binary classification. However, in the presence of multi-modality within the two classes, every ensemble classifier specifically discriminates between different groups of positive and negative cluster labels, and hence is designed for predicting different groups of cluster labels as opposed to their associated class labels. In scenarios where the cluster labels within the same class are highly diverse in nature, taking an aggregate of the classifier responses would lose information about the cluster labels predicted by an ensemble classifier. This motivates the need for combining classifier responses without loosing information about the cluster labels predicted by every classifier, similar to the methods used for combining ensemble responses in multi-class classification literature.

For every cluster label c_i, the corresponding row of c_i in $\mathbf{M}$ represents the optimal vector of classifier responses for a test instance that belongs to c_i. We can thus associate the loss of a cluster label c_i at a test instance $\mathbf{x}$, termed as $\text{Loss}(c_i, \mathbf{x})$, in terms of the agreement between the corresponding row of c_i in $\mathbf{M}$ and the response vector, $f(\mathbf{x})$. This can be defined as follows:

$$\text{Loss}(c_i, \mathbf{x}) = \sum_{j=1}^{m} \alpha_j L(z_j),$$

$$\text{where } z_j = t_i f_j(\mathbf{x}), \quad \text{and } t_i = \mathbf{M}(A(c_i), j).$$

Here, z_j measures the disagreement between the response of the j^{th} classifier on $\mathbf{x}$,

Base Classification Algorithm	Loss Function, $L(z)$
Support Vector Machine	$\max\{1 - z, 0\}$
AdaBoost	e^{-z}
Bagging	$(1 - \text{sign}(z))/2$
Decision Trees	$(1 - \text{sign}(z))/2$
Random Forests	$(1 - \text{sign}(z))/2$

Table 4.1: Loss functions used for decoding

$f_j(\mathbf{x})$, and the class label, $t_i = \mathbf{M}(A(c_i), j)$, assigned to c_i while learning the j^{th} classifier. It can be observed that z_j is positive when the signs of both t_i and $f_j(\mathbf{x})$ agree, whereas z_j is negative when t_i and $f_j(\mathbf{x})$ disagree. Further, $L(z)$ is an appropriate loss function that penalizes disagreements between t_i and $f_j(\mathbf{x})$, depending on the choice of the base classification algorithm. Table 4.1 provides a list of the loss functions used for different choices of base classification algorithms presented in this chapter. Finally, α_j is the weight associated with each ensemble classifier computed as follows:

$$\alpha_j = \begin{cases} \text{Acc}_j, & \text{if } \text{Acc}_j > 0.5. \\ 0, & \text{otherwise.} \end{cases}$$

where Acc_j is the accuracy of f_j, computed over its training set, $\mathcal{D}_j$. Using α_j thus helps in discarding classifiers that have been trained poorly, possibly due to the presence of overlaps among the involved cluster labels in the feature space. We can then choose $\hat{c}_i$ as the cluster label which provides the minimum loss, $\hat{c}_i = \arg\min \text{Loss}(c_i, \mathbf{x})$. The predicted label, $\hat{y}$ at a test instance $\mathbf{x}$ is then given as

$$\hat{y} = \begin{cases} +1, & \text{if } \hat{c}_i \in \{P_1, ..., P_{k_P}\}. \\ -1, & \text{if } \hat{c}_i \in \{N_1, ..., N_{k_N}\}. \end{cases}$$

It can be observed that having sufficient separation among the rows of the coding matrix for cluster labels belonging to opposite classes ensures that the loss of the true cluster label at an instance is sufficiently smaller than the loss of the cluster labels belonging to the other class, even after incurring few errors in $f(\mathbf{x})$. Row separation thus helps in imparting robust error-correcting properties to BECOC, making them resilient to errors in the ensemble responses.

4.4 Experimental Results

We used support vector machines (SVMs) using linear kernel and decision trees as the base classifiers for the ensemble learning methods. We further considered AdaBoost, bagging, and random forests as base classification algorithms in order to compare the proposed ensemble learning methods with traditional ensemble learning techniques. The

trade-off parameter of SVM was chosen to be 0.5 in all our experiments. We considered pruned decision trees with maximum number of internal nodes equal to 30, in order to prevent over-fitting of decision trees. The sizes of the ensembles for AdaBoost, bagging, and random forests were chosen to be 50 each, while decision trees were used as base classifiers for AdaBoost and bagging. The number of positive and negative clusters were kept equal in all experiments ($k_P = k_N = k$). We used the classification error rate as the evaluation metric for comparing the performance of classification algorithms.

4.4.1 Results on Synthetic Datasets

We used the synthetic dataset shown in Figure 4.1, which is representative of real-world classification scenarios involving multi-modality within the classes. Each of the two classes comprised of instances generated in a 2-dimensional feature space from 10 bivariate Gaussian distributions, with varying means and variances. The prior probability of each of the 10 Gaussian distributions in each class was kept equal. We used 200 randomly sampled instances for training, and a separate set of $20,000$ instances for testing, where the random sampling procedure was repeated 10 times.

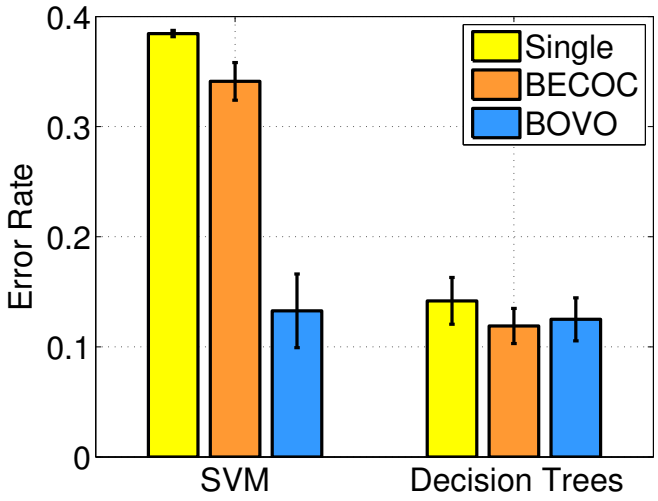


Figure 4.2: Comparison of ensemble learning methods on the synthetic dataset for different base classifiers, using $k = 10$ positive and negative clusters.

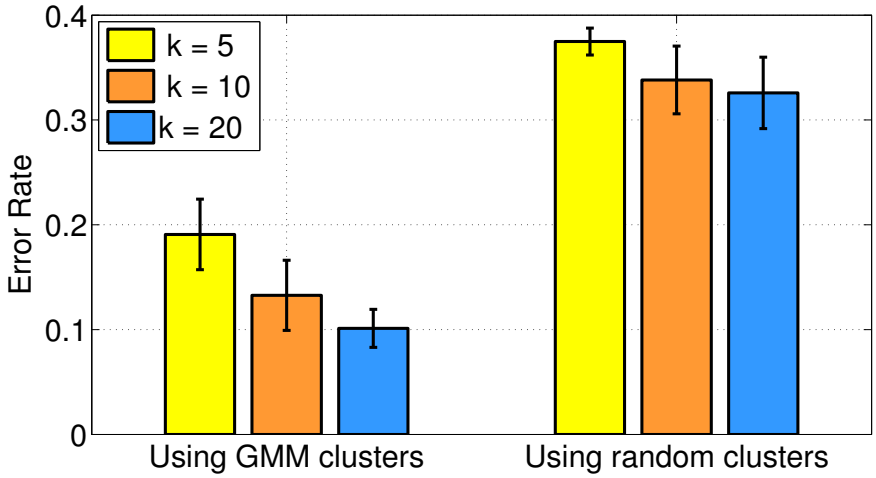


Figure 4.3: Varying clustering choices on the synthetic dataset, with SVM as the base classifier and BOVO as the ensemble learning method.

Figure 4.2 compares the performance of proposed ensemble learning methods, BECOC and BOVO, with single classifier for different choices of base classification algorithm: SVM and decision trees. The number of positive and negative clusters, k, was chosen to be 10. It can be observed that both BECOC and BOVO show better performance than learning a single classifier for both choices of the base classifier. This demonstrates the importance of using ensemble learning methods in the presence of multi-modality within the two classes, as opposed to learning a single classifier. Furthermore, BOVO shows better performance than BECOC on using SVM as the base classifier, while BECOC provides better performance than BOVO on using decision trees as the base classifier.

We next study the impact of varying the number of clusters used to represent the multi-modality within the two classes on the performance of an ensemble classifier. Figure 4.3 shows the performance of the BOVO classifier, learned using SVM as the base classifier, for different choices of the number of clusters, $k = 5, 10, 20$. It can be observed that the error rate of BOVO is higher for $k = 5$ than $k = 10$, which implies that the multi-modality in the synthetic data (generated using 10 Gaussian distributions) is not being fully explained by 5 clusters, resulting in an inferior learning of the ensemble classifiers. However, increasing the number of clusters from $k = 10$ to $k = 20$ further

leads to reduction in the error rate. This shows that over-clustering of the two classes does not significantly impact the performance of BOVO for $k = 20$.

Next, in order to assess the meaningfulness of the clustering step in the construction of the ensemble classifiers, we randomly assigned every training instance to either of the k randomly generated clusters, resulting in an artificial partitioning of the data into k random clusters. Figure 4.3 shows the performance of BOVO using random clusters instead of using clusters learned by GMM. It can be observed that the error rate of BOVO using $k = 5$ random clusters is very close to the error rate of a single base classifier, SVM. However, the error rate of BOVO starts reducing as the number of clusters is increased from $k = 5$ to $k = 20$. This can be attributed to the fact that learning an ensemble of classifiers where each classifier uses a random subset of the data for training (given a random clustering) is similar in essence to bagging. Furthermore, it can be observed from Figure 4.3 that using a meaningful clustering technique, such as the GMM clustering, is able to provide significantly lower error rates than using randomly assigned clusters (similar to bagging). This shows the strength of using information about the multi-modality within the two classes for ensemble construction, as opposed to using bootstrap samples of training instances.

4.4.2 Results on Global Lake Monitoring Dataset

We consider a real-world application of global lake monitoring using remote sensing datasets. Lakes are important natural resources that act as major sources of freshwater, which is essential for supporting a variety of human needs, such as drinking, agriculture, and industrial needs [50]. Monitoring the extent and growth of lakes at a global scale is thus important for effective water management. To this effect, remote-sensing datasets provide timely and cost-effective observational data of lakes at a global scale. We use the optical remote sensing dataset obtained via the MODerate-resolution Imaging Spectoradiometer (MODIS) instrument onboard NASA's Terra and Aqua satellites. This data product (MCD43A4) is publically available through the MODIS repository [51] at 500 meter resolution for every 8 days, starting from Feb 18, 2000. This dataset has seven reflectance bands, covering visible, infrared, and thermal parts of the electromagnetic spectrum, which can be used as features for discriminating between water and land.

Ground truth information about the extent of lakes was obtained via the Shuttle

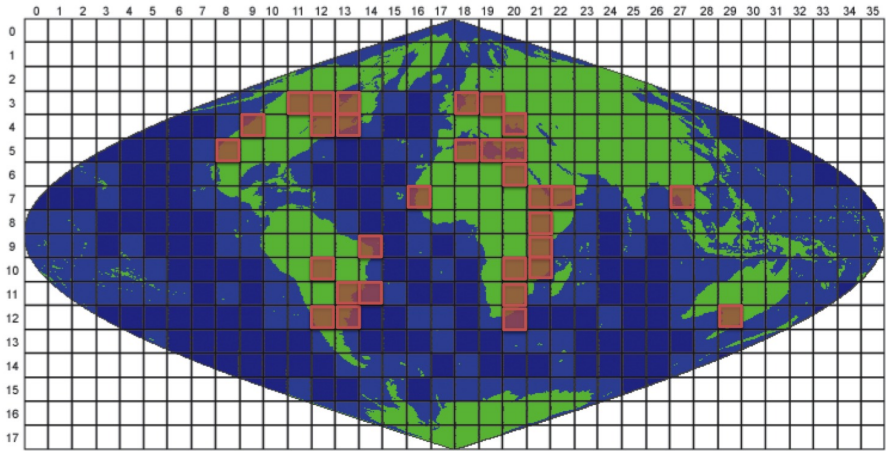


Figure 4.4: The 33 MODIS tiles (highlighted as red boxes) that were used for constructing the evaluation dataset.

Radar Topography Mission's (SRTM) Water Body Dataset (SWBD), which provides a mapping of all water bodies for a large fraction of the Earth (60^o S to 60^oN) for a short duration of 11 days around Feb 18, 2000 (the closest date at MODIS scale). The SWBD dataset, publically available through the MODIS repository as the MOD44W product, thus provides a label of land or water for every MODIS pixel at 500m for a single date, Feb 18, 2000.

We consider a global set of 180 lakes collected from 33 different MODIS tile divisions across the globe (highlighted in red in Figure 4.4) as our evaluation dataset. For each lake, we created a buffer region of 20 pixels at 500m resolution around the periphery of the water body, and used the buffer region as well as the interior of the water body to construct the evaluation dataset. After removing instances at the immediate boundaries of the water bodies for which the ground truth might not be accurate and ignoring instances with missing values, the evaluation set comprised of $\approx$ 2.6 million data instances, where every instance had an associated binary label of water (positive) or land (negative). This dataset approximately had 2.8 times more negatives than the positives. We randomly sampled 2000 positive instances and 2000 negatives instances for training, while the remainder of the evaluation data was used for testing. We repeated this balanced random sampling procedure 10 times. The number of clusters used to represent the multi-modality within the two classes was chosen as $k = 10$.

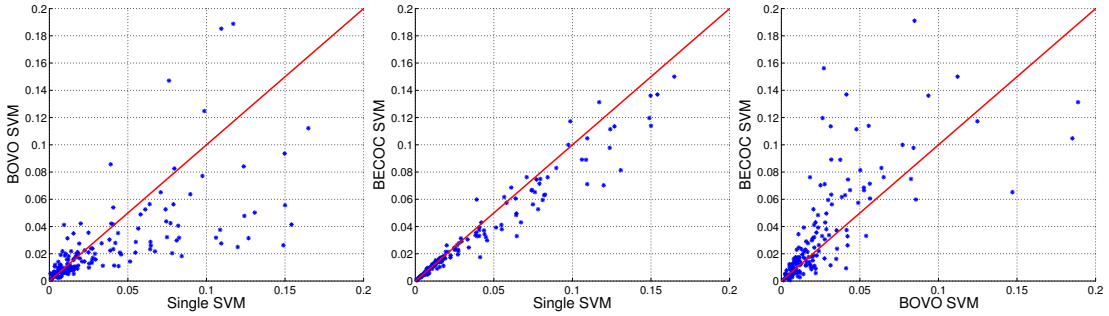


Figure 4.5: Scatter plots of mean error rates at 180 lakes using SVM as the base classifier.

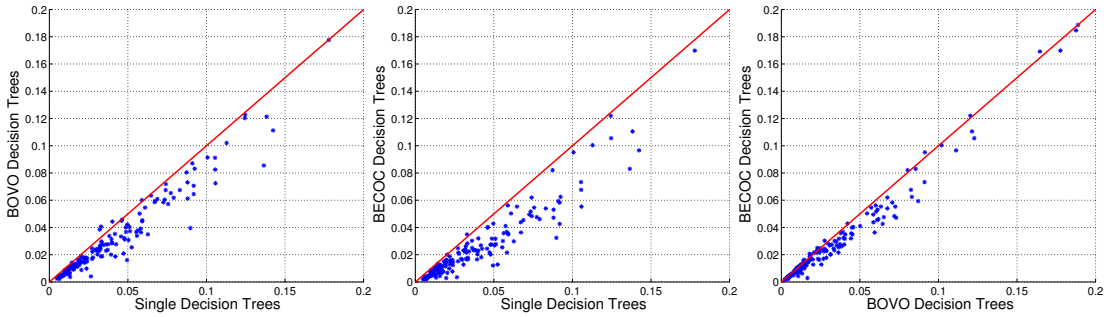
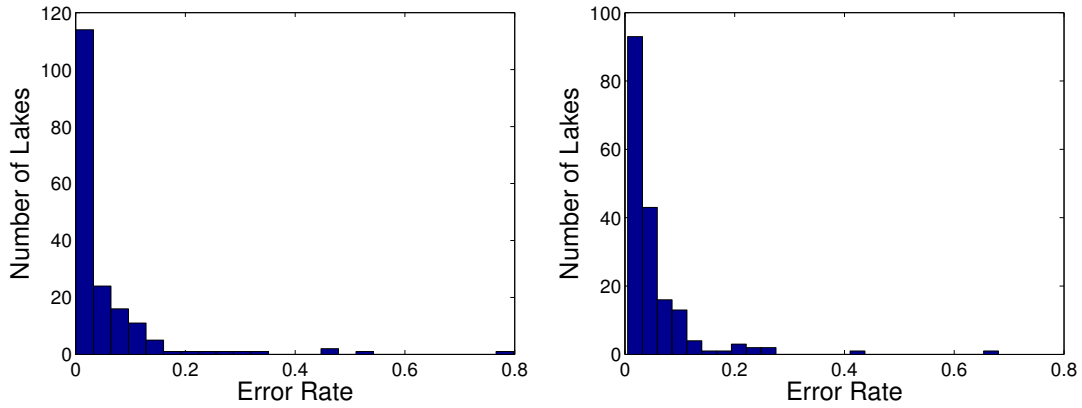


Figure 4.6: Scatter plots of mean error rates at 180 lakes using decision trees as the base classifier.

In order to compare the performance of classification algorithms at the level of individual lakes, we computed the test error rate of each algorithm over every lake individually. Figure 4.7 presents histograms of the mean error rates at every lake, averaged over 10 iterations, using SVM (Figure 4.7(a)) and decision trees (Figure 4.7(b)). It can be observed that a majority of lakes have error rates lower than 0.2 for both SVM and decision trees. We thus explore differences between classification algorithms over lakes with mean error rates lower than 0.2, using the scatter plots shown in Figure 4.5 and Figure 4.6, using SVM and decision trees as base classifiers respectively. Every point on a scatter plot involving algorithm i and algorithm j represents the mean error rate of algorithm i and algorithm j at a particular lake. The red line in each of the scatter plots shows the plot of $y = x$ for ease of comparison.

It can be observed that BECOC and BOVO show better performance than single



(a) Using Single SVM (b) Using Single Decision Tree

Figure 4.7: Histogram of mean error rates of 180 lakes, averaged over 10 iterations, using Single SVM and Single Decision Tree.

classifiers over a majority of lakes, for both SVMs and decision trees. Further, the improvements in performance of decision trees are smaller but more consistent than SVMs, owing to the non-linear nature of decision trees that makes it better suited for handling multi-modality within the classes. In order to assess the statistical significance of the lake-wise differences between classification algorithms, we computed the p-value of an Algorithm i showing lower mean error rates than Algorithm j over 180 lakes, by using one-tailed Wilcoxon signed rank tests. We denote this as the p-value of Algorithm i *over* Algorithm j, for different choices of (i, j) in Table 4.2, over a broad range of base classifiers. Differences that are significant with a p-value lower than 0.06 have been highlighted in bold.

BECOC can be seen to provide statistically significant improvements in the performance of single classifiers over all choices of base classifiers. Its ability to improve the performance of existing binary ensemble learning methods, such as AdaBoost, random forests, and bagging, highlights the importance of using the multi-modal structure in learning classifier ensembles. On the other hand, BOVO can be seen to provide better performance than the single classifier for SVM, decision trees, and AdaBoost, but shows poorer performance than the single classifier when used with random forests and bagging. Furthermore, the performance of BOVO is significantly better than BECOC for SVMs, but is worse than BECOC for decision trees, random forests, and bagging.

This highlights that BOVO shows poor performance when the base classifier is complex, while it is best suited for linear base classifiers. This phenomena has been discussed in detail in Section 4.5, using an illustrative example. In contrast, BECOC is able to provide robust improvements in the classification performance over a broad range of base classifiers.

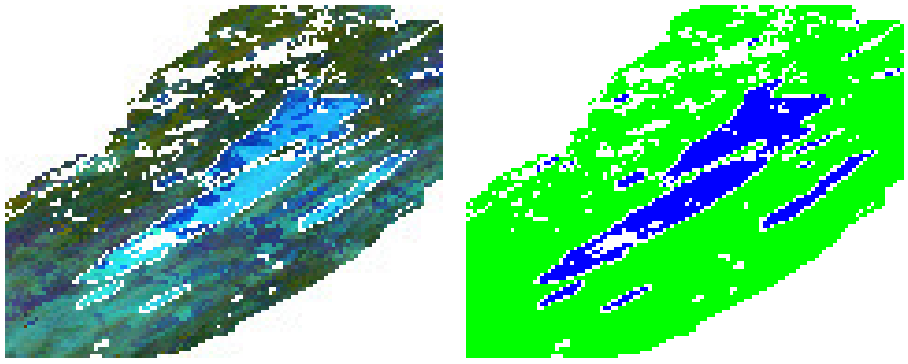
Base Classifier	BECOC *over* Single	BOVO *over* Single	Single *over* BECOC	Single *over* BOVO	BECOC *over* BOVO	BOVO *over* BECOC
SVM	$\mathbf{1.82 \times 10^{-8}}$	$\mathbf{4.34 \times 10^{-9}}$	1	1	1	$\mathbf{1.66 \times 10^{-7}}$
Decision Trees	$\mathbf{5.09 \times 10^{-29}}$	$\mathbf{3.88 \times 10^{-26}}$	1	1	$\mathbf{2.65 \times 10^{-23}}$	1
AdaBoost	$\mathbf{2.42 \times 10^{-19}}$	$\mathbf{7.87 \times 10^{-5}}$	1	1	0.19	0.81
Random Forests	$\mathbf{9.8 \times 10^{-4}}$	1	1	$\mathbf{1.43 \times 10^{-7}}$	$\mathbf{7.49 \times 10^{-14}}$	1
Bagging	**0.054**	1	0.95	$\mathbf{6.70 \times 10^{-8}}$	$\mathbf{1.61 \times 10^{-13}}$	1

Table 4.2: Table of p-values for Algorithm i showing lower mean error rates than Algorithm j over 180 lakes, represented as the p-value of Algorithm i *over* Algorithm j, for different choices of the base classifier.

4.5 Discussion of Results

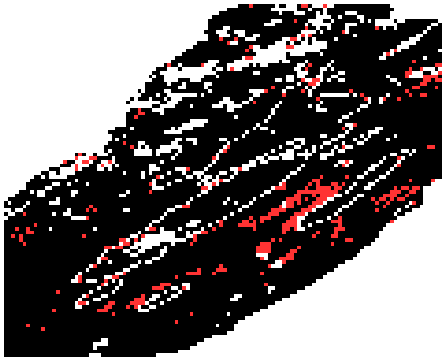
We present a discussion of the differences in classification algorithms using an illustrative set of lakes. Figure 4.8 compares the performance of BOVO and single classifier using SVM as the base classifier at Lake Lac La Loche in Saskatchewan, Canada. Figure 4.8(a) shows a false color composite (using the 7^{th}, 5^{th}, and 4^{th} bands, as red, green and blue colors respectively) of the test instances in the lake, while Figure 4.8(b) shows the ground truth at this lake, where blue and green pixels represent water and land classes respectively. The white pixels represent instances that were excluded from the test set. Figures 4.8(c) and 4.8(d) respectively show the errors of BOVO and single classifier as red pixels. It can be observed that some patches of land around the water body are covered by snow, which appear to be visually similar to water in the false color composite. This shows the presence of a variety in the land patches that leads to the poor performance of a single SVM classifier. However, BOVO is able to take into account the presence of multiple varieties of land and water bodies and is thus able to provide significant reduction in the error rate as compared to the single classifier.

Figure 4.9 compares the performance of BECOC and single classifier using decision trees as the base classifier at Walker Lake in Nevada, USA. It can be seen that the errors

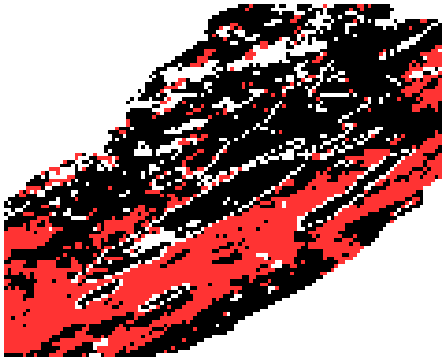


(a) False color composite

(b) Ground truth



(c) Errors of BOVO (red)

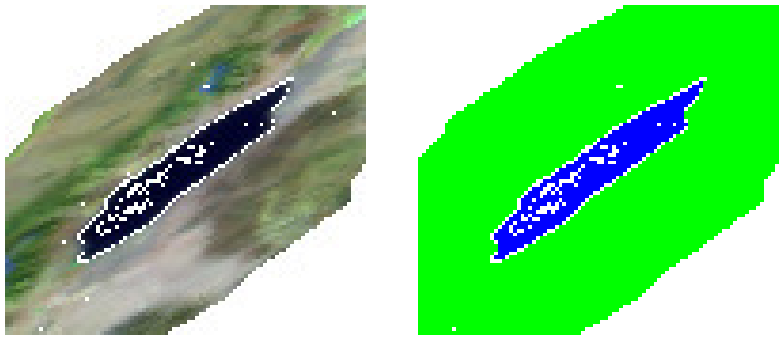


(d) Errors of Single (red)

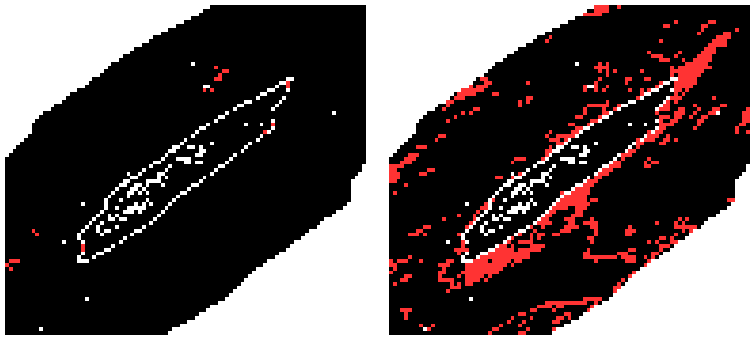
Figure 4.8: Comparing the performance of BOVO and Single at Lac La Loche Lake, Saskatchewan, Canada, using SVM as the base classifier. Error rate of BOVO = 0.05; Error rate of Single = 0.32.

of the single decision tree are randomly distributed in space, indicating over-fitting of decision trees. On the other hand, BECOC is able to provide a robust classification performance with significantly fewer errors than the single classifier.

Since the BECOC approach benefits from the error-correcting properties of ECOC, it is robust to the presence of noise or a small number of errors in the classification responses. On the other hand, BOVO is more susceptible to the presence of noise in the training data, especially when the base classifier is non-linear and complex. For example, in the presence of a cluster in the training data that comprises of noisy instances, BOVO would attempt to learn pair-wise classifiers to specifically discriminate the noisy cluster from the other classes, and thus will be prone to over-fitting when the base classifier



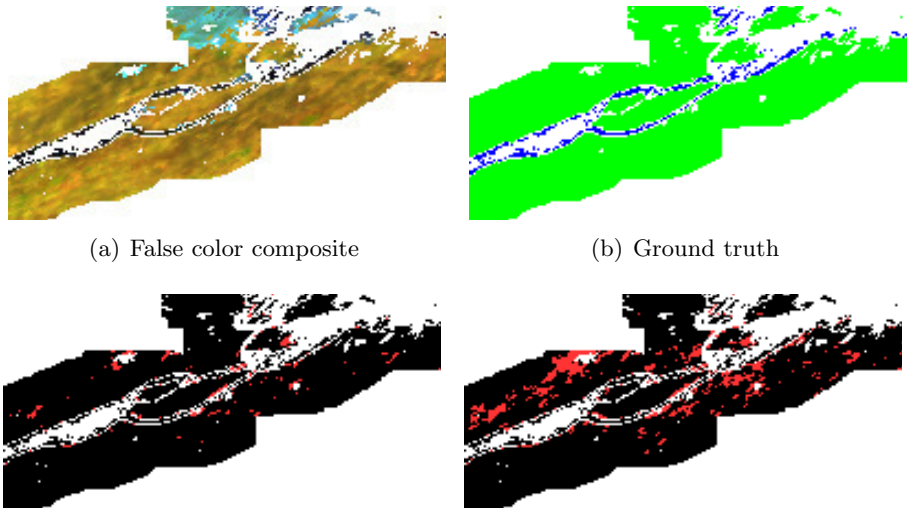
(a) False color composite (b) Ground truth



(c) Errors of BECOC (red) (d) Errors of Single (red)

Figure 4.9: Comparing the performance of BECOC and Single at Walker Lake, Nevada, using Decision Trees as the base classifier. Error rate of BECOC = 0.01; Error rate of Single = 0.05.

is complex. In contrast, the performance of BECOC is robust to the presence of a few noisy instances, since every classifier discriminates between a subset of positive and negative clusters. However, this results in a lower model capacity of BECOC in discriminating between arbitrary pairs of positive and negative modes. Hence, it is the trade-off between the model complexity and model capacity of BOVO and BECOC that determines their suitability for different choices of base classifiers. This can be illustrated by comparing the performance of BECOC and BOVO at Saint Lawrence River in Montreal, Canada, shown in Figure 4.10, using bagging as the base classifier. It can be seen that BOVO is making errors over a large patch of land that has a darker signature in the false color composite than other land patches. In the presence of noisy training instances with similar feature values labeled as water, it is quite likely



(a) False color composite (b) Ground truth

(c) Errors of BECOC (red) (d) Errors of BOVO (red)

Figure 4.10: Comparing the performance of BECOC and BOVO at Saint Lawrence River, Montreal, Canada, using Bagging as the base classifier. Error rate of BECOC = 0.03; Error rate of BOVO = 0.05.

for BOVO to learn pair-wise classifiers that discriminate between the cluster of noisy training instances and other land classes, leading to poor classification performance.

4.6 Conclusions and Future Work

We study the importance of using information about the multi-modality within the two classes in ensemble learning for binary classification. Inspired by the existing ensemble learning approaches for multi-class classification, we develop ensemble learning methods for binary classification that make use of the bipartite nature of the positive and negative modes in the data. Constructing classifier ensembles using information about the the multi-modal structure of the two classes, as opposed to using random samples, helps in ensuring sufficient diversity among the classifiers and adequate representation of the data modes in the learning of the classifier ensemble. We demonstrate the effectiveness of the ensemble learning methods presented in this chapter in comparison with learning a single classifier or using traditional ensemble learning techniques over a synthetic dataset and a real-world application involving global lake monitoring.

There are a number of aspects of the proposed ensemble learning methods that need further investigation, presenting ample opportunities for future research. It can be observed that BECOC shows better performance than BOVO when decision trees are used as the base classifier, while BOVO shows better performance than BECOC when SVM is used as the base classifier. The sensitivity of the ensemble classifiers on the choice of the base classifier and the presence of noise in the training set needs to be theoretically understood. The usability of the generic ensemble learning framework presented in this chapter needs to be explored with varying choices of clustering techniques, number of clusters, and base classifiers. Furthermore, applications of the proposed methods on other real-world datasets involving heterogeneity within the classes need to be explored.

Chapter 5

Adapting Predictions using Group-level Properties of Test Instances

5.1 Introduction

As discussed in the previous chapter, a number of binary classification problems commonly experience population heterogeneity within the two classes, which is characterized by the presence of multiple modes of each of the two classes in the feature space. Figure 5.1 shows a schematic illustration of a classification problem involving multiple modes of the positive and negative classes. In such situations, different pairs of positive and negative modes can show varying degrees of overlap in the feature space. This is represented in Figure 5.1 as edges with varying thickness, where the thickness of an edge reflects the degree of overlap between the pair of modes. Learning a single classifier that discriminates between all varieties of positive and negative modes is then challenging, especially in the presence of highly overlapping pairs of modes. We denote this phenomena as class confusion and the pair of modes participating in a class confusion as confusing modes in the remainder of this chapter.

We consider binary classification problems where the classification has to be performed over different test scenarios, and every test scenario involves only a subset of all

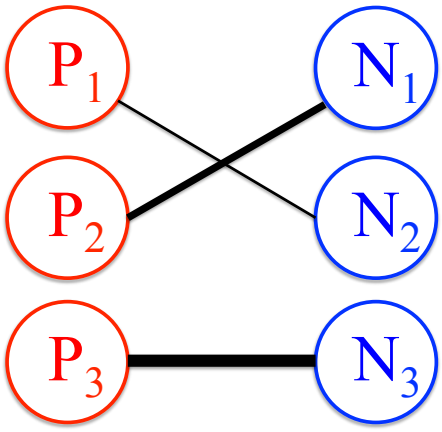


Figure 5.1: A schematic illustration of multi-modality within the classes, where each class comprises of three modes. Thickness of an edge shows the degree of overlap between the pair of modes.

the positive and negative modes in the data. As an illustrative example, in the context of classifying locations on the Earth as water or land, a test scenario would comprise of instances observed in the vicinity of the same water body and at the same time-step. In such a setting, different pairs of positive and negative modes may emerge or disappear in different test scenarios, and even though some modes may be participating in class confusion, the subset of modes appearing in a given test scenario can be considered to be locally separable among each other. This shows a promise in using information about the context of a test scenario for overcoming class confusion.

To illustrate the importance of using the local context of a test scenario in the learning of a classifier, consider the toy dataset shown in Figure 5.2. This dataset comprises of instances belonging to two classes where each class comprises of two distinct modes, shown as colored circles in Figure 5.2. It can be observed that modes P_1 and N_1 are easily separable in the feature space, whereas modes P_2 and N_2 show class confusion. Assuming that we have access to a training dataset with adequate representation from every mode in the data, let us consider learning pair-wise classifiers, $C_{i,j}$, to distinguish between every pair of positive and negative modes, P_i and N_j. This would result in an ensemble of classifiers which can then be applied on any unlabeled instance in a test scenario to estimate its class label. Now let us consider a test scenario involving instances from P_1 and N_1, denoted by $S_{1,1}$. Since P_1 and N_1 are easily separable in

the feature space and both P_1 and N_1 do not participate in any class confusion, test instances in $S_{1,1}$ would be correctly labeled even by a single classifier that discriminates between all positive and negative modes.

However, if we consider a test scenario $S_{1,2}$ involving instances from P_1 and N_2, we would notice that even though P_1 and N_2 are easily separable in the feature space, the presence of class confusion between P_2 and N_2 would hamper the classification performance at N_2, since instances belonging to N_2 can be easily misclassified to be belonging to P_2. To overcome this challenge, consider the following simplistic approach: let us assign a relevance score to every pair-wise classifier, $C_{i,j}$, in accordance with its likelihood of being used in the context of a test scenario. In particular, classifiers that discriminate between modes having a higher likelihood of being observed given the distribution of instances in a test scenario would receive higher relevance scores. Using this approach, we can assign a relevance score to every pair-wise classifier for both test scenarios, $S_{1,1}$ and $S_{1,2}$, and consider it to be either "Relevant" or "Not Relevant", as summarized in Table 5.1. For $S_{1,1}$, the only relevant classifier would then be $C_{1,1}$, which would correctly label all test instances in $S_{1,1}$. However, for $S_{1,2}$, both $C_{1,2}$ and $C_{2,2}$ would be considered as relevant, as the test instances in $S_{1,2}$ would show high likelihood for all the three modes, P_1, P_2, and N_2. However, $C_{2,2}$ would show poor cross-validation accuracy on the training set, since it discriminates between a pair of confusing modes, P_2 and N_2. $C_{2,2}$ could thus be discarded from the set of relevant classifiers, resulting in the only relevant classifier for $S_{1,2}$ to be $C_{1,2}$. $C_{1,2}$ would then be able to correctly label all test instances in $S_{1,2}$, and thus avoid class confusion in this particular situation. Note that the ability of the above simplistic scheme in overcoming class confusion arises from the fact that the distribution of test instances belonging to a test scenario contains reasonable information about its local context. We use this property as a guiding principle for motivating our proposed approach.

We propose the Adaptive Heterogeneous Ensemble Learning (AHEL) algorithm that takes into account the context of test instances belonging to a test scenario for overcoming class confusion in certain scenarios. We demonstrate the effectiveness of our approach in comparison with baseline approaches on a synthetic dataset and a real-world application involving global water monitoring. The remainder of this chapter is organized as follows: Section 5.2 provides a brief overview of related work. Section 5.3

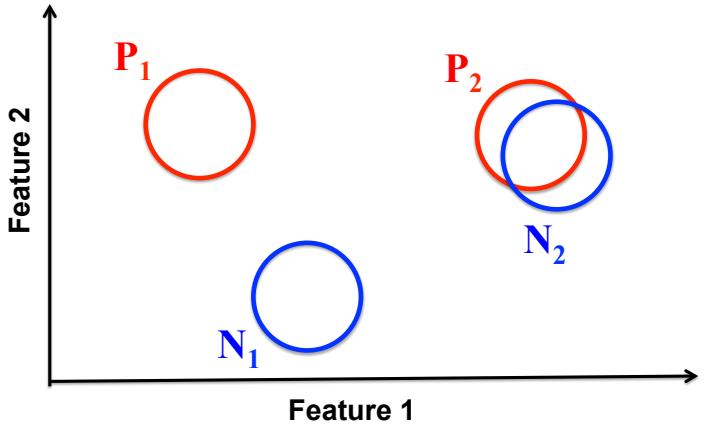


Figure 5.2: A toy dataset showing multi-modality within the classes, where P_2 and N_2 show class confusion.

	Test Scenario	
Classifier	$S_{1,1}$	$S_{1,2}$
$C_{1,1}$	"Relevant"	"Not Relevant"
$C_{1,2}$	"Not Relevant"	"Relevant"
$C_{2,1}$	"Not Relevant"	"Not Relevant"
$C_{2,2}$	"Not Relevant"	"Relevant"

Table 5.1: Table summarizing whether a particular classifier, $C_{i,j}$ is relevant for a particular test scenario or not.

presents the proposed approach. Section 5.4 presents experimental results. Section 5.5 includes concluding remarks and discusses directions for future work.

5.2 Related Work

The presence of multi-modality within the classes and its impact on classification performance has been previously discussed in [52], where the concept of modes was introduced as "small disjuncts". The impact of overlapping modes on the performance of a classifier has also been empirically analyzed in [53]. Furthermore, an ensemble learning approach for binary classification was recently presented in [54], that made use of the heterogeneity within the classes for constructing ensembles, instead of using random partitions of

the input data. It was shown that such an ensemble learning method is able to capture the heterogeneity within the classes and thus result in improved classification performance. However, none of these approaches are suitable for handling the phenomena of class confusion by making use of the local context of a test scenario.

Existing approaches that make use of the context of test instances for adapting its labeling decisions involve local learning algorithms, e.g. the k-Nearest Neighbor (KNN) algorithm [55] and other concept-based local learning algorithms [56, 57]. These algorithms make use of training instances only in the local neighborhood of an individual test instance for estimating its class label. However, none of these approaches are designed to account for multi-modality within the classes and to incorporate information about a group of instances belonging to a test scenario as opposed to using the locality of an individual test instance. The use of unlabeled instances as a guide in the learning process has also been explored by semi-supervised learning [20] and transductive learning [58] approaches. The primary objective of such approaches is to address the paucity of labeled data by making use of the structure in the test instances, e.g. using clustering approaches [59]. This is different from our problem since our primary objective is to use the unlabeled instances for inferring the classification context of a test scenario involving confusing modes, even in the presence of sufficient training data. Another body of research that considers adapting the learning of a classifier in the context of a test scenario involves techniques for handling concept drift [60–62], and transfer learning approaches [15]. However none of these approaches have explored the presence of multi-modal distribution within each of the two classes, and are thus not directly relevant for our problem.

5.3 Proposed Approach

Notations Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_1^n$ denote the training dataset with n labeled instances, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \{-1, +1\}$ is its binary response label. Let us assume that this training dataset comprises of n_+ positively labeled instances, denoted by $\mathcal{X}_+ = \{\mathbf{x}_i\}_1^{n_+}$, and n_- negatively labeled instances, denoted by $\mathcal{X}_- = \{\mathbf{x}_i\}_1^{n_-}$. Given this training dataset, our objective is to estimate the binary response, $y \in \{-1, 1\}$, for every test instance, $\mathbf{x}$, belonging to a test scenario, $\mathcal{X}_S =$

$\{\mathbf{x}_i\}_1^s$.

We present the Adaptive Heterogeneous Ensemble Learning (AHEL) algorithm that comprises of the following steps:

5.3.1 Learning the Multi-modality in Training Data

We assume that our training dataset, $\mathcal{D}$, contains a variety of instances from all possible positive and negative modes in the data, but explicit information about the multi-modal structure of the two classes is not known and needs to be inferred. To achieve this, we consider clustering the training instances belonging to each of the two classes separately, similar to the approach used in [54]. This results in the decomposition of the positive class, $\mathcal{X}_+$, into m_+ clusters or modes and the negative class, $\mathcal{X}_-$, into m_- clusters or modes, respectively. The choice of the clustering algorithm and the number of clusters, m_+ and m_-, used for representing the multi-modality within the classes depends on the characteristics of the data. For every cluster label c, let $\mathcal{X}_c$ denote the set of training instances with cluster label c, where c can either be one of the positive cluster labels, P_1 to P_{m_+}, or the negative cluster labels, N_1 to N_{m_-}.

We further consider every cluster label c to have an associated conditional probability distribution, $\mathcal{P}(\mathbf{x}|c)$, for every instance $\mathbf{x} \in \mathbb{R}^d$. This can either be available as a by-product of the clustering algorithm or can be inferred from the distribution of instances in $\mathcal{X}_c$. As an example, we consider $\mathcal{P}(x|c)$ to follow a normal distribution in the feature space with the sample mean, $\bar{\mathbf{x}_c}$, as its center and with unit variance, whenever $\mathcal{P}(x|c)$ is not explicitly available during the clustering process. However, it should be noted that the choice of the probability distribution used for representing $\mathcal{P}(x|c)$ depends on the target application and can be acquired via domain knowledge.

5.3.2 Constructing an Ensemble of Classifiers

We construct an ensemble of classifiers to discriminate between every pair of positive and negative cluster labels in $\mathcal{D}$, similar in essence to the Bipartite One-vs-One (BOVO) ensemble construction strategy proposed in [54]. This ensures adequate representation of every mode in the ensemble construction process, along with maintaining sufficient

diversity among the classifiers. This can be contrasted with traditional ensemble learning approaches for binary classification, e.g. bagging, boosting, and random forests, which make use of random partitions of the training data as opposed to using a stratified sampling of the training instances in accordance with the multi-modal structure of the two classes.

For every pair of positive and negative cluster labels, (P_i, N_j), we learn a classifier, f_l, to discriminate between $\mathcal{X}_{P_i}$ and $\mathcal{X}_{N_j}$, using an appropriate choice of the base classifier. This results in the learning of an ensemble of classifiers, $\{f_1, \ldots, f_{m^\star}\}$, where $m^\star = m_+ \times m_-$. We further compute the cross-validation accuracy of every classifier, f_l, using 5-fold cross-validation on $\mathcal{X}_{P_i}$ and $\mathcal{X}_{N_j}$, and use it as a measure of the accuracy of f_l, denoted by $Acc(f_l)$.

5.3.3 Assigning Adaptive Weights to Classifiers

For every classifier, f_l, we assign it a weight, $w(f_l, \mathcal{X}_S)$, representing its importance of being used for classification in the context of a test scenario, $\mathcal{X}_S$. In particular, we want to assign higher weights to classifiers that discriminate between pairs of modes that have a higher likelihood of being observed, given the distribution of instances in a test scenario, $\mathcal{X}_S$. Such a weighting scheme is achieved as follows.

For every test instance $\mathbf{x}$ belonging to $\mathcal{X}_S$, we compute its probability of being generated from a mode c as $\mathcal{P}(\mathbf{x}|c)$. We can then assign a relevance score to every mode c, denoted by $\mathcal{R}(c, \mathcal{X}_S)$, which indicates its likelihood of being observed given the distribution of instances in $\mathcal{X}_S$, defined as:

$$\mathcal{R}(c, \mathcal{X}_S) = \sum_{\mathbf{x} \in \mathcal{X}_S} \mathcal{P}(\mathbf{x}|c) \tag{5.1}$$

For a classifier, f_l, that discriminates between P_i and N_j, the relevance of using f_l in the context of $\mathcal{X}_S$, denoted by $\mathcal{R}(f_l, \mathcal{X}_S)$, depends on the relevance of observing modes P_i and N_j in $\mathcal{X}_S$, and can be estimated as:

$$\mathcal{R}(f_l, \mathcal{X}_S) = \mathcal{R}(P_i, \mathcal{X}_S) \times \mathcal{R}(N_j, \mathcal{X}_S) \tag{5.2}$$

$\mathcal{R}(f_l, \mathcal{X}_S)$ ensures that classifiers receive high weights only if both the modes involved in learning f_l have a high likelihood of being observed in $\mathcal{X}_S$. Each classifier f_l is further assigned a score, $\alpha(f_l)$, denoting its ability to differentiate between its pair of participating modes. $\alpha(f_l)$ can be computed as:

$$\alpha(f_l) = \begin{cases} Acc(f_l), & \text{if } Acc(f_l) > 0.6. \\ 0, & \text{otherwise.} \end{cases}$$

The weight of a classifier f_l in the context of test scenario $\mathcal{X}_S$ is then estimated as:

$$w(f_l, \mathcal{X}_S) = \alpha(f_l) \times \mathcal{R}(f_l, \mathcal{X}_S) \tag{5.3}$$

To illustrate the usefulness of $w(f_l, \mathcal{X}_S)$ in choosing the appropriate set of classifiers, especially in the presence of class confusion, consider a test scenario $\mathcal{X}_S$ that involves instances from P_c and N_{nc}, such that P_c shows class confusion with some other mode N_c not present in $\mathcal{X}_S$. In such a situation, P_c, N_c, and N_{nc} would receive the highest relevance scores in the context of $\mathcal{X}_S$. By taking the products of the relevance scores, the two classifiers that would receive the highest relevance scores would then be the ones that separate (P_c and N_c) and (P_c and N_{nc}). On the other hand, none of the pair-wise classifiers separating P_c, N_c, and N_{nc} from some other mode, O, will have a high relevance score, due to the low relevance score of O. The classifier separating (P_c and N_c) will eventually receive a low weight owing to its poor cross-validation accuracy and will be discarded. Thus, the classifier separating (P_c and N_{nc}) will be appropriately selected with the highest weight, resulting in adequate classification performance even in the presence of class confusion.

Note that our proposed weighting scheme inherently assumes that every test scenario involves a subset of positive and negative modes that are separable among each other but may show class confusion with other modes observed globally that are not present in the current test scenario. It is also assumed that a test scenario involving a confusing mode has instances from both the classes, thus requiring the use of a classifier in the first place. Furthermore, the ability of the above weighting scheme in avoiding class confusion hinges on the presence of atleast a single non-confusing mode in the test

scenario, which can dominate the assignment of relevance scores to classifiers.

5.3.4 Combining Ensemble Responses

We apply the ensemble of classifiers on a test instance, $\mathbf{x} \in \mathcal{X}_S$, to obtain a vector of ensemble responses, $f(\mathbf{x}) = [f_1(\mathbf{x}), ..., f_{m^\star}(\mathbf{x})]$. For each ensemble response, $f_l(\mathbf{x})$, we compute its loss w.r.t. a cluster label, c, as follows:

$$\text{Loss}(c, f_l) = \begin{cases} L(+f_l), & \text{if } c = P_i. \\ L(-f_l), & \text{if } c = N_j. \\ 0, & \text{otherwise.} \end{cases}$$

where, P_i and N_j are the positive and negative cluster labels used for learning f_l, and $L(z)$ is an appropriate loss function, e.g. the hinge loss function, $L(z) = \max\{1 - z, 0\}$, commonly used with support vector machines (SVMs) as base classifiers. The combined loss of all ensemble responses w.r.t a cluster label c is then defined as:

$$\text{Loss}(c, f(\mathbf{x})) = \sum_{l=1}^{m^\star} w(f_l, \mathcal{X}_S)\text{Loss}(c, f_l) \tag{5.4}$$

We choose $\hat{c}$ as the cluster label which provides the minimum loss, $\hat{c} = \arg\min_c \text{Loss}(c, f(\mathbf{x}))$. The test instance $\mathbf{x}$ is then classified as positive if $\hat{c}$ is a positive cluster label, otherwise it is classified as negative.

5.4 Experimental Results

We compared the performance of AHEL with the baseline approach of learning a single non-linear classifier, termed as the GLOBAL approach. We also compared our results with the Bipartite One-vs-One (BOVO) ensemble learning approach that was presented in [54], which is able to handle heterogeneity within the classes but is unable to adapt its learning using the local context of a test scenario. In order to compare our performance with local learning algorithms, we considered the k-nearest neighbor (KNN) algorithm with $k = 5$ as a baseline approach. Furthermore, in order to emphasize the importance of using the distribution of an entire group of instances belonging to a test scenario as

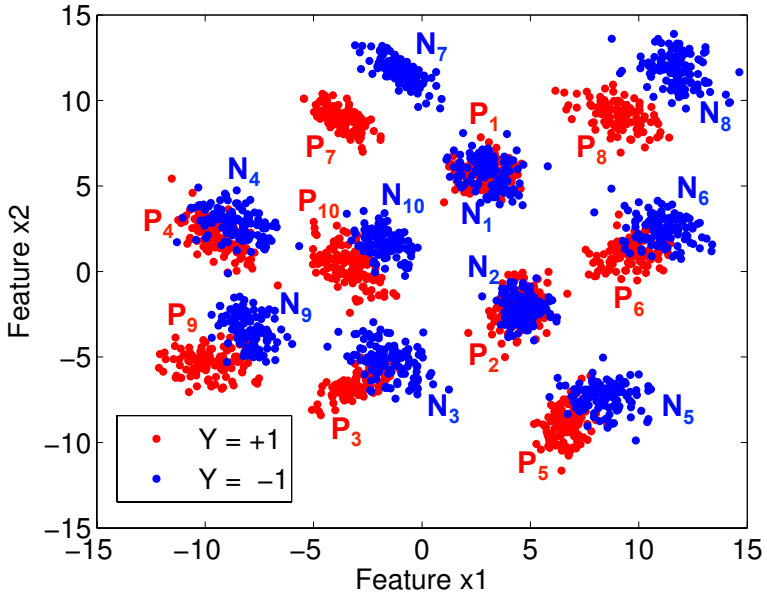


Figure 5.3: Synthetic dataset with 10 positive modes: P_1 to P_{10}, and 10 negative modes: N_1 to N_{10}, with varying degrees of class confusion among pairs of modes.

opposed to an individual test instance, we considered a variant of our algorithm that uses instance-specific information for assigning weights to ensemble classifiers, termed as the Instance-specific Heterogeneous Ensemble Learning (IHEL) algorithm. Specifically, IHEL considers the relevance of using a classifier f_l on a test instance $\mathbf{x}$ as $\mathcal{R}(f_l, \mathbf{x}) = \max(\mathcal{P}(\mathbf{x}|P_i), \mathcal{P}(\mathbf{x}|N_j))$, where f_l discriminates between P_i and N_j. IHEL thus follows the same formulation as AHEL, except for the fact that it uses $\mathcal{R}(f_l, \mathbf{x})$ in place of $\mathcal{R}(f_l, \mathcal{X}_S)$.

We used support vector machines (SVMs) with radial basis function (RBF) kernel as the base classifier for the GLOBAL approach and all ensemble learning methods used in this chapter. The optimal hyper-parameters of SVM were chosen using 5- fold cross-validation on the training set in every experiment. The number of positive and negative clusters were kept equal in all experiments ($m_+ = m_- = m$). The classification error rate was used as the evaluation metric for comparing the performance of classification algorithms in every experiment.

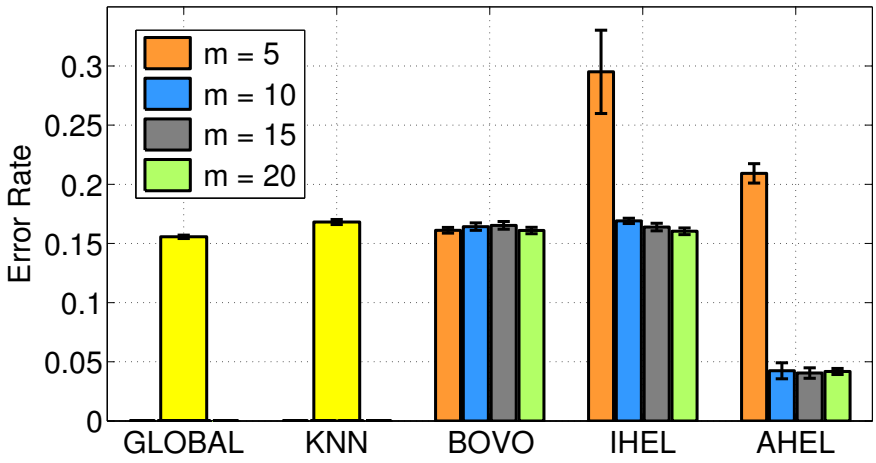


Figure 5.4: Comparing classification performance on synthetic dataset.

5.4.1 Results on Synthetic Dataset

We considered the synthetic dataset shown in Figure 5.3, which comprises of 10 positive and 10 negative modes, where every mode is generated using a bi-variate Gaussian distribution. Note that some pairs of modes in this dataset are easily separable (e.g. P_7 and N_7), while others show a high degree of class confusion (e.g. P_1 and N_1). These synthetic modes are representative of the variety of positive and negative modes that are experienced in real-world classification problems. We randomly sampled 200 instances each from every positive and negative mode for constructing the global training dataset. To simulate a variety of test scenarios, we randomly sampled 1000 instances each from every pair of positive and negative modes, P_i and N_j, to construct 100 test scenarios, $S_{i,j}$. The random sampling procedure for obtaining the training and test sets was repeated 10 times.

Figure 5.4 compares the error rates of competing classification algorithms on the overall test set, comprising of instances from all possible 100 test scenarios. The bi-secting K-means (BKM) algorithm [29] was used as the preferred clustering strategy for BOVO, IHEL, and AHEL, with varying number of clusters, m. It can be seen that both GLOBAL and BOVO have error rates close to 0.15, since they are unable to incor-porate the local context of test scenarios for overcoming class confusion. Furthermore, techniques that use instance-specific context of individual test instances, namely KNN

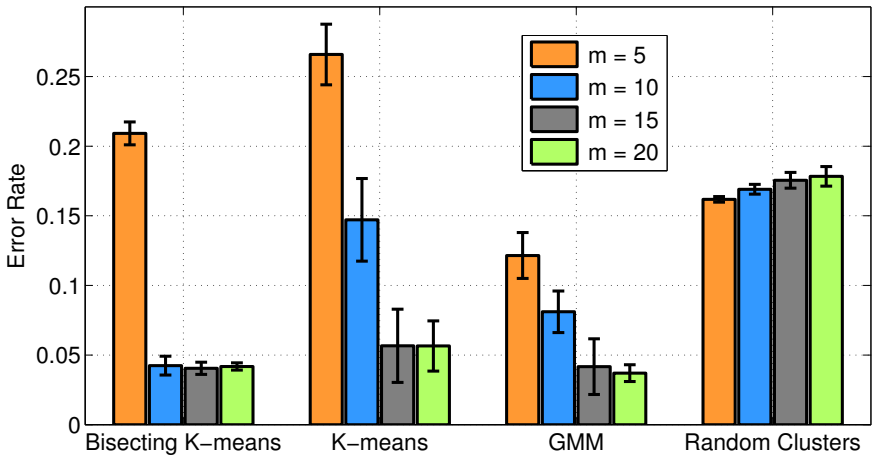


Figure 5.5: Varying the clustering strategy used in AHEL.

and IHEL, show no significant improvement than GLOBAL. In contrast, AHEL shows a significant reduction in the error rate for $m \geq 10$ when compared with all the baseline approaches, since it uses the overall distribution of instances belonging to a test scenario for adapting its learning.

Figure 5.5 compares the performance of AHEL using varying clustering algorithms and number of clusters (m) used to represent the multi-modality within the classes. It can be seen that the performance of AHEL is initially poor for $m = 5$ because the clustering is unable to capture the heterogeneity within the classes, resulting in under-clustering, which degrades the performance of AHEL. However, as m is increased from 5 to 20, AHEL is able to adequately capture the heterogeneity within the classes and thus show drastic improvements in classification performance for all clustering algorithms. Note that the the performance of AHEL using Bisecting K-means is better than that of AHEL using K-means and Gaussian Mixture Model (GMM) clustering for $m \geq 10$, due to the tendency of K-means and GMM clustering to merge larger clusters and thus exhibit under-clustering. However, the performance of AHEL does not deteriorate even in the presence of over-clustering as m is increased from 10 to 20. Instead, the variance of the error rates of AHEL keeps decreasing as m is increased beyond 10, demonstrating the robustness of AHEL even with a large number of ensemble classifiers. Figure 5.5 also shows that the performance of AHEL is significantly better when a

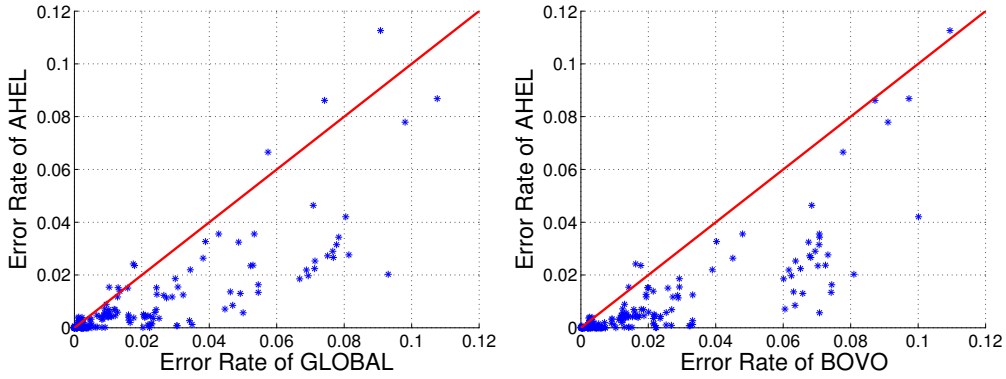
meaningful clustering strategy is used (e.g. BKM, K-means, and GMM), instead of using an artificial partitioning of the data into random clusters, demonstrating the utility of using information about the multi-modality within the two classes while learning classifier ensembles.

5.4.2 Global Water Monitoring Results

We consider a real-world application of AHEL for monitoring water bodies at a global scale using remote sensing variables. Monitoring water bodies is important for effective water management and for understanding the impact of human actions and climate change on water bodies. To this end, remote sensing variables capture a variety of information about the Earth's surface that can be used for labeling every location on the Earth at a given time as water or land (binary classes). However, the presence of a rich variety of land and water categories that exist at a global scale makes it challenging to perform global water monitoring. There is an opportunity to overcome this challenge by using the local context of a test scenario, involving test instances observed in the vicinity of the same water body at the same time-step.

We used the seven reflectance bands collected by the MODerate-resolution Imaging Spectoradiometer (MODIS) instruments onboard NASA's satellites as the set of features for classification, which are available at 500m resolution for every 8 days. Ground truth information was obtained via the Shuttle Radar Topography Mission's (SRTM) Water Body Dataset (SWBD), which provides a mapping of all water bodies for a large fraction of the Earth (60^o S to 60^oN), but for a single date: Feb 18, 2000. We considered a diverse set of 99 lakes collected from different regions of the world for the purpose of evaluation. For each lake, we created a buffer region of 20 pixels at 500m resolution around the periphery of the water body, and used the buffer region as well as the interior of the water body to construct the evaluation dataset. After removing instances at the immediate boundaries of the water bodies and ignoring instances with missing values, this evaluation dataset comprised of ≈ 1.3 million data instances, where every instance had an associated binary label of water (positive) or land (negative). We randomly sampled 2000 instances each from both classes to construct the global training dataset. The remainder of the evaluation dataset was considered for testing. Since different pairs of water and land categories appear together in different regions of the world and at

different times, we needed to consider test scenarios involving different pairs of water and land categories for the purpose of evaluation. To achieve this, we first clustered the water and land classes in the test set into $m = 15$ clusters each using the Bisecting K-means clustering algorithm. Every pair of water and land clusters, (W_i, L_j), was then considered as a different test scenario, $S_{i,j}$. We repeated the sampling procedure for obtaining the training and test sets 10 times.

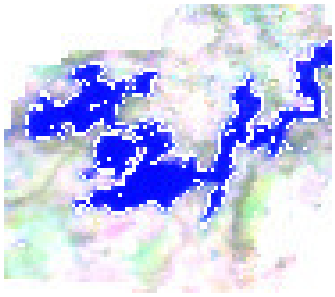


(a) Comparing AHEL and GLOBAL (b) Comparing AHEL and BOVO

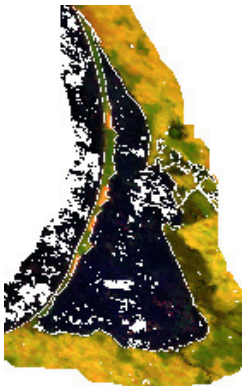
Figure 5.6: Scatter plots of mean error rates of Global, BOVO, and AHEL across all test scenarios.

Figure 5.6 presents scatter plots comparing the performance of AHEL with baseline approaches individually across all 225 test scenarios. Every point on a scatter plot compares the mean error rate of two classification algorithms on a particular test scenario, where the red line in each scatter plot shows the plot of $y = x$ for ease of comparison. It can be seen that AHEL shows drastic improvements in classification performance than GLOBAL and BOVO across a vast majority of test scenarios. In order to assess the statistical significance of the differences in the classification performance, we computed the p-value of AHEL showing lower mean error rate than GLOBAL and BOVO over all 225 test scenarios using one-tailed Wilcoxon signed rank tests, which came out to be equal to 1.74×10^{-25} and 2.02×10^{-35} respectively. This shows that the improvements in classification performance of AHEL are statistically significant.

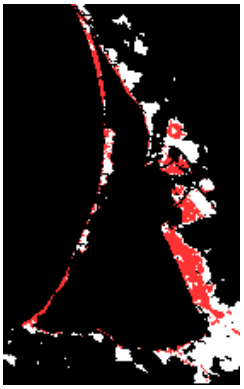
We next analyze the differences in the performance of AHEL and baseline approaches over two illustrative test scenarios, $S_{5,1}$ and $S_{10,1}$. Figure 5.7 compares the classification



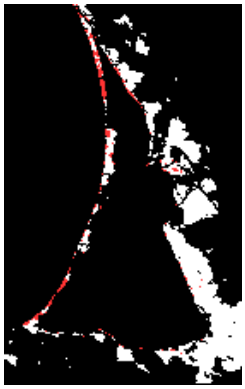
(a) False color composite image of water in Gariep Dam, South Africa



(b) False color composite image of land near Curonian Lagoon, Russia



(c) Errors of GLOBAL (shown in red) over L_1 (shown in white)

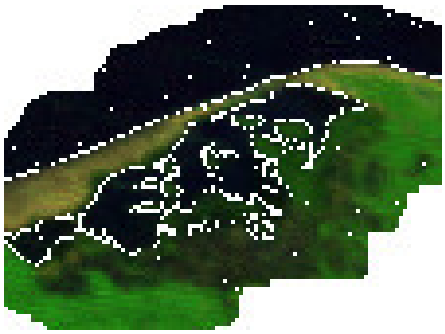


(d) Errors of AHEL (shown in red) over L_1 (shown in white)

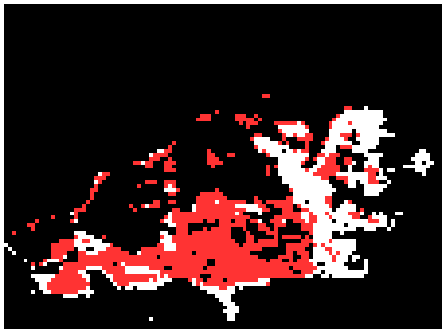
Figure 5.7: Comparing GLOBAL and AHEL at $S_{5,1}$.

performance of GLOBAL and AHEL on the test scenario $S_{5,1}$ involving W_5 and L_1. Figure 5.7(a) shows the false color composite image (using the 7^{th}, 5^{th}, and 4^{th} bands, as red, green and blue colors respectively) of Gariep Dam in South Africa, which has all its water instances coming from W_5, shown in blue color. Figure 5.7(b) shows the false color composite image of Curonian Lagoon in Russia, which has a portion of its land from the land category L_1, indicated as red and white pixels in Figures 5.7(c) and 5.7(d). For these instances belonging to category L_1, Figures 5.7(c) and 5.7(d) show the misclassifications (errors) of GLOBAL and AHEL respectively as red pixels. It can be observed that GLOBAL is making errors over a large portion of L_1 as compared to

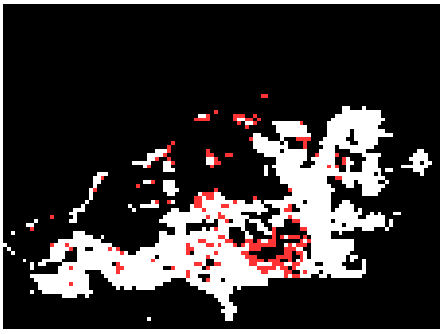
(a) False color composite image of water in Lake Tana, Ethiopia



(b) False color composite image of land near Burullus Lake, Egypt



(c) Errors of BOVO (shown in red) over L_1 (shown in white)



(d) Errors of AHEL (shown in red) over L_1 (shown in white)

Figure 5.8: Comparing BOVO and AHEL at $S_{10,1}$.

AHEL. This is because L_1 comprises of land instances that appear very close to shallow water (see the false color in Figure 5.7(b)), resulting in its class confusion in the global training set. However, the false color of W_5 in Figure 5.7(a) can be seen to be very different from that of L_1 in Figure 5.7(b). Hence, in the local context of $S_{5,1}$, AHEL is able to handle the class confusion and thus show improved classification performance. The mean error rates of GLOBAL and AHEL for $S_{5,1}$ are 0.081 and 0.027 respectively. Figure 5.8 presents a similar analysis of the performance of BOVO and AHEL for the test scenario $S_{10,1}$. The mean error rates of BOVO and AHEL for $S_{10,1}$ are 0.07 and 0.019 respectively.

5.5 Conclusions and Future Work

We consider binary classification problems where both classes show a multi-modal distribution in the feature space and the classification has to be performed over different test scenarios, where every test scenario involves only a subset of all the positive and negative modes in the data. We propose the Adaptive Heterogeneous Ensemble Learning (AHEL) algorithm that constructs an ensemble of classifiers to discriminate between every pair of positive and negative modes, and uses the local context of test scenarios for adaptively weighting the ensemble of classifiers. We demonstrate the effectiveness of AHEL in comparison with baseline approaches on a synthetic dataset and a real-world application involving global water monitoring. Future extensions of our work could explore variants of our weighting scheme that can account for the imbalance among the classes, commonly experienced in real-world classification problems. Future work can also focus on studying the theoretical properties of AHEL, which can help in generalizing it to handle a broader family of class confusion scenarios in the presence of multi-modality within the classes.

Chapter 6

Theory-guided Data Science

Many of the discussions in the previous chapters on learning with population heterogeneity was built around the use of physical knowledge to inform predictive learning frameworks with the context of every data instance. This is part of a broader paradigm of research explored in this thesis to systematically integrate scientific knowledge with data science methods, for improved accuracy as well as physical consistency of the generated results. In this chapter, we build the foundations of this emerging paradigm, termed as *theory-guided data science*, by describing several ways of combining scientific knowledge with data science methods, that have started to gain attention in a variety of scientific and engineering disciplines.

6.1 Introduction

As we enter into the era of "big data," the scale and speed with which data science methods are proliferating almost every application task is unprecedented. Apart from transforming commercial industries such as retail and advertising, data science is also beginning to play an important role in advancing scientific discovery. Historically, science has progressed by first generating hypotheses (or theories) and then collecting data to confirm or refute these hypotheses. However, in the big data era, ample data, which is being continuously collected without a specific theory or hypothesis in mind, offers further opportunity for discovering new knowledge. Indeed, the role of data science in scientific disciplines is beginning to shift from providing simple analysis tools (e.g.,

detecting particles in Large Hadron Collider experiments [63, 64]) to providing full-fledged knowledge discovery frameworks (e.g., in bio-informatics [65] and climate science [66, 67]). Based on the success of data science in applications where Internet-scale data is available (with billions or even trillions of samples), e.g., natural language translation, optical character recognition, object tracking, and most recently, autonomous driving, there is a growing anticipation of similar accomplishments in scientific disciplines [68–70]. To capture this excitement, some have even referred to the rise of data science in scientific disciplines as "the end of theory" [71], the idea being that the increasingly large amounts of data makes it possible to build actionable models without using scientific theories.

Unfortunately, this notion of black-box application of data science has met with limited success in scientific domains (e.g., [72–74]). A well-known example of the perils in using data science methods in a theory-agnostic manner is Google Flu Trends, where a data-driven model was learned to estimate the number of influenza-related physician visits based on the number of influenza-related Google search queries in the United States [75]. This model was built using search terms that were highly correlated with the flu propensity in the Center for Disease Control (CDC) data. Despite its initial success, this model later overestimated the flu propensity by more than a factor of two, as measured by the number of influenza-related doctor visits in subsequent years, according to CDC data [73].

There are two primary characteristics of knowledge discovery in scientific disciplines that have prevented data science models from reaching the level of success achieved in commercial domains. First, scientific problems are often under-constrained in nature as they suffer from paucity of representative training samples while involving a large number of physical variables. Further, physical variables commonly show complex and non-stationary patterns that dynamically change over time. For this reason, the limited number of labeled instances available for training or cross-validation can often fail to represent the true nature of relationships in scientific problems. Hence, standard methods for assessing and ensuring generalizability of data science models may break down and lead to misleading conclusions. In particular, it is easy to learn spurious relationships that look deceptively good on training and test sets (even after using methods such as cross-validation), but do not generalize well outside the available labeled data.

This was one of the main reasons behind the failure of Google Flu Trends, since the data used for training the model in the first few years was not representative of the trends in subsequent years [73]. The paucity of representative samples is one of the prime challenges that differentiates scientific problems from mainstream problems involving Internet-scale data such as language translation or object recognition, where large volumes of labeled or unlabeled data have been critical in the success of recent advancements in data science such as deep learning.

The second primary characteristic of scientific domains that have limited the success of black-box data science methods is the basic nature of scientific discovery. While a common end-goal of data science models is the generation of actionable models, the process of knowledge discovery in scientific domains does not end at that. Rather, it is the translation of learned patterns and relationships to *interpretable* theories and hypotheses that leads to advancement of scientific knowledge, e.g., by explaining or discovering the physical cause-effect mechanisms between variables. Hence, even if a black-box model achieves somewhat more accurate performance but lacks the ability to deliver a mechanistic understanding of the underlying processes, it cannot be used as a basis for subsequent scientific developments. Further, an interpretable model, that is grounded by explainable theories, stands a better chance at safeguarding against the learning of spurious patterns from the data that lead to non-generalizable performance. This is especially important when dealing with problems that are critical in nature and associated with high risks (e.g., healthcare).

The limitations of black-box data science models in scientific disciplines motivate a novel paradigm that uses the unique capability of data science models to automatically learn patterns and models from large data, without ignoring the treasure of accumulated scientific knowledge. We refer to this paradigm that attempts to integrate scientific knowledge and data science as *theory-guided data science* (TGDS). The paradigm of TGDS has already begun to show promise in scientific problems from diverse disciplines. Some examples include the discovery of novel climate patterns and relationships [76, 77], closure of knowledge gaps in turbulence modeling efforts [78, 79], discovery of novel compounds in material science [80–82], design of density functionals in quantum chemistry [83], improved imaging technologies in bio-medical science [84, 85], discovery of genetic biomarkers [86], and the estimation of surface water dynamics at

a global scale [87, 88]. These efforts have been complemented with recent review papers [66, 89–91], workshops (e.g., a 2016 conference on physics informed machine learning [92]) and industry initiatives (e.g., a recent IBM Research initiative on "physical analytics" [93]).

This chapter attempts to build the foundations of theory-guided data science by presenting several ways of bringing scientific knowledge and data science models together, and illustrating them using examples of applications from diverse domains. A major goal of this chapter is to formally conceptualize the paradigm of "theory-guided data science", where scientific theories are systematically integrated with data science models in the process of knowledge discovery.

The remainder of this chapter is structured as follows. Section 6.2 provides an introduction to the paradigm of theory-guided data science and presents an overview of research themes in TGDS. Sections 6.3, 6.4, 6.5, 6.6, and 6.7 describe several approaches in every research theme of TGDS, using illustrative examples from diverse disciplines. Section 6.8 provides concluding remarks.

6.2 Summary of Paradigm

A common problem in scientific domains is to represent relationships among physical variables, e.g., the combustion pressure and launch velocity of a rocket or the shape of an aircraft wing and its resultant air drag. The conventional approach for representing such relationships is to use models based on scientific knowledge, i.e., theory-based models, which encapsulate cause-effect relationships between variables that have either been empirically proven or theoretically deduced from first principles. These models can range from solving closed-form equations (e.g. using Navier–Stokes equation for studying laminar flow) to running computational simulations of dynamical systems (e.g. the use of numerical models in climate science, hydrology, and turbulence modeling). An alternate approach is to use a set of training examples involving input and output variables for learning a data science model that can automatically extract relationships between the variables.

As depicted in Figure 6.1, theory-based and data science models represent the two

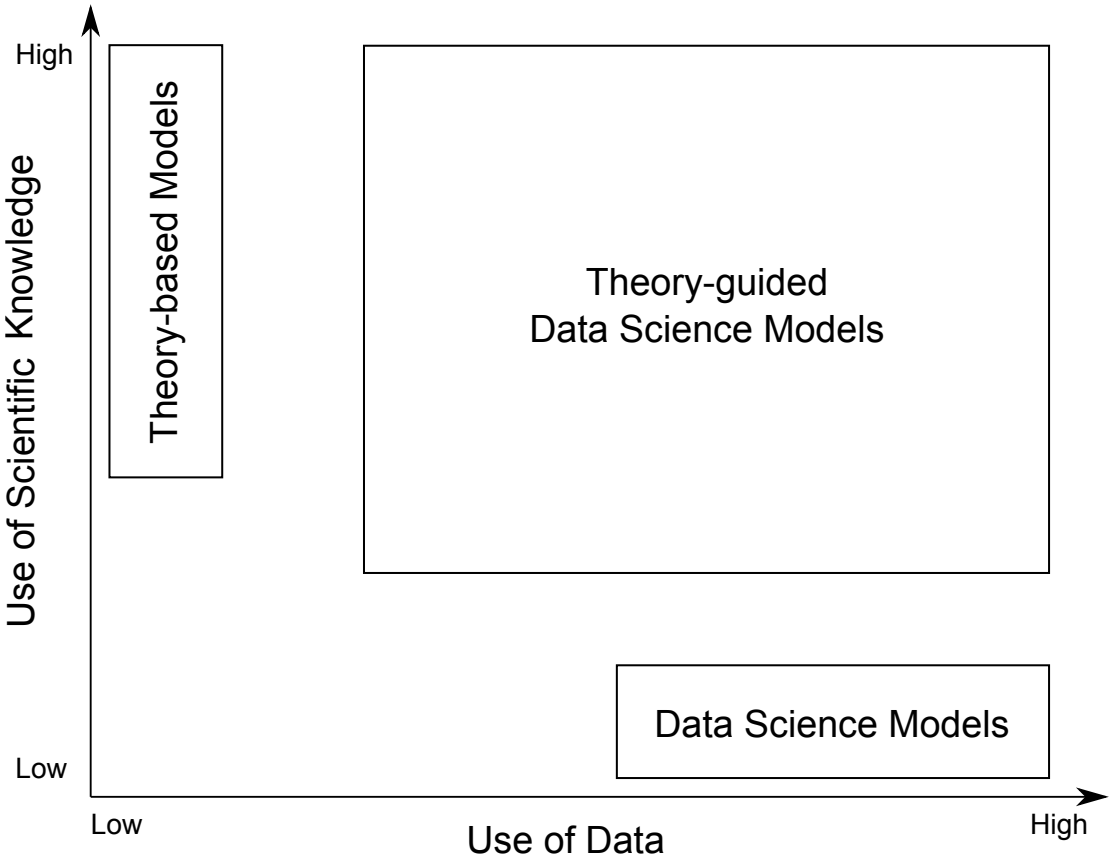


Figure 6.1: A representation of knowledge discovery methods in scientific applications. The x-axis measures the use of data while the y-axis measures the use of scientific knowledge. Theory-guided data science explores the space of knowledge discovery that makes ample use of the available data while being observant of the underlying scientific knowledge.

extremes of knowledge discovery, which depend on only one of the two sources of information available in any scientific problem, i.e., scientific knowledge or data. They both enjoy unique strengths and have found success in different types of applications. Theory-based models (see top-left corner of Figure 6.1) are well-suited for representing processes that are conceptually well understood using known scientific principles. On the other hand, traditional data science models mainly rely on the information contained in the data and thus reside in the bottom-right corner of Figure 6.1. They have a wide range of applicability in domains where we have ample supply of representative data samples, e.g., in Internet-scale problems such as text mining and object recognition.

Despite their individual strengths, theory-based and data science models suffer from certain deficiencies when applied in problems of great scientific relevance, where both theory and data are currently lacking. For example, a number of scientific problems involve processes that are not completely understood by our current body of knowledge, because of the inherent complexity of the processes. In such settings, theory-based models are often forced to make a number of simplifying assumptions about the physical processes, which not only leads to poor performance but also renders the model difficult to comprehend and analyze. We illustrate this scenario using the following example from hydrological modeling.

Example 1 (Hydrological Modeling)**.** One of the primary objectives of hydrology is to study the processes responsible for the movement, distribution, and quality of water across the planet. Some examples of such processes include the discharge of water from the atmosphere via precipitation, and the infiltration of water underneath the Earth's surface, known as subsurface flow. Understanding subsurface flow is important as it is intricately linked with terrestrial ecosystem processes, agricultural water use, and sudden adverse events such as floods. However, our knowledge of subsurface flow using state-of-the-art hydrological models is quite limited [94]. This is mainly because subsurface flow operates in a regime that is difficult to measure directly using *in-situ* sensors such as boreholes. In addition, subsurface flow involves a number of complex sub-processes that interact in non-linear ways, which are difficult to encapsulate in current theory-based models [95]. Due to these challenges, existing hydrological models make use of a broad range of parameters in several weakly-informed physical equations. Thus, global hydrological models tend to show poor predictive performance in describing

subsurface flow processes [96]. In addition, they also lose physical interpretability due to the large number of model parameters that are difficult to interpret meaningfully with respect to the domain.

■

If we apply "black-box" data science models in scientific problems, we would notice a completely different set of issues arising due to the inadequacy of the available data in representing the complex spaces of hypotheses encountered in physical domains. Further, since most data science models can only capture associative relationships between variables, they do not fully serve the goal of understanding causative relationships in scientific problems.

Hence, neither a data-only nor a theory-only approach can be considered sufficient for knowledge discovery in complex scientific applications. Instead, there is a need to explore the continuum between theory-based and data science models, where both theory and data are used in a synergistic manner. The paradigm of *theory-guided data science* (TGDS) attempts to address the shortcomings of data-only and theory-only models by seamlessly blending scientific knowledge in data science models (see Figure 6.1). By integrating scientific knowledge in data science models, TGDS aims to learn dependencies that have a sufficient grounding in physical principles and thus have a better chance to represent causative relationships. TGDS further attempts to achieve better generalizability than models based purely on data by learning models that are consistent with scientific principles, termed as *physically consistent models*.

To illustrate the role of "consistency with scientific knowledge" in ensuring better generalization performance, consider the example of learning a parametric model for a predictive learning problem using a limited supply of labeled samples. Ideally, we would like to learn a model that shows the best generalization performance over any unseen instance. Unfortunately, we can only observe the model performance on the available training set, which may not be truly representative of the true generalization performance (especially when the training size is small). In recognition of this fact, a number of learning frameworks have been explored to favor the selection of *simpler* models that may have lower accuracy on the training data (compared to more complex models) but are likely to have better generalization performance. This methodology, that builds on the well-known statistical principle of bias-variance trade-off [97], can be

described using Figure 6.2.

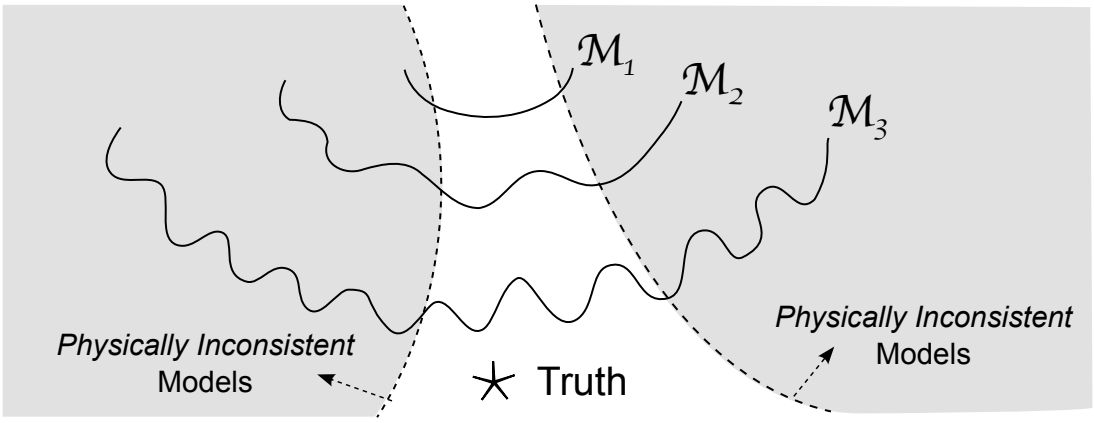


Figure 6.2: Scientific knowledge can help in reducing the model variance by removing physically inconsistent solutions, without likely affecting their bias.

Figure 6.2 shows an abstract representation of a succession of model families with varying levels of complexity (shown as curved lines), where $\mathcal{M}_1$ represents the set of least complex models while $\mathcal{M}_3$ contains highly complex models. Every point on the curved lines represents a model that a learning algorithm can arrive at, given a particular realization of training instances. The *true* relationship between the input and output variables is depicted as a star in Figure 6.2. We can observe that the learned models belonging to $\mathcal{M}_3$, on average, are quite close to the true relationship. However, even a small change in the training set can bring about large changes in the learned models of $\mathcal{M}_3$. Hence, $\mathcal{M}_3$ shows low *bias* but high *variance*. On the other hand, models belonging to $\mathcal{M}_1$ are quite robust to changes in the training set and thus show low variance. However, $\mathcal{M}_1$ shows high bias as its models are generally farther away from the true relationship as compared to models of $\mathcal{M}_3$. It is the trade-off between reducing bias and variance that is at the heart of a number of machine learning algorithms [97–99].

In scientific applications, there is another source of information that can be used to ensure the selection of generalizable models, which is the available scientific knowledge. By pruning candidate models that are inconsistent with known scientific principles (shown as shaded regions in Figure 6.2), we can significantly reduce the variance of models without likely affecting their bias. A learning algorithm can then be focused on the

space of physically consistent models, leading to generalizable and scientifically interpretable models. Hence, one of the overarching visions of TGDS is to include physical *consistency* as a critical component of model performance along with training accuracy and model complexity. This can be summarized in a simple way by the following revised objective of model performance in TGDS:

$$\text{Performance} \propto \text{Accuracy} + \text{Simplicity} + \mathit{Consistency}.$$

There are various ways of introducing physical consistency in data science models, in different forms and capacities. While some approaches attempt to naturally incorporate physical consistency in existing learning frameworks of data science models, others explore innovative ways of blending data science principles with theory-based models. In the following sections, we describe five broad categories of approaches for combining scientific knowledge with data science, that are illustrative of emerging examples of TGDS research in diverse disciplines. Note that many of these approaches can be applied together in multiple combinations for a particular problem, depending on the nature of scientific knowledge and the type of data science method. The five research themes of TGDS can be briefly summarized as follows.

First, scientific knowledge can be used in the design of model families to restrict the space of models to physically consistent solutions, e.g., in the selection of response and loss functions or in the design of model architectures. These techniques are discussed in Section 6.3. Second, given a model family, we can also guide a learning algorithm to focus on physically consistent solutions. This can be achieved, for instance, by initializing the model with physically meaningful parameters, by encoding scientific knowledge as probabilistic relationships, by using domain-guided constraints, or with the help of regularization terms inspired by our physical understanding. These techniques are discussed in Section 6.4. Third, the outputs of data science models can be refined using explicit or implicit scientific knowledge. This is discussed in Section 6.5. Fourth, another way of blending scientific knowledge and data science is to construct *hybrid* models, where some aspects of the problem are modeled using theory-based components while other aspects are modeled using data science components. Techniques for constructing hybrid TGDS models are discussed in Section 6.6. Fifth, data science methods can also help

in augmenting theory-based models to make effective use of observational data. These approaches are discussed in Section 6.7.

6.3 Theory-guided Design of Data Science Models

An important decision in the learning of data science models is the choice of model family used for representing the relationships between input and response variables. In scientific applications, if the domain knowledge suggests a particular form of relationship between the inputs and outputs, care must be taken to ensure that the same form of relationship is used in the data science model. Here, we discuss two different ways of using scientific knowledge in the design of data science models. First, we can use synergistic combinations of response and loss functions (e.g. in generalized linear models or artificial neural networks) that not only simplify the optimization process and thus lead to low training errors, but are also consistent with our physical understanding and hence result in generalizable solutions. Another way to infuse domain knowledge is by choosing a model architecture (e.g. the placement of layers in artificial neural networks) that is compliant with scientific knowledge. We discuss both these approaches in the following.

6.3.1 Theory-guided Specification of Response

Many data science models provide the option for specifying the form of relationship used for describing the response variable. For example, a generic family of models, which can represent a broad variety of relationships between input and response variables, is the generalized linear model (GLM). There are two basic building blocks in a GLM, the link function $g(.)$, and the probability distribution $P(y|\mathbf{x})$. Using these building blocks, the expected mean μ of the target variable y is determined as a function of the weighted linear combination of inputs, $\mathbf{x}$, as follows:

$$\begin{aligned} g(\mu) &= \mathbf{w}^T\mathbf{x} + b, \quad \text{or equivalently,} \\ \mu &= g^{-1}(\mathbf{w}^T\mathbf{x} + b), \end{aligned} \tag{6.1}$$

where $\mathbf{w}$ and b and the parameters of GLM to be learned from the data. Some common

choices of link and probability distribution functions are listed in Table 6.1, resulting in varying types of regression models.

To ensure the learning of GLMs that produce physically meaningful results, it is important to choose an appropriate specification of the response variable that matches with domain understanding. For example, while modeling response variables that show extreme effects (highly skewed distributions), e.g., occurrences of unusually severe floods and droughts, it would be inappropriate to assume the response variable to be Gaussian distributed (the standard assumption used in linear regression models). Instead, a regression model that uses the Gumbel distribution to model extreme values would be more accurate and physically meaningful.

In general, the idea of specifying model response using scientific principles can be explored in many types of learning algorithms. An example of theory-guided specification of response can be found in the field of ophthalmology, where the use of Zernike polynomials was explored by Twa et al. [100] for the classification of corneal shape using decision trees.

6.3.2 Theory-guided Design of Model Architecture

Scientific knowledge can also be used to influence the architecture of data science models. An example of a data science model that provides ample room for tuning the model architecture is artificial neural networks (ANN), which has recently gained widespread acceptance in several applications such as vision, speech, and language processing. There are a number of design considerations that influence the construction of an effective ANN model. Some examples include the number of hidden layers and the nature of connections among the layers, the sharing of model parameters among nodes, and the choice of activation and loss functions for effective model learning. Many of these design

Table 6.1: Table showing some commonly used combinations of link function and probability distribution functions in generalized linear models.

Name	Link Function	Probability Distribution
Linear	μ	Gaussian
Poisson	$\log(\mu)$	Poisson
Logistic	$\log(\mu/(1-\mu))$	Binomial

considerations are primarily motivated to simplify the learning procedure, minimize the training loss, and ensure robust generalization performance using statistical principles of regularization.

There is a huge opportunity in informing these design considerations with our physical understanding of a problem, to obtain generalizable as well as scientifically interpretable results. For example, in an attempt to build a model of the brain that learns view-invariant features of human faces, the use of biologically plausible rules in ANN architectures was recently explored in [101]. It was observed that along with preserving view-invariance, such theory-guided ANN models were able to capture a known aspect of human neurology (namely, the mirror-symmetric tuning to head orientation) that was being missed by traditional ANN models. This made it possible to learn scientifically interpretable models of human cognition and thus advance our understanding of the inner workings of the brain. In the following, we describe two promising directions for using scientific knowledge while constructing ANN models: by using a modular design that is inspired by domain understanding, and by specifying the connections among the nodes in a physically consistent manner.

Domain knowledge can be used in the design of ANN models by decomposing the overall problem into modular sub-problems, each of which represents a different physical sub-process. Every sub-problem can then be learned using a different ANN model, whose inputs and outputs are connected with each other in accordance with the physical relationships among the sub-processes. For example, in order to describe the overall hydrological process of surface water discharge, we can learn modular ANN models for different sub-processes such as the atmospheric process of rainfall and evaporation, the process of surface water runoff, and the process related to groundwater seepage. Every ANN model can be fed with appropriately chosen domain features at the input and output layers. This will help in using the power of deep learning frameworks while following a high-level organization in the ANN architecture that is motivated by domain knowledge.

Domain knowledge can also be used in the design of ANN models by specifying node connections that capture theory-guided dependencies among variables. A number of variants of ANN have been explored to capture spatial and temporal dependencies between the input and output variables. For example, recurrent neural networks (RNN)

are able to incorporate the sequential context of time in speech and language processing [102]. RNN models have been recently explored to capture notions of long and short term memory (LSTM) with the help of skip connections among nodes to model information delay [103]. Such models can be used to incorporate time-varying domain characteristics in scientific applications. For example, while surface water runoff directly influences surface water discharge without any delay, groundwater runoff has a longer latency and contributes to the surface water discharge after some time lag. Such differences in time delay can be effectively modeled by a suitably designed LSTM model. Another variant of ANN is the convolutional neural network (CNN) [104], which has been widely applied in vision and image processing applications to capture spatial dependencies in the data. It further facilitates the sharing of model parameters so that the learned features are invariant to simple transformations such as scaling and transformation. Similar approaches can be explored to share the parameters (and thus reduce model complexity) over more generic similarity structures among the input features that are based on domain knowledge.

6.4 Theory-guided Learning of Data Science Models

Having chosen a suitable model design, the next step of model building involves navigating the search space of candidate models using a learning algorithm. In the following, we present four different ways of guiding the learning algorithm to choose physically consistent models. First, we can use physically consistent solutions as initial points in iterative learning algorithms such as gradient descent methods. Second, we can restrict the space of probabilistic models with the help of theory-guided priors and relationships. Third, scientific knowledge can be used as constraints in optimization schemes for ensuring physical consistency. Fourth, scientific knowledge can be encoded as regularization terms in the objective function of learning algorithms. We describe each of these approaches in the following.

6.4.1 Theory-guided Initialization

Many learning algorithms that are iterative in nature require an initial choice of model parameters as a first step to commence the learning process. For such algorithms, an

inferior initialization can lead to the learning of a poor model. Domain knowledge can help in the process of model initialization so that the learning algorithm is guided at an early stage to choose generalizable and physically consistent models.

An example of theory-guided initialization of model parameters includes a recent matrix completion approach for plant trait analysis [105], where the rows of the matrix correspond to plants from diverse environments while the columns correspond to plant traits such as leaf area, seed mass, and root length. Since observations about plant traits are sparsely available, such a plant trait matrix would be highly incomplete [106]. Filling the missing entries in a plant trait matrix can help us understand the characteristics of different plant species and their ability to adapt to varying environmental conditions. A traditional data science approach to this problem is to use matrix completion algorithms that have found great success in online recommender systems [107]. However, many of these algorithms are iterative in nature and use fixed or random values to initialize the matrix. In the presence of domain knowledge, we can improve these algorithms by using the species mean of every attribute as initial values in the matrix completion process. This relies on the basic principle that the species mean provides a robust estimate of the average behavior across all organisms. This approach has been shown to provide significant improvements in the accuracy of predicting plant traits over traditional methods [105]. Changes from the species mean can also be learned using subsequent matrix completion operations, which could be physically interpreted as the effect of varying environmental conditions on plant traits.

One of the data science models that requires special efforts in choosing an appropriate combination of initial model parameters is the artificial neural network, which is known to be susceptible to getting stuck at local minimas, saddle points, and flat regions in the loss curve. In the era of deep learning, much progress has been made to avoid the problem of inferior ANN initialization with the help of *pretraining* strategies. The basic idea of these strategies is to train the ANN model over a simpler problem (with ample availability of representative data) and use the trained model to initialize the learning for the original problem. These pretraining strategies have made major impact on our ability to learn complex hierarchies of features in several application domains such as speech and image processing. However, they rely on plentiful amounts of unlabeled or labeled data and hence are not directly applicable in scientific domains where the data

sizes are small relative to the number of variables. One way to address this challenge is by devising novel pretraining strategies where computational simulations of theory-based models are used to initialize the ANN model. This can be especially useful when theory-based models can produce approximate simulations quickly, e.g., approximate model simulations of turbulent flow (see Example 5). Such pretrained theory-guided ANN models can then be fine-tuned using expert-quality ground truth.

6.4.2 Theory-guided Probabilistic Models

Probabilistic graphical models provide a natural way to encode domain-specific relationships among variables as edges between nodes representing the variables. However, manually encoding domain knowledge in graphical models requires a great deal of expert supervision, which can be cumbersome for problems involving a large number of variables with complex interactions–a common feature of scientific problems. In the presence of a large number of nodes, it is common to apply automated graph estimation techniques such as the use of graph Lasso [108]. The basic objective of such techniques is to estimate a sparse inverse covariance matrix that maximizes the model likelihood given the data. To assist such techniques with scientific knowledge, a promising research direction is to explore graph estimation techniques that maximize data likelihood while limiting the search to physically consistent solutions.

Another approach to reduce the variance of model parameters (and thus avoid model overfitting) is to introduce priors in the model space. An example of the use of theory-guided priors is the problem of non-invasive electrophysiological imaging of the heart. In this problem, the electrical activity within the walls of the heart needs to be predicted based on the ECG signal measured on the torso of a subject. There are approximately 2000 locations in the walls of the heart where electrical activity needs to be predicted, based on ECG data collected from approximately 100 electrodes on the torso. Given the large space of model parameters and the paucity of labeled examples with ground-truth information, a traditional black-box model that only uses the information contained in the data is highly prone to learning spurious patterns. However, apart from the knowledge contained in the data, we also have domain knowledge (represented using electrophysiological equations) about how electrical signals are transmitted within the heart via the myocardial fibre structure. These equations can be used to determine the

spatial distribution of the electric signals in the heart at time t based on the predicted electric signals at $t-1$. Incorporating such theory-guided spatial distributions as priors and using it along with externally collected ECG data in a hierarchical Bayesian model has been shown to provide promising results over traditional data science models [84,85]. Another example of theory-guided priors can be found in the field of geophysics [109], where the knowledge of convection-diffusion equations was used as priors for determining the connectivity structure of subsurface aquifers.

6.4.3 Theory-guided Constrained Optimization

Constrained optimization techniques are extensively used in data science models for restricting the space of model parameters. For example, support vector machines use constraints for ensuring separability among the classes, while maximizing the margin of the hyperplane. There is also a rich literature on constraint-based pattern mining [110, 111] and clustering [112]. The use of constraints provides a natural way to integrate domain knowledge in the learning of data science models. In scientific applications where theory-based constraints can be represented using linear equality or inequality conditions, they can be readily integrated in existing constrained optimization formulations, which are known to provide computationally efficient solutions especially when the objective function is convex.

However, many scientific problems involve constraints that are represented in complex forms, e.g., using partial differential equations (PDE) or non-linear transformations of variables, which are not easily handled by traditional constrained optimization methods. For example, the Naiver–stokes equation for momentum expresses the following constraint between the flow velocity $\mathbf{v}$ and the fluid pressure p:

$$\rho\Big(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\Big) = -\nabla p + \nabla\cdot(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}),$$

where ρ is the fluid density, μ is the fluid dynamic viscosity, and ∇ represents the gradient operator with respect to the spatial coordinates.

To utilize such complex forms of constraints in data science models, it is necessary to develop constrained optimization techniques that can use common forms of partial differential equations encountered in scientific disciplines. An example of a data-driven

approach that uses domain-driven PDEs can be found in a recent work in climate science [113, 114], where physically constrained time-series regression models were developed to incorporate memory effects in time as well as the nonlinear noise arising from energy-conserving interactions.

In the following, we present detailed discussions of two illustrative examples of the use of theory-guided constraints. While Example 2 explores the use of constraints for predicting electron density in computational chemistry, Example 3 explores the use of elevation-based constraints among locations for mapping surface water dynamics.

Example 2 (Computational Chemistry)**.** In computational chemistry, solving Schrödinger's equation is at the basis of all quantum mechanical calculations for predicting the properties of solids and molecules. Schrödinger's equation can be expressed as

$$\begin{aligned} \mathbf{H}\Psi &= \mathbf{E}\Psi, && (6.2)\\ &= (\mathbf{T}+\mathbf{U}+\mathbf{V})\Psi, && (6.3) \end{aligned}$$

where $\mathbf{H}$ is the electronic Hamiltonian operator, Ψ is the wavefunction that describes the quantum state of the system, and $\mathbf{E}$ is the total energy consisting of three terms, the kinetic energy, $\mathbf{T}$, the electron-electron interaction energy, $\mathbf{U}$, and the potential energy arising due to external fields, $\mathbf{V}$ (e.g., due to positively charged nuclei). Since the computational complexity in directly solving the Schrödinger's equation grows rapidly with the number of particles, N, it is infeasible for solving large many-particle systems in practical applications.

To address this, a new class of quantum chemical modeling approaches was developed by Hohenberg and Kohn in 1964 [115], which uses the electron density $n(r)$ as a basic primitive in all calculations, instead of the wavefunction Ψ. This has resulted in the rise of density functional theory (DFT) methods, which have become a standard tool for solving many-particle systems. In DFT, every variable can be expressed as a functional of the electron density function $n(r)$ (where a functional is a function of functions). For example, the total energy $\mathbf{E}$ can be expressed in terms of functionals of $n(r)$ as follows:

$$\mathbf{E}[n] = \mathbf{T}[n] + \mathbf{U}[n] + \mathbf{V}[n]. \quad (6.4)$$

The density, $n_0(r)$, that leads to the lowest total energy, $\mathbf{E}[n_0]$, is known as the ground-state density of the system, which is a critical quantity to determine.

However, obtaining $n_0(r)$ is challenging because of the interaction functional, $\mathbf{U}[n]$, whose exact form is unknown. Different approximations of the interaction term have been developed to solve for the ground-state density of a system, the most notable being the class of Kohn-Sham (KS) DFT methods. However, their performance is sensitive to the quality of approximation used in modeling the interactions. Also, KS DFT methods have a computational complexity of $O(N^3)$, which makes them challenging to apply on large systems.

To overcome the challenges in existing DFT methods, a recent work by Li et al. [83] explored the use of data science models to approximate $\mathbf{T}[n]$, and use such approximations to predict the ground-state density, $n_0(r)$. In this work, kernel ridge regression methods were used to model the kinetic energy, $\mathbf{T}[n]$, of a 4-particle system as a functional of its electron density, $n(r)$. Having learned $\hat{\mathbf{T}}[n]$, we can obtain the ground-state energy, $n_0(r)$, using the following Euler-Lagrangian equation:

$$\frac{\delta \hat{\mathbf{T}}[n_0]}{\delta n_0(r)} = \mu - v(r), \tag{6.5}$$

where $v(r)$ is the external potential and μ is an adjustable constant. This imposes a theory-guided constraint on the model learning, such that $\hat{\mathbf{T}}[n]$ must not only show good performance in predicting the kinetic energy, but should also accurately estimate the ground-state density, $n_0(r)$, using Equation 6.5. A functional that adheres to this constraint can be called "self-consistent."

It was shown in [83] that a regression model that only focuses on minimizing the training error leads to highly inconsistent solutions of the ground-state density, and is thus not useful for quantum chemical calculations. This inconsistency can be traced to the inability of regression models in capturing functional derivative forms that are used in Equation 6.5. In particular, the derivative of $\hat{\mathbf{T}}[n]$ can easily leave the space of densities observed in the training set, and thus arrive at ill-conditioned solutions especially when the training size is small.

To overcome this limitation, a modified Euler-Lagrange constraint was proposed in [83], which restricted the space of $n_0(r)$ to the density manifold observed in the

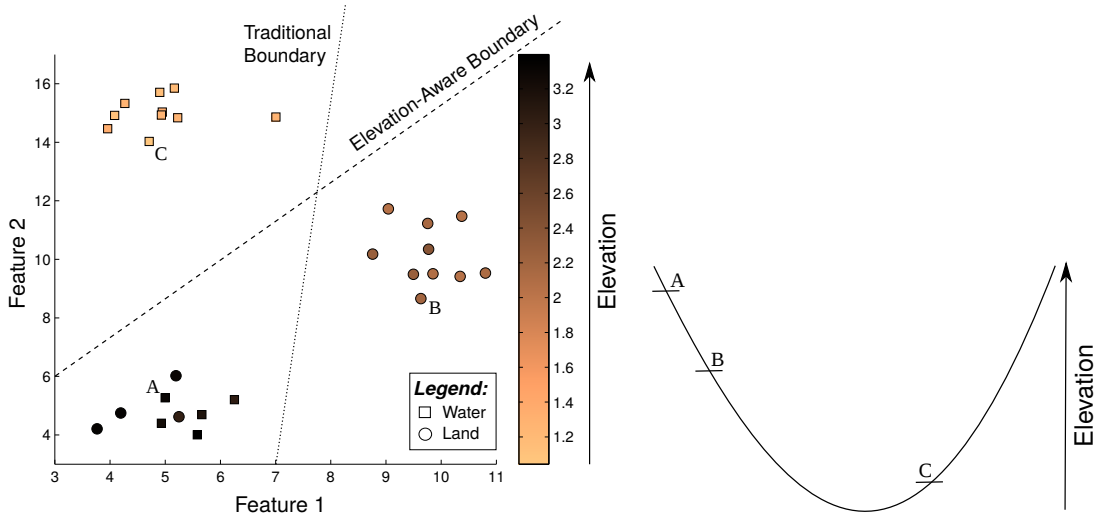
training set. This helped in learning accurate as well as self-consistent ground-state densities using the knowledge contained in the data as well as domain theories.

■

Example 3 (Mapping Surface Water Dynamics)**.** Remote sensing data from Earth observing satellites presents a promising opportunity for monitoring the dynamics of surface water body extent at regular intervals of time. It is possible to build predictive models that use multi-spectral data from satellite images as input features to classify pixels of the image as water or land. However, these models are challenged by the poor quality of labeled data, noise and missing values in remote sensing signals, and the inherent variability of water and land classes over space and time [6, 116].

To address these challenges, there is an opportunity for improving the quality of classification maps by using the domain knowledge that water bodies have a concave elevation structure. Hence, locations at a lower elevation are filled up first before the water level reaches locations at higher elevations. Thus, if we have access to elevation information (e.g. from bathymetric measurements obtained via sonar instruments), we can use it to constrain the classifier so that it not only minimizes the training error in the feature space but also produces labels that are consistent with the elevation structure. To illustrate this, consider an example of a two-dimensional training set shown in Figure 6.3(a), where the squares and circles represent training instances belonging to water and land classes, respectively. Along with the features, we also have information about the elevation of every instance, shown using the intensity of colored points in Figure 6.3(a).

If we disregard the elevation information and learn a linear classifier to simply minimize the training errors, we would learn the decision boundary shown using a dotted line in Figure 6.3(a). This classifier would make some mistakes in the lower-left corner of the feature space, where the class confusion is difficult to resolve using a linear separator. However, if we use the elevation information, we can see that the entire group of instances in the lower lower-left corner has a higher elevation than the instances shown on the right (labeled as land), and are thus less likely to be filled with water. For example, notice that location A is at a higher elevation than both B and C (see Figure 6.3(b)). Hence, if B is labeled as land, it would be inconsistent to classify A as water and instead it should be classified as land. The use of such constraints can help in learning a generalizable classification model even with poorly labeled training data.



(a) Distribution of water and land training samples from a specific water body in feature space. Shading reflects elevation information at the locations of training samples.

(b) Lake cross-section.

Figure 6.3: An illustrative example of the use of elevation-based ordering (domain theory) for learning physically consistent classification boundaries of water and land. Along with the distribution of training instances in the feature space, we also have information about their elevation, as shown in Figure 6.3(a)). This information can be used to learn an elevation-aware classification boundary that produces physically viable labels, e.g. if B is labeled as land, then A must necessarily be labeled as land as it is at a higher elevation, as shown in Figure 6.3(b).

■

6.4.4 Theory-guided Regularization

One way to constrain the search space of model parameters is to use regularization terms in the objective function, which penalize the learning of overly complex models. A number of regularization techniques have been explored in the data science community to enforce different measures of model complexity. For example, minimizing the L_p norm of model parameters has been extensively used for obtaining various effects of regularization in parametric model learning. While the L_2 norm has been used to avoid overly large parameter values in ridge regression and support vector machines, minimizing the L_1 norm results in the Lasso formulation and the Dantzig selector, both

of which encode sparsity in the model parameters.

However, these techniques are agnostic to the physical feasibility of the learned model and thus can lead to physically inconsistent solutions. For example, while predicting the elastic modulus using bond energy and melting point, Lasso may favor melting point over bond energy even though a direct causal link exists between bond energy and the modulus [89]. This can result in the elimination of meaningful attributes and the selection of secondary attributes that are not directly relevant. Hence, there is a need to devise regularization techniques that can incorporate scientific knowledge to restrict the search space of model parameters. For example, instead of using the L_p norm for regularization, we can find solutions on physically consistent sub-spaces of models. The Gaussian widths of such sub-spaces can be used as a regularization term in techniques such as the generalized Dantzig selector [117, 118]. In the following, we describe two research directions for theory-guided regularization that have been explored in different applications: using variants of Lasso to incorporate domain-specific structure among parameters, and the use of multi-task learning formulations to account for the heterogeneity in data sub-populations.

The group Lasso [119] is a useful variant of Lasso that has been explored in problems involving structured attributes. It assumes the knowledge of a grouping structure among the attributes, where only a small number of groups are considered relevant. As an example in bio-marker discovery, the groups of attributes may correspond to sets of bio-markers that are related via a common biological pathway. Group Lasso helps in selecting physically meaningful groups of attributes in the data science models, and various extensions of group Lasso have been explored for handling different types of domain characteristics, e.g., overlapping group Lasso [120], tree-guided group Lasso [121], and sparse group Lasso [122].

In recent work [123], applications of sparse group Lasso were explored to model the domain characteristics of climate variables. In this work, climate variables observed over a range of spatial locations were used to predict a climate phenomenon of interest. By treating the set of variables observed at every location as a group, the use of group Lasso ensured that if a location is selected, all of the climate variables observed at that location will be used as relevant features. Such features thus represent meaningful (spatially coherent) regions in space that can be studied to identify physical pathways

of relationships in climate science.

Another example of Lasso-based regularization that encodes domain knowledge can be found in the problem of discovering genetic markers for diseases. In this problem, data-driven approaches such as elastic nets are traditionally used to determine the relative importance of genetic markers in the context of a disease. However, geneticists understand that the relevant markers typically are located in close proximity on the genome sequence due to a property called linkage disequilibrium, which suggests that genetic information that is closely located travels together between generations of the population. This domain knowledge can be incorporated as a regularizer to ensure that the discovered genetic markers are typically located in close proximity on the genome. In fact, Liu and colleagues [86] introduced a smoothed minimax concave penalty to Lasso that captured squared differences in regression coefficients between adjacent markers to ensure that the difference in genetic effects between adjacent markers is small.

Domain knowledge can also be used to guide the regularization of a multi-task learning (MTL) model, as explored for the problem of forest cover estimation in [7]. In the presence of heterogeneity in data sub-populations, different groups of instances in the data show different relationships between the inputs and outputs. For example, different types of vegetation (e.g. forests, farms, and shrublands) may show varying responses to a target variable in remote sensing signals. MTL provides a promising solution to handle sub-population heterogeneity in such cases, by treating the learning at every sub-population as a different task. Further, by sharing the learning at related tasks, MTL enforces a robust regularization on the learning across all tasks, even in the scarcity of training data.

However, most MTL formulations require explicit knowledge of the composition of every task and the similarity structure among the tasks, which is not always known in practical applications. For example, the exact number and distribution of vegetation types is often unavailable, and when they are known, they are available at varying granularties [6]. In recent work [7], the presence of heterogeneity due to varying vegetation types was first inferred by clustering vegetation time series, which was then used to induce similarity in the model parameters at related vegetation types. This resulted in an MTL formulation where the task structure was inferred using contextual variables, obtained using domain knowledge.

6.5 Theory-guided Refinement of Data Science Outputs

Domain knowledge can also be used to refine the outputs of data science models so that they are in compliance with our current understanding of physical phenomena. This style of TGDS leverages scientific knowledge at the final stage of model building where the outputs of any data science model are made consistent with domain knowledge. In the following, we describe some of the approaches for refining data science outputs using domain knowledge that is either explicitly known (e.g. in the form of closed-form equations or model simulations) or implicitly available (e.g. in the form of latent constraints).

6.5.1 Using Explicit Domain Knowledge

Data science outputs are often refined to reduce the effect of noise and missing values and thus improve the overall quality of the results. For example, in the analysis of spatio-temporal data, there is a vast body of literature on refining model outputs to enforce spatial coherence and temporal smoothness among predictions. Data science outputs can also be refined to improve a quality measure, e.g., in the discovery of frequent itemsets by *pruning* candidate patterns. Building on these methods, a promising direction is to develop model refinement approaches that make ample use of domain knowledge, encoded in the form of scientific theories, for producing physically consistent results.

An example of theory-guided refinement of data science outputs can be found in the problem of material discovery, where the objective is to find novel materials and crystal structures that show a desirable property, e.g., their ability to filter gases or to serve as a catalyst. Traditional approaches for predicting crystal structure and properties rely on *ab initio* calculations such as density functional theory methods. However, since the space of all possible materials is extremely large, it is impractical to perform computationally expensive *ab initio* calculations on every material to estimate their structure and properties. Recently, a number of teams in material science have explored the use of probabilistic graphical models for predicting the structure and properties of a material, given a training database of materials with known structure and properties [80–82]. This provided a computationally efficient approach to reduce the space of

candidate materials that show a desirable property, using the knowledge contained in the training data. The results of the data science models were then cross-checked using expensive *ab initio* calculations to further refine the model outputs. This line of research has resulted in the discovery of a hundred new ternary oxide compounds that were previously unknown using traditional approaches [80], highlighting the effectiveness of TGDS in advancing scientific knowledge.

6.5.2 Using Implicit Domain Knowledge

In scientific applications, the domain structure among the output variables may not always be known in the form of explicit equations that can be easily integrated in existing model refinement frameworks. This requires jointly solving the dual problem of inferring the domain constraints and using the learned constraints to refine model outputs. We illustrate this using an example in mapping surface water dynamics, where implicit constraints among locations (based on a hidden elevation ordering) are estimated and leveraged for refining classification maps of water bodies.

Example 4 (Post-processing using elevation constraints)**.** As described in Example 3, it is difficult to map the dynamics of surface water bodies by solely using the knowledge contained in remote sensing data, and there is promise in using information about the elevation structure of water bodies to assist classification models. However, such information is seldom available at the desired granularity for most water bodies around the world. Hence, there is a need to infer the latent ordering among the locations (based on their elevation) so that they can be used to produce accurate and physically consistent labels. One way to achieve this is by using the history of imperfect water/land labels produced by a data science model at every location over a long period of time. In particular, a location that has been classified as water for a longer number of time-steps has a higher likelihood of being at a deeper location than a location that has been classified as water less frequently. This implicit elevation ordering, if extracted effectively, can help in improving the classification maps by post-processing the outputs to be consistent with elevation ordering. Further, the post-processed labels can help in obtaining a better estimate of the elevation ordering, thus resulting in an iterative solution that

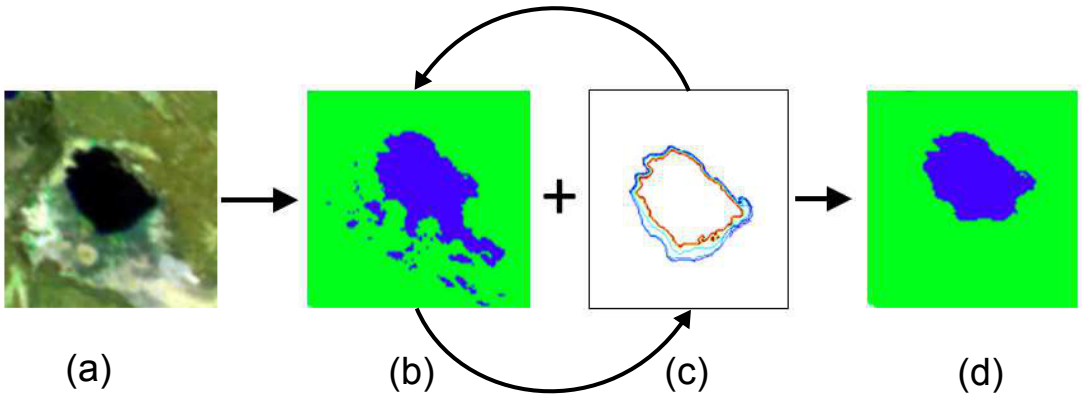


Figure 6.4: Mapping the extent of Lake Abhe (on the border of Ethiopia and Djibouti in Africa) using implicit theory-guided constraints. (a) Remote sensing image of the water body (prepared using multi-spectral false color composites). (b) Initial classification maps. (c) Elevation contours inferred from the history of classification labels. (d) Final classification maps refined using elevation-based constraints.

simultaneously infers the elevation ordering and produces physically consistent classification maps. This approach was successfully used in [87, 88] to build global maps of surface water dynamics. Figure 6.4 illustrates the effectiveness of this approach using an example lake in Africa, where the post-processed classification map does not suffer from the errors of the initial classification map and visually matches well with the remote sensing image of the water body.

■

Other examples of the use of implicit constraints includes mapping urbanization [124] and tree plantation conversions [125, 126], where hidden Markov models were used to incorporate domain knowledge about the transitions among land covers.

6.6 Learning Hybrid Models of Theory and Data Science

One way to combine the strengths of scientific knowledge and data science is by creating *hybrid* combinations of theory-based and data science models, where some aspects of the problem are handled by theory-based components while the remaining ones are modeled using data science components. There are several ways of fusing theory-based and data

science models to create hybrid TGDS models. One way is to build a two-component model where the outputs of the theory-based component are used as inputs in the data science component. This idea is used in climate science for statistical downscaling of climate variables [127], where the climate model simulations, available at coarse spatial and temporal resolutions, are used as inputs in a statistical model to predict the climate variables at finer resolutions. Theory-based model outputs can also be used to supervise the training of data science models, by providing physically consistent estimates of the target variable for every training instance.

An alternate way of creating a hybrid TGDS model is to use data science methods to predict intermediate quantities in theory-based models that are currently being missed or inaccurately estimated. By feeding data science outputs into theory-based models, such a hybrid model can not only show better predictive performance but also amend the deficiencies in existing theory-based models. Further, the outputs of theory-based models may also be used as training samples in data science components [128], thus creating a two-way synergy between them. Depending on the nature of the model and the requirements of the application, there can be multiple ways of introducing data science outputs in theory-based models. In the following, we provide an illustrative example of this theme of TGDS research in the field of turbulence modeling.

Example 5 (Turbulence Modeling)**.** One of the important problems in aerospace engineering is to model the characteristics of turbulent flow, which consists of chaotic changes in the flow velocity, and complex dissipation of momentum and energy. Turbulence modeling is used in a number of applications such as the design and reliability assessment of airfoils in aeroplanes and space vehicles. Key to the study of fluid dynamics is the Navier–Stokes equations, which describe the behavior of viscous fluids under motion. Although the Navier–Stokes equations can be readily applied in simple flow problems involving incompressible and irrotational flow, obtaining an exact representation for turbulent flow requires computationally expensive solutions such as direct numerical simulations (DNS) at fine spatial grids. The high computational costs of DNS make it infeasible for studying practical turbulence problems in the industry, which are typically solved using inexact but computationally cheap approximations. One such approximation is the Reynolds–averaged Navier–Stokes (RANS) equations, which introduces a term called as the Reynolds stress, τ, to represent the apparent stress due

to fluctuations caused by turbulence. Since the exact form of the Reynolds stress is unknown, different approximations of τ have been explored in previous studies, resulting in a variety of RANS models. Despite the continued efforts in approximating τ, current RANS models are still insufficient for modeling complex flows with separation, curvature, or swirling. To overcome their limitations, recent work by Wang et al. [79] explored the use of machine learning methods to assist RANS models and reduce their discrepancies. In particular, the Reynolds stress was approximated as

$$\tau = \tau_{RANS} + \Delta\tau_{ML}, \tag{6.6}$$

where τ_{RANS} is obtained from a RANS model while $\Delta\tau_{ML}$ is the model discrepancy that is estimated using a random forest model. Although this approach can be used with any generic RANS model to estimate its discrepancy, it does not alter the form of approximation used in obtaining τ_{RANS}, since $\Delta\tau_{ML}$ is learned independently of τ_{RANS}. In another work by Singh et al. [78], a machine learning component was used to directly augment a RANS approximation in the following manner:

$$-\tau_{ij} = 2\rho\nu S^*_{ij} - \frac{2}{3}\rho K\delta_{ij}, \tag{6.7}$$

$$\frac{D\nu}{Dt} = \beta \times \mathbf{P} - \mathbf{D} + \mathbf{T}, \tag{6.8}$$

where Equation 6.7 is the standard Boussinesq equation relating the Reynolds stress τ_{ij} to the effective viscosity ν, and Equation 6.8 is a variant of the Spalart Allmaras model that estimates ν as a function of a machine learning term, β (learned using an artificial neural network), and other physical terms, $\mathbf{P}$, $\mathbf{D}$, and $\mathbf{T}$, corresponding to production, destruction, and transport processes, respectively. This class of modeling framework, which integrates machine learning terms in theory-based models, has been called field inversion and machine learning (FIML) [129].

Both these works illustrate the potential of coupling data science outputs with theory-based models to reduce model discrepancies in complex scientific applications. The exact choice of the data science model and its contribution to the theory-based model can be explored in future investigations. Similar lines of TGDS research can be

explored in other domains where current theory-based models are lacking, e.g., hydrological models for studying subsurface flow [94].

■

6.7 Augmenting Theory-based Models using Data Science

There are many ways we can use data science methods to improve the effectiveness of theory-based models. Data can be assimilated in theory-based models for improved selection of model states in numerical models. Data science methods can also help in calibrating the parameters of theory-based models so that they provide a better realization of the physical system. We describe both these approaches in the following.

6.7.1 Data Assimilation in Theory-based Models

One of the long-standing approaches of the scientific community for integrating data in theory-based models is to use data assimilation approaches, which has been widely used in climate science and hydrology [130]. These domains typically involve dynamical systems, such as the progression of climate phenomena over time, which can be represented as a sequence of physical states in numerical models. Data assimilation is a way to infer the most likely sequence of states such that the model outputs are in agreement with the observations available at every time-step. In data assimilation, the values of the current state are constrained to depend on previous state values as well as the current data observations. For example, if we use the Gaussian distribution to model the linear transition between consecutive states, this translates to a Kalman filter. However, in general, the dependencies among the states in data assimilation methods are modeled using more complex forms of distributions that are governed by physical laws and equations. Data assimilation provides a promising step in the direction of integrating data with theory-based models so that the knowledge discovery approach relies both on scientific knowledge and observational data.

6.7.2 Calibrating Theory-based Models using Data

Theory-based models often involve a large number of parameters in their equations that need to be calibrated in order to provide an accurate representation of the physical

system. A naïve approach for model calibration is to try out every combination of parameter values, perhaps by searching over a discrete grid defined over the parameters, and choose the combination that produces the maximum likelihood for the data. However, this approach is practically infeasible when the number of parameters are large and every parameter takes many possible values. A number of computationally efficient approaches have been explored in different disciplines for parsimoniously calibrating model parameters with the help of observational data. For example, a seminal work on model calibration in the field of hydrology is the Generalized Likelihood Uncertainty Estimation (GLUE) technique [131]. This approach models the uncertainty associated with every parameter combination using Monte Carlo approaches, and uses a Bayesian formulation to incrementally update the uncertainties as new observations are made available. At any given iteration, the parameter combination that shows maximum agreement with the observations is employed in the model, the results of which are used to update the uncertainties on the next iteration.

The problem of parameter selection has recently received considerable attention in the machine learning community in the context of multi-armed bandit problems [132–134]. The basic objective in these problems is to incrementally select parameter values so that we can *explore* the space of parameter choices and *exploit* the parameter choice that provides the maximum reward, using a limited number of observations. Variants of these techniques have also been explored for settings where the parameters take continuous values instead of discrete steps [135, 136]. These techniques provide a promising direction for calibrating the high-dimensional parameters of theory-based models.

6.8 Conclusion

In this chapter, we formally conceptualized the paradigm of theory-guided data science (TGDS) that seeks to exploit the promise of data science without ignoring the treasure of knowledge accumulated in scientific principles. We provided a taxonomy of ways in which scientific knowledge and data science can be brought together in any application with some availability of domain knowledge. These approaches range from methods that strictly enforce physical consistency in data science models (e.g., while designing model

architecture or specifying theory-based constraints) to methods that allow a relaxed usage of scientific knowledge where our scientific understanding is weak (e.g., as priors or regularization terms). We presented examples from diverse disciplines to illustrate the various research themes of TGDS and also discussed several avenues of novel research in this rapidly emerging field.

One of the central motivations behind TGDS is to ensure better generalizability of models (even when the problem is complex and data samples are under-representative) by anchoring data science algorithms with scientific knowledge. TGDS also aims at advancing our knowledge of the physical world by producing scientifically interpretable models. Reducing the search space of the learning algorithm to physically consistent models may also have an additional benefit of reducing the computational cost of the algorithm.

The TGDS research themes are not exhaustive and we anticipate the development of novel TGDS themes in the future that explore innovative ways of blending scientific theory with data science. While most of the discussion in this chapter focuses on supervised learning problems, similar TGDS research themes can be explored for other traditional tasks of data mining, machine learning, and statistics. For example, the use of physical principles to constrain spatio-temporal pattern mining algorithms has been explored in [137, 138] for finding ocean eddies from satellite data. The need to explore TGDS models for uncertainty quantification is discussed in [91] in the context of understanding and projecting climate extremes. Scientific knowledge can also be used to advance other aspects of data science, e.g., the design of scientific work-flows [139,140] or the generation of model simulations [141].

Chapter 7

Conclusion and Future Directions

This thesis introduced the problem of learning predictive models with heterogeneity in populations of data instances. This problem commonly arises in several real-world applications of predictive learning where the underlying systems are comprised of multiple data populations. Some examples include the heterogeneity in populations of human subjects for medical diagnosis, the heterogeneity in observations of spatio-temporal variables across space and time, and the heterogeneity in characteristics of user interactions on social networking websites. In the presence of population heterogeneity, a central challenge is that the training data comprises of instances belonging from multiple populations, and the instances in the test set may be from a different population than that of the training instances. This limits the effectiveness of standard predictive learning frameworks that are based on assumptions of population homogeneity, which are ideally true only in simplistic settings.

A number of methods have been developed for addressing population heterogeneity in predictive learning problems, although as isolated efforts in disparate applications, lacking a concerted focus for a common objective. This thesis provided an over-arching structure to the existing body of work on predictive learning with population heterogeneity, by building a common taxonomy for reviewing existing efforts in Chapter 2. In particular, it introduced the concepts of explicit and implicit context of data instances in different application settings, which can be used for inferring the nature of predictive relationships at every instance in the presence of population heterogeneity.

This thesis presented several ways of using explicit as well as implicit context of data

instances in predictive learning frameworks, for addressing the challenges associated with population heterogeneity. It introduced a novel multi-task learning framework in Chapter 3 for problems where we have access to some ancillary variables that can be grouped using clustering methods to produce homogeneous partitions of data instances, thus addressing the challenge of population heterogeneity. This thesis also introduced a novel strategy for constructing ensembles in binary classification settings in Chapter 4, using information about the multi-modal structure of both classes arising due to the heterogeneity in their populations. When the context of data instances is implicitly defined such that the test data is known to comprise of contextually similar groups, this thesis presented a novel framework for adapting classification decisions using the group-level properties of test instances in Chapter 5, in the absence of incremental labels.

An underlying theme of research in this thesis has been to incorporate physical knowledge of the application domain in predictive learning frameworks, for addressing the challenge of population heterogeneity. This thesis introduced a novel paradigm of knowledge discovery in Chapter 6, termed as theory-guided data science, that aims to pursue the broader goal of systematically integrating scientific knowledge, which is often encoded as physics (or theory) based models, in data science frameworks. This thesis builds the foundations of this emerging paradigm by reviewing a variety of ways scientific knowledge can be combined with data science methods, which are gaining prominence in diverse scientific and engineering disciplines.

There are several research directions in the rapidly advancing field of data science that are enabled by the contributions presented in this thesis. First, the multi-task learning framework presented in Chapter 3 can be generalized to problems where clustering the ancillary variables may not be straight-forward. This will be especially useful for problems where the ancillary variables are available in network representations (e.g., in social data mining problems), or are extremely high-dimensional (e.g., genetic profiles in bioinformatics problems). Second, instead of treating the clustering of ancillary variables and the learning of predictive models at every cluster as two independent tasks, joint frameworks that simultaneously create homogeneous partitions of data instances and learn the predictive relationships at every partition need to be explored. Third,

although the mode-specific ensembles presented in Chapter 4 have been shown to empirically provide better predictive performance than traditional ensemble learning methods, a deeper theoretical analysis of its strengths and limitations needs to be investigated. Fourth, in the problem of adapting the responses of mode-specific ensembles using the group-level properties of test instances, the effect of noise in the training and test data, presence of irrelevant attributes, and imbalance among the modes within a class (or across different classes) needs to be thoroughly investigated. More advanced weighting schemes than those presented in Chapter 5 can also be explored. In particular, we can consider to use the statistical distance between the distribution of instances in a test group and the subset of modes used for training an ensemble classifier for deciding its adaptive weight. We can also explore weighting strategies that use priors on the joint likelihood of subsets of modes to prune the set of ensemble classifiers relevant for a group of test instances.

Finally, the paradigm of theory-guided data science, presented in Chapter 6 is ripe with possibilities of future directions in this emerging field of research. We hope that this thesis serves as a first step in building the foundations of theory-guided data science and encourages follow-on work to develop in-depth theoretical formalizations of this paradigm. While success in this endeavor will need significant innovations in our ability to handle the diversity of forms in which scientific knowledge is represented and ingested in different disciplines (e.g., differences in granularity and type of information, degree of completeness, and uncertainty in knowledge), the concrete research approaches presented in this thesis can be considered as a stepping stone in this ambitious journey. We anticipate the deep integration of theory-based and data science to become a quintessential tool for scientific discovery in future research. The paradigm of theory-guided data science, if effectively utilized, can help us realize the vision of the "fourth paradigm" [142] in its full glory, where data serves an integral role at every step of scientific knowledge discovery.

References

[1] Gordon Bell, Tony Hey, and Alex Szalay. Beyond the data deluge. *Science*, 323(5919):1297–1298, 2009.

[2] Economist. The data deluge. *Special Supplement*, 2010.

[3] Manyika James, Chui Michael, Brown Brad, Bughin Jacques, D Richard, R Charles, and HB Angela. Big data: The next frontier for innovation, competition, and productivity. *The McKinsey Global Institute*, 2011.

[4] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.

[5] P.N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. *Introduction to data mining (Second Edition)*. Pearson Addison Wesley, 2017.

[6] Anuj Karpatne, Zhe Jiang, Ranga Raju Vatsavai, Shashi Shekhar, and Vipin Kumar. Monitoring land-cover changes: A machine-learning perspective. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):8–21, 2016.

[7] Anuj Karpatne, Ankush Khandelwal, Shyam Boriah, and Vipin Kumar. Predictive learning in the presence of heterogeneity and limited training data. In *SDM*, pages 253–261. SIAM, 2014.

[8] Anuj Karpatne, Ankush Khandelwal, and Vipin Kumar. Ensemble learning methods for binary classification with multi-modality within the classes. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 730–738. SIAM, 2015.

[9] Anuj Karpatne and Vipin Kumar. Adaptive heterogeneous ensemble learning using the context of test instances. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 787–792. IEEE, 2015.

[10] Anuj Karpatne, Gowtham Atluri, James Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[11] Anuj Karpatne, Mace Blank, Michael Lau, Shyam Boriah, Karsten Steinhaeuser, Michael Steinbach, and Vipin Kumar. Importance of vegetation type in forest cover estimation. In *Intelligent Data Understanding (CIDU), 2012 Conference on*, pages 71–78. IEEE, 2012.

[12] Sudipto Banerjee, Bradley P Carlin, and Alan E Gelfand. *Hierarchical modeling and analysis for spatial data*. Crc Press, 2014.

[13] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[14] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In *Advances in neural information processing systems*, pages 409–415, 2001.

[15] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[16] Mira Bernstein, Vin De Silva, John C Langford, and Joshua B Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Department of Psychology, Stanford University, 2000.

[17] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2005.

[18] Matthew Blaschko and Christoph Lampert. Learning to localize objects with structured output regression. *Computer Vision–ECCV 2008*, pages 2–15, 2008.

[19] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[20] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2:3, 2006.

[21] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *ACM SIGKDD*, pages 109–117. ACM, 2004.

[22] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615, 2006.

[23] Daniel Sheldon. Graphical multi-task learning. 2008.

[24] Laurent Jacob, Francis Bach, and Jean-Philippe Vert. Clustered multi-task learning: A convex formulation. *arXiv preprint arXiv:0809.2085*, 2008.

[25] Peter MacCullagh and John Ashworth Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

[26] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[27] Carl Erik Froberg and Carl Erik Frhoberg. *Introduction to numerical analysis*. Addison-Wesley Reading, Massachusetts, 1969.

[28] A. Karpatne, M. Blank, M. Lau, S. Boriah, K. Steinhaeuser, M. Steinbach, and V. Kumar. Importance of vegetation type in forest cover estimation. *CIDU*, 2012.

[29] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[30] Thijs T. van Leeuwen, Andrew J. Frank, Yufang Jin, Padhraic Smyth, Michael L. Goulden, Guido R. van der Werf, and James T. Randerson. Optimal use of land surface temperature data to detect changes in tropical forest cover. *Journal of Geophysical Research*, 116, 2011.

[31] Gilberto Câmara, Dalton de Morisson Valeriano, and João Vianei Soares. Metodologia para o cálculo da taxa anual de desmatamento na Amazônia legal. São José dos Campos, INPE, 2006.

[32] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[33] Dmitriy Fradkin. Clustering inside classes improves performance of linear classifiers. In *ICTAI'08*, volume 2, pages 439–442. IEEE, 2008.

[34] Nathalie Japkowicz. Supervised learning with unsupervised output separation. In *International Conference on Artificial Intelligence and Soft Computing*, volume 3, pages 321–325, 2002.

[35] Thomas G Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2(263):286, 1995.

[36] Eun Bae Kong and Thomas G Dietterich. Error-correcting output coding corrects bias and variance. In *ICML*, pages 313–321, 1995.

[37] Ludmila I Kuncheva. Using diversity measures for generating error-correcting output codes in classifier ensembles. *Pattern Recognition Letters*, 26(1):83–90, 2005.

[38] Oriol Pujol, Petia Radeva, and Jordi Vitria. Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1007–1012, 2006.

[39] Trevor Hastie, Robert Tibshirani, et al. Classification by pairwise coupling. *The annals of statistics*, 26(2):451–471, 1998.

[40] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.

[41] Gavin Brown. Ensemble learning. In *Encyclopedia of Machine Learning*, pages 312–320. Springer, 2010.

[42] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[43] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

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

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

[46] Ricardo Vilalta and Irina Rish. A decomposition of classes via clustering to explain and improve naive bayes. In *Machine Learning: ECML 2003*, pages 444–455. Springer, 2003.

[47] Nisar Ahmed and Mark Campbell. On estimating simple probabilistic discriminative models with subclasses. *Expert Systems with Applications*, 39(7):6659–6664, 2012.

[48] Koby Crammer and Yoram Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2-3):201–233, 2002.

[49] Francesco Masulli and Giorgio Valentini. Effectiveness of error correcting output codes in multiclass learning problems. In *Multiple Classifier Systems*, pages 107–116. Springer, 2000.

[50] Charles J Vörösmarty, Pamela Green, Joseph Salisbury, and Richard B Lammers. Global water resources: vulnerability from climate change and population growth. *Science*, 289(5477):284–288, 2000.

[51] Land Processes Distributed Active Archive Center. `http://lpdaac.usgs.gov`.

[52] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, 6(1):40–49, 2004.

[53] Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. Class imbalances versus class overlapping: an analysis of a learning system behavior. In *MICAI 2004: Advances in Artificial Intelligence*, pages 312–321. Springer, 2004.

[54] Anuj Karpatne, Ankush Khandelwal, and Vipin Kumar. Ensemble learning methods for binary classification with multi-modality within the classes. In *SDM*, 2015.

[55] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[56] Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992.

[57] Vladimir Vapnik and Léon Bottou. Local algorithms for pattern recognition and dependencies estimation. *Neural Computation*, 5(6):893–909, 1993.

[58] Thorsten Joachims et al. Transductive learning via spectral graph partitioning. In *ICML*, volume 3, pages 290–297, 2003.

[59] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in neural information processing systems*, pages 585–592, 2002.

[60] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.

[61] Jeremy Z Kolter and M Maloof. Dynamic weighted majority: A new ensemble method for tracking concept drift. In *ICDM*, pages 123–130, 2003.

[62] Leandro L Minku, Allan P White, and Xin Yao. The impact of diversity on online ensemble learning in the presence of concept drift. *TKDE*, 22(5):730–742, 2010.

[63] Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2):577–584, 2005.

[64] Davide Castelvecchi et al. Artificial intelligence called in to tackle lhc data deluge. *Nature*, 528(7580):18–19, 2015.

[65] Pierre Baldi and Søren Brunak. *Bioinformatics: the machine learning approach*. MIT press, 2001.

[66] James H. Faghmous and Vipin Kumar. A Big Data Guide to Understanding Climate Change: The Case for Theory-Guided Data Science. *Big Data*, 3, 2014.

[67] James H Faghmous, Vipin Kumar, and Shashi Shekhar. Computing and climate. *Computing in Science & Engineering*, 17(6):6–8, 2015.

[68] D Graham-Rowe, D Goldston, C Doctorow, M Waldrop, C Lynch, F Frankel, R Reid, S Nelson, D Howe, SY Rhee, et al. Big data: science in the petabyte era. *Nature*, 455(7209):8–9, 2008.

[69] TO Jonathan, AM Gerald, et al. Special issue: dealing with data. *Science*, 331(6018):639–806, 2011.

[70] Terrence J Sejnowski, Patricia S Churchland, and J Anthony Movshon. Putting big data to good use in neuroscience. *Nature neuroscience*, 17(11):1440–1441, 2014.

[71] Chris Anderson. The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. *Wired Magazine*, 2008.

[72] Peter M Caldwell, Christopher S Bretherton, Mark D Zelinka, Stephen A Klein, Benjamin D Santer, and Benjamin M Sanderson. Statistical significance of climate sensitivity predictors obtained by data mining. *Geophysical Research Letters*, 41(5):1803–1808, 2014.

[73] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The Parable of Google Flu: Traps in Big Data Analysis. *Science (New York, N.Y.)*, 343(6176):1203–5, March 2014.

[74] Gary Marcus and Ernest Davis. Eight (no, nine!) problems with big data. *The New York Times*, 6(04):2014, 2014.

[75] Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2009.

[76] Jaya Kawale, Stefan Liess, Arjun Kumar, Michael Steinbach, Peter Snyder, Vipin Kumar, Auroop R Ganguly, Nagiza F Samatova, and Fredrick Semazzi. A graph-based approach to find teleconnections in climate data. *Statistical Analysis and Data Mining*, 6(3):158–179, 2013.

[77] James H Faghmous, Ivy Frenger, Yuanshun Yao, Robert Warmka, Aron Lindell, and Vipin Kumar. A daily global mesoscale ocean eddy dataset from satellite altimetry. *Scientific data*, 2, 2015.

[78] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine learning-augmented predictive modeling of turbulent separated flows over airfoils. *arXiv preprint arXiv:1608.03990*, 2016.

[79] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning for predictive turbulence modeling: Using data to improve rans modeled reynolds stresses. *arXiv preprint arXiv:1606.07987*, 2016.

[80] Geoffroy Hautier, Christopher C Fischer, Anubhav Jain, Tim Mueller, and Gerbrand Ceder. Finding natures missing ternary oxide compounds using machine learning and density functional theory. *Chemistry of Materials*, 22(12):3762–3767, 2010.

[81] Christopher C Fischer, Kevin J Tibbetts, Dane Morgan, and Gerbrand Ceder. Predicting crystal structure by merging data mining with quantum mechanics. *Nature materials*, 5(8):641–646, 2006.

[82] Stefano Curtarolo, Gus LW Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. The high-throughput highway to computational materials design. *Nature materials*, 12(3):191–201, 2013.

[83] Li Li, John C Snyder, Isabelle M Pelaschier, Jessica Huang, Uma-Naresh Niranjan, Paul Duncan, Matthias Rupp, Klaus-Robert Müller, and Kieron Burke. Understanding machine-learned density functionals. *International Journal of Quantum Chemistry*, 2015.

[84] Ken CL Wong, Linwei Wang, and Pengcheng Shi. Active model with orthotropic hyperelastic material for cardiac image analysis. In *Functional Imaging and Modeling of the Heart*, pages 229–238. Springer, 2009.

[85] Jingjia Xu, John L Sapp, Azar Rahimi Dehaghani, Fei Gao, Milan Horacek, and Linwei Wang. Robust transmural electrophysiological imaging: Integrating sparse and dynamic physiological models into ecg-based inference. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 519–527. Springer, 2015.

[86] Jin Liu, Kai Wang, Shuangge Ma, and Jian Huang. Accounting for linkage disequilibrium in genome-wide association studies: a penalized regression method. *Statistics and its interface*, 6(1):99, 2013.

[87] Ankush Khandelwal, Varun Mithal, and Vipin Kumar. Post classification label refinement using implicit ordering constraint among data instances. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 799–804. IEEE, 2015.

[88] Ankush Khandelwal, Anuj Karpatne, Miriam Marlier, Julia Kim, Dennis Lettenmaier, and Vipin Kumar. An approach for global monitoring of surface water extent variations using modis data. In *Remote Sensing of Environment (in review)*, 2017.

[89] Nicholas Wagner and James M Rondinelli. Theory-guided machine learning in materials science. *Frontiers in Materials*, 3:28, 2016.

[90] James Faghmous, Arindam Banerjee, Shashi Shekhar, Michael Steinbach, Vipin Kumar, Auroop R. Ganguly, and Nagiza Samatova. Theory-guided data science for climate change. *Computer*, 47(11):74–78, 2014.

[91] A. R. Ganguly, E. A. Kodra, A. Agrawal, A. Banerjee, S. Boriah, Sn. Chatterjee, So. Chatterjee, A. Choudhary, D. Das, J. Faghmous, P. Ganguli, S. Ghosh, K. Hayhoe, C. Hays, W. Hendrix, Q. Fu, J. Kawale, D. Kumar, V. Kumar, W. Liao, S. Liess, R. Mawalagedara, V. Mithal, R. Oglesby, K. Salvi, P. K. Snyder, K. Steinhaeuser, D. Wang, and D. Wuebbles. Toward enhanced understanding and projections of climate extremes using physics-guided data mining techniques. *Nonlinear Processes in Geophysics*, 21(4):777–795, 2014.

[92] *Physics Informed Machine Learning Conference*, Santa Fe, New Mexico, 2016.

[93] Physical analytics, ibm research. http://researcher.watson.ibm.com/researcher/view_group.php?id=6566. Accessed: 2016-10-20.

[94] Mehdi Ghasemizade and Mario Schirmer. Subsurface flow contribution in the hydrological cycle: lessons learned and challenges aheada review. *Environmental earth sciences*, 69(2):707–718, 2013.

[95] Claudio Paniconi and Mario Putti. Physically based modeling in catchment hydrology at 50: Survey and outlook. *Water Resources Research*, 51(9):7090–7129, 2015.

[96] Marc FP Bierkens. Global hydrology 2015: State, trends, and directions. *Water Resources Research*, 51(7):4923–4947, 2015.

[97] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[98] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Intorduction to Data Mining*. Addison-Wesley, 2005.

[99] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[100] Michael D Twa, Srinivasan Parthasarathy, Cynthia Roberts, Ashraf M Mahmoud, Thomas W Raasch, and Mark A Bullimore. Automated decision tree classification of corneal shape. *Optometry and vision science: official publication of the American Academy of Optometry*, 82(12):1038, 2005.

[101] Joel Z. Leibo, Qianli Liao, Fabio Anselmi, Winrich A. Freiwald, and Tomaso Poggio. View-tolerant face recognition and hebbian learning imply mirror-symmetric neural tuning to head orientation. *Current Biology*, 2016.

[102] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[103] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, pages 338–342, 2014.

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[105] Franziska Schrodt, Jens Kattge, Hanhuai Shan, Farideh Fazayeli, Julia Joswig, Arindam Banerjee, Markus Reichstein, Gerhard Bönisch, Sandra Díaz, John Dickie, et al. Bhpmf–a hierarchical bayesian approach to gap-filling and trait prediction for macroecology and functional biogeography. *Global Ecology and Biogeography*, 24(12):1510–1521, 2015.

[106] Jens Kattge, Sandra Diaz, Sandra Lavorel, IC Prentice, Paul Leadley, Gerhard Bönisch, Eric Garnier, Mark Westoby, Peter B Reich, IJ Wright, et al. Try–a global database of plant traits. *Global change biology*, 17(9):2905–2935, 2011.

[107] Prem Melville and Vikas Sindhwani. Recommender systems. In *Encyclopedia of machine learning*, pages 829–838. Springer, 2011.

[108] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[109] Huseyin Denli, Niranjan Subrahmanya, et al. Multi-scale graphical models for spatio-temporal processes. In *Advances in Neural Information Processing Systems*, pages 316–324, 2014.

[110] Jean-Francois Boulicaut and Baptiste Jeudy. Constraint-based data mining. In *Data Mining and Knowledge Discovery Handbook*, pages 399–416. Springer, 2005.

[111] Jian Pei and Jiawei Han. Constrained frequent pattern mining: a pattern-growth view. *ACM SIGKDD Explorations Newsletter*, 4(1):31–39, 2002.

[112] Sugato Basu, Ian Davidson, and Kiri Wagstaff. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press, 2008.

[113] Andrew J Majda and John Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26(1):201, 2012.

[114] Andrew J Majda and Yuan Yuan. Fundamental limitations of ad hoc linear and quadratic multi-level regression models for physical systems. *Discrete and Continuous Dynamical Systems B*, 17(4):1333–1363, 2012.

[115] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

[116] Anuj Karpatne, Ankush Khandelwal, Xi Chen, Varun Mithal, James Faghmous, and Vipin Kumar. Global monitoring of inland water dynamics: state-of-the-art, challenges, and opportunities. In *Computational Sustainability*, pages 121–147. Springer, 2016.

[117] Gareth M James and Peter Radchenko. A generalized dantzig selector with shrinkage tuning. *Biometrika*, 96(2):323–337, 2009.

[118] Soumyadeep Chatterjee, Sheng Chen, and Arindam Banerjee. Generalized dantzig selector: Application to the k-support norm. In *Advances in Neural Information Processing Systems*, pages 1934–1942, 2014.

[119] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[120] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM, 2009.

[121] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 543–550, 2010.

[122] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

[123] Soumyadeep Chatterjee, Karsten Steinhaeuser, Arindam Banerjee, Snigdhansu Chatterjee, and Auroop R Ganguly. Sparse group lasso: Consistency and climate applications. In *SDM*, pages 47–58. SIAM, 2012.

[124] Varun Mithal, Ankush Khandelwal, Shyam Boriah, Karsten Steinhaeuser, and Vipin Kumar. Change detection from temporal sequences of class labels: application to land cover change mapping. In *SIAM International Conference on Data mining, Austin, TX, USA*, pages 2–4. Citeseer, 2013.

[125] Xiaowei Jia, Ankush Khandelwal, James Gerber, Kimberly Carlson, Paul West, Leah Samberg, and Vipin Kumar. Automated plantation mapping in southeast asia using remote sensing data. Technical Report 16-029, Department of Computer Science, University of Minnesota, Twin Cities, 2016.

[126] Xiaowei Jia, Ankush Khandelwal, Nayak Guru, James Gerber, Kimberly Carlson, Paul West, and Vipin Kumar. Predict land covers with transition modeling and incremental learning. In *SIAM International Conference on Data Mining (accepted)*, 2017.

[127] Robert L Wilby, TML Wigley, D Conway, PD Jones, BC Hewitson, J Main, and DS Wilks. Statistical downscaling of general circulation model output: a comparison of methods. *Water resources research*, 34(11):2995–3008, 1998.

[128] Peter Sadowski, David Fooshee, Niranjan Subrahmanya, and Pierre Baldi. Synergies between quantum mechanics and machine learning in reaction prediction. *Journal of Chemical Information and Modeling*, 56(11):2125–2128, 2016.

[129] Eric J Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016.

[130] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[131] Keith Beven and Andrew Binley. The future of distributed models: model calibration and uncertainty prediction. *Hydrological processes*, 6(3):279–298, 1992.

[132] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[133] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[134] MI Jordan and TM Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[135] Rajeev Agrawal. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951, 1995.

[136] Robert D Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704, 2004.

[137] J. H. Faghmous, L. Styles, V. Mithal, S. Boriah, S. Liess, F Vikebo, M. d. S. Mesquita, and V. Kumar. Eddyscan: A physically consistent ocean eddy monitoring application. In *Intelligent Data Understanding (CIDU), 2012 Conference on*, pages 96 –103, oct. 2012.

[138] James H. Faghmous, Hung Nguyen, Matthew Le, and Vipin Kumar. Spatio-temporal consistency as a means to identify unlabeled objects in a continuous data field. In *AAAI*, pages 410–416, 2014.

[139] Vasant G Honavar. The promise and potential of big data: A case for discovery informatics. *Review of Policy Research*, 31(4):326–330, 2014.

[140] Yolanda Gil, Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geoffrey Fox, Dennis Gannon, Carole Goble, Miron Livny, Luc Moreau, and Jim Myers. Examining the challenges of scientific workflows. *Ieee computer*, 40(12):26–34, 2007.

[141] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. Calogan: Simulating 3d high energy particle showers in multi-layer electromagnetic calorimeters with generative adversarial networks. *arXiv preprint arXiv:1705.02355*, 2017.

[142] Tony Hey, Stewart Tansley, Kristin M Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.