ABSTRACT

RAVINDRAN, SRINATH. Learning Rare Patterns with Multilevel Models. (Under the direction of Dennis Bahler.)

In applications of machine learning as diverse as computer vision, information retrieval, text classification, chemoinformatics, and bioinformatics, a variety of common issues have been identified involving frequency of occurrence, variation and similarities of examples, and lack of examples. These issues continue to be important hurdles in machine intelligence and this work focuses on developing robust machine learning models that address the same. In particular, the approaches we discuss in this work fall under the general framework of Multilevel Models. Recent research has shown that multilevel and hierarchical models are well suited for finding solutions to learning in the presence of complex data.

We are interested in utilizing the power of multilevel models to solve the problem of learning in the presence of a small number of examples. Unsupervised learning algorithms such as k-means clustering and Dirichlet process mixture models can be used to identify similarity between the data points and form meaningful clusters from them. Such algorithms can also be used to place the rare example in appropriate groups, and such an assignment can be used further to learn the concept.

Often studied as a part of outlier detection and imbalanced class learning, the problem of learning with a small number of examples is important and has yet to be fully understood. As a first step in towards understanding the learning problem, we discuss a generalized definition of rareness. Unlike existing treatments of rareness or class imbalance which are based on frequency of occurrence of patterns, our definition uses both frequency of occurrence and the relationship between the patterns in the dataset to define

rareness.

In our work, we focus on improving predictive performance when learning with less data. This problem has attracted attention in recent times. Lack of data manifests itself into various forms including rareness, imbalance and sparsity, and each of these affect learning in different ways. For the purposes of this work, we highlight how multilevel models can be used to solve both regression and classification problems. We discuss how the similarity between examples in the data can be utilized to effectively learn rare, but potentially important new concepts. We also discuss how our model can be used even with extreme cases of rareness, such as in a one-shot learning problem. We demonstrate the extension of our classification model using the case study of hand gesture recognition.

Overall, the multilevel models presented in this work outperform traditional approaches in learning tasks involving rare patterns, as demonstrated on datasets of varying complexities.

Learning Rare Patterns with Multilevel Models

by
Srinath Ravindran

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2017

APPROVED BY:

Jon Doyle	James Lester II
Munindar Singh	Dennis Bahler Chair of Advisory Committee

DEDICATION

To my parents, Subhadra and Ravindran.

BIOGRAPHY

Srinath Ravindran was born in Coimbatore, a city in the state of Tamil Nadu, India. He grew up in Chennai, the best city in the world, and also the state capital of Tamil Nadu. He finished high school at Bala Vidya Mandir, Chennai and received his Bachelor of Engineering degree in Computer Science and Engineering from Anna University, Chennai in 2007. He received a Master of Science degree in Computer Science from North Carolina State University (NCSU), Raleigh, in 2009.

Srinath's research interests include machine learning, computer vision, knowledge representation and reasoning, and human-computer interaction. While still a graduate student, he taught three undergraduate courses on Artificial Intelligence, Introduction to Programming and Discrete Mathematics.

ACKNOWLEDGEMENTS

I take this opportunity to express my gratitude to the people who directly or indirectly contributed to the completion of this thesis.

First, I would like to thank my advisor, Dr. Dennis Bahler for his guidance, support, and patience. He encouraged me to work on interesting projects in addition to my dissertation, pursue internships, and teach courses. Next, I would like to thank my committee members, Dr. Jon Doyle, Dr. James Lester, and Dr. Munindar Singh. I am grateful that they all agreed to be part of my doctoral committee.

Dr. Thuente, Dr. Reeves, and Dr. Rouskas served as the Directors of Graduate program during my years in grad school, and I thank them for their guidance, and encouragement. I would be remiss if I do not thank Dr. Christopher Healey and Dr. Robert St. Amant. I have worked with them on several occasions, and have learned a lot through the interactions.

I have spent a lot of time at the Knowledge Discovery Lab to an extent that it was my second home. I'll always remember the late nights, open houses, presentations, broken refrigerators, video games, cricket world cup, and more. My stay in the lab was made memorable by a lot of people: Thomas, Lloyd, Marivic, Kyung Wha Hong, Lihua Hao, Joe Hsiao, Andrew Wicker, Shea McIntee, Karthik, Nazli, Prairie Rose, and Adam Marrs. I extend a special thanks to my grad school friends: Reuben, Arun Prakash, Alex Jamestin, Supritam Sen, Arpan, Trisha, Sina Bahram, Hilay, Kalpesh, Lifford, Vineha, and Pradeep Murukanaiah.

I am grateful to Dr. David Aha for being my mentor at the AAAI doctoral consortium.

I am deeply grateful to my family for all their support, encouragement, sacrifice, and belief in my ability.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

There are a variety of machine learning approaches that work well for problems involving independently and identically distributed (IID) data, and cases with enough examples to represent the variations in data. However, in many practical applications including computer vision, text classification, bioinformatics, and others, we may not have sufficient examples, the data may not be drawn from identical distributions, or there could be complex relations among individual data points.

These problems that make the learning task difficult and often error prone typically fall under the following categories:

1. *Within-class Variation* and *Inter-class Similarity*: examples within one category or class have different properties, whereas examples from different categories or classes may have similar properties.

2. *Lack of Good Quality Examples* and *Rare Examples*: examples that either belong to an unknown class or are assigned an imprecise class label.

3. *Rare Example*: examples may be the only representatives of their kind and could

be present in either training or test data.

Over the past few years, a variety of approaches have been developed to address each of the issues mentioned above. Existing approaches to prediction most often do not consider these problems together, instead treating these as separate problems. However, there are many applications having some combination of these problems.

An example application is chemical toxicity prediction. Chemicals belong to various congeneric classes such as acids, alcohols and amines. Often the chemicals in one class exhibit properties similar to chemicals in another congeneric class while being different from chemicals in the same class. The former phenomenon is an example of *within-class variation* and the latter *inter-class similarity*. Moreover, the occurrence of some chemicals may be *rare or unseen*. That is, the chemical may be the only representative of its class in the training data, or a chemical that is not be present training set could be presented at a later stage (test set) for a prediction task. Finally, the labels assigned to each of the chemicals are determined experimentally and this task is costly and, as often in other applications, error prone.

Our research and the recent literature lead us to conclude that multilevel and hierarchical models are well suited for finding solutions to learning in the presence of complex data. We study these models in the context of the aforementioned complexities in data. In our research so far, we have studied and developed approaches based on multilevel models and have obtained promising results compared to more traditional models. However, there are a variety of unsolved problems and we address some of those concerns in this work.

1.1 Terms and Definitions

We will use several key words or phrases across all the chapters in this document; we present them in this section. We have tried to make the usage of terms consistent across chapters, while not departing from some of the conventions used in the literature.

In machine learning problems, we are presented with a data set, we would like to learn from. For instance, in order to learn to recognize hand-written digits, the dataset will contain several images of digits 0-9, each written by a person with pen and paper. Some examples of such a dataset include MNIST [42] and the Hand-written digit recognition datasets [5]. Every image of each digit is called a data point, instance, an observation, or an example. In our work we will use the term *example* to collectively represent these terms. Generally, each dataset D has a finite set of such examples, and the number of examples is denoted by N. The number of examples in a set of examples is also denoted by $|D|$. A subset of examples that exhibit common or similar features are referred to as a pattern.

For the problem of recognizing hand-written digits, the learning algorithm must learn the digit based on the examples in the dataset. Examples of each digit share some visual similarities with each other, and at the same time are visually different from the other digits. The visual similarities and differences are represented by features or attributes. In case of the images representing digits, the line segments that make up the outline of the digits are one of the features. If we consider the set of examples representing the number 4, people have different ways of writing the number, and each of those variations constitute a pattern.

When predicting a new image of a digit, the machine learning algorithm, assigns a label corresponding to the digit that was recognized. In such classifications problems, the

labels are referred to as *Class Labels*, and the images of a certain digit, say 5, are said to belong to class 5. The literature uses the terms classes and categories interchangeably, and we will use the term *class* throughout the document.

More formally, let us assume that we are given training data $\mathbb{T}$ consisting of N examples, where each example x_i has a label y_i. The set of examples is represented as $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. The learning task is then to produce a classifier h: $X \rightarrow Y$ which maps an object $x \in X$ to its class label $y \in Y$. Each example, x_i is often represented by a $\mathbb{D}$-dimensional feature vector $x = (x_{i1}, \ldots, x_{i\mathbb{D}}) \in \mathbb{R}^{\mathbb{D}}$. Each dimension in the feature space represents a feature or attribute, and x_i is a point in the feature space.

It is important to note that the dataset represents only a sample of examples, and not the entire population. The dataset, therefore, will not reflect the population nor mimic the distribution of examples in the population. For instance, while the digit zero (0) might occur more frequently than other digits in real life, the data set could consist of the same number of examples for each digit.

1.2 Our Work

This research aims to study the following:

1. Learning in the presence of a small number of examples

2. Feature representation techniques to aid learning in complex scenarios

Learning in the presence of small number of examples is an important problem. While there is existing previous work in the the area, there are many issues that require attention. In many learning scenarios, obtaining sufficient examples is hard either because

of the nature of the problem or the cost involved in gathering them. These examples occurring with low frequency could be

1. A new sub-class of a known class
2. The first occurrence of a newly discovered class
3. The only known examples of some class

Each of these problems is commonly studied under topics such as rare examples [62], class imbalance [35], transfer learning [50], and dataset shift [56]. We discuss the relationship among these problem in Section 2.2. We also briefly discuss the effects of rare data and small number of examples in the context of transfer learning and hierarchical models [61]. We have identified ways in which rare data can affect performance in transfer learning and hierarchical models.

These problems are not relevant only in traditional machine learning problems but also within the current trend of "Big-Data". While big-data problems in general deal with handling large amounts of data, the aforementioned problems are still present.

In the context of big-data, one might also have come across the phrase "needle in the haystack" used as an umbrella term to describe all these problems. We feel that the term does not do justice towards helping the community understand the problems. This is just the tip of the iceberg and there are a variety of such misused and misunderstood terms that affect further research. One of the contributions of this work is toward resolving these issues. We continue this discussion in Chapter 2.

Next, we focus on solving the specific category of examples, *Rare Examples*. The term "rare" is another misused term, which we define in Chapter 2. We discuss the relationship of rare examples with other related problems. Then, we describe two approaches: Simi-

larity Based Multilevel Model (SBMM) for Regression in Chapter 4, and Classification Models for Learning Rare Data in Chapter 5.

SBMM is a multilevel regression model that uses the similarity among data points to effectively predict the target values for a new example. Data points within a cluster or within the same neighborhood are considered similar to each other. The approach is based on the assumption that any datapoint has some degree of similarity to some subset of previously known data points.

The Similarity Based Classification Model in Chapter 5, like the SBMM, uses the similarity among the data points to classify them. When presented with a rare example, the model either assigns a new class label, or an existing class label, based on how similar the example is from the previously learned examples.

Both the models work well when there is sufficient background knowledge. Once a model is pre-trained with labeled examples, it is presented with a new example to learn. As we will show in the respective chapters, the models are not only able to learn the new example, but also predict future examples with high accuracy.

While describing the models, we also discuss the problem of feature selection in the context of learning in the presence of small number of examples. A good choice of features and representation can lead to improved predictive performance with complex data. Obtaining a set of invariant features that can enable robust learning, but in many task situations, identifying a set of invariant features is difficult. Moreover, features may fall into several types, including temporal, spacial, graphical and other structured input. It is essential to develop approaches that can handle such a variety of features while learning with complex data. Further, utilizing multiple types of features can improve performance in many AI tasks.

Chapter 2

Learning with a Small Number of Examples

2.1 Introduction

Traditional machine learning assumes the availability of a large number of examples for every possible class involved in the learning problem. The crux of our work lies in improving predictive performance when the data contains a small number of examples. Learning in such settings has attracted attention in recent times for a variety of reasons.

1. Cognitive: Humans have the ability to learn or generalize concepts with as few as one example, given sufficient background knowledge.

2. Practical: Gathering examples involves scientific experimentation, exploration of the environment, etc.

3. Nature of the problem: Some examples occur more frequently than others as an artifact of the problem domain.

While this list is not comprehensive, it provides good motivation to study this topic. Lack of data can be perceived in various forms including rareness, imbalance, missing data and sparsity, and each of these affect learning in different ways. This problem is still relevant in the era of *big data*.

In this chapter, we discuss different types of learning scenarios involving a small number of examples. We provide insights from existing literature and present our findings as well. In the process, we resolve conflicting terminologies. For example, there is a significant misunderstanding of some concepts, including the use of the term "rare". Sometimes rare data is used to mean class imbalance, while sometimes it is used to refer to examples that occur infrequently.

We also discuss the relationship among concepts in machine learning such as data shift, imbalance, rareness and missing data.

2.2 Small Number of Examples

Learning in the presence of a small number of examples is an important problem and there are many issues that require attention. In many learning scenarios, obtaining sufficient examples is hard either because of the nature of the problem or the cost involved in gathering them. These examples occurring with low frequency could be

1. A new sub-class of a known class
2. The first occurrence of a newly discovered class
3. The only known example(s) of some class

The problem of learning with small number of examples can be characterized using the following aspects.

1. **Number of Examples:** Based on the number or proportion of examples in the training set.

   In some problems, the number or proportion of examples in the training set can be different for different classes. One very common example is that of *Class Imbalance*, where given two classes C_1 and C_2, the number of examples of C_1, represented as $|C_1|$, can be significantly more than the number of examples of class C_2, represented as $|C_2|$. Such datasets are also called skewed datasets. Class imbalance can be generalized to multi-class classification problems.

   Apart from class imbalance, lack of examples in the training set presents itself in other forms such as rare examples and missing information.

   An extreme problem in this category is *One-shot Learning*, where we learn a new class using only one example, given sufficient background knowledge.

2. **Training Phase vs Test Phase:** Based on the existence of examples in the training set or test/ classification set.

   A lack of examples or presence of a rare example can exist either during training phase or the test phase. Ideally, one would expect statistical models to be capable of learning concepts even from under-represented classes and also to predict unseen examples with great accuracy when the model is used. In reality, however, this is harder to achieve.

   Lack of training examples can also be studied from the standpoint of class imbalance. Presence of rare examples in the test is a challenging problem too. The rare example is more likely to be misclassified either because the example could resemble data from a different class more than other examples from the same class,

or there may not be enough preexisting knowledge about the class of the example being considered.

Achieving a good balance between bias and variance at training time is often the primary weapon of choice when dealing with lack of examples. There is no one definite solution as this problem is domain and problem dependent.

3. **Feature Space:** Based on differences in the features and values of the attributes among examples.

   Datasets with complete data are easier to learn from compared to complex or incomplete datasets. Often, the complexity in the dataset is due to the feature set. The choice of features, lack of values for the features and variance in the values of the features are all typical reasons for the complexity.

   In several learning scenarios, the choice of features is a major problem. Ease of collecting data, cost effectiveness of some features, and lack of sufficient domain knowledge can lead to poor choice of a feature set. Such a choice can worsen the learning problem in an already skewed dataset.

2.2.1 Class Imbalance

In many real-world applications the problem of learning from imbalanced data has attracted growing attention from both academia and industry. Learning with imbalanced data is concerned with learning algorithms that work in the presence of underrepresented data and class distribution skews. Any data set that exhibits an unequal distribution of examples between its classes can be considered imbalanced.

As noted in [35], there are different types of imbalances and three specific types of imbalances are considered most important:

1. Relative or between-class imbalance

2. Imbalance due to rare examples and within-class imbalance

3. A combination of imbalanced data and small sample size

The most common type of imbalance is **Between-Class Imbalance**. An unbalanced dataset contains significantly more examples in one class than in the other classes. For example, in a binary classification problem with two classes "A" and "B", class "A" could outnumber class "B" by a large factor and orders such as 10000:1 are not uncommon. The same can be extended to multi-class classification problems. The under-represented class is referred to as the **Minority Class**.

Class imbalance affects the performance of the classifier. In many cases, the classifier is unable to learn the minority class effectively due to the lack of examples. Minority classes occur in a variety of application domains, including identifying fraudulent credit card transactions, text categorization, detecting certain objects from images and predicting cancer, to name a few. In each of these applications, the minority class is the one of greater importance, and the accuracy of classification is, in many cases, a matter of life-or-death. A naive classifier would learn the majority class well, but is unable to learn the minority class extensively due to a lack of examples. A new example from the minority class is more likely to be misclassified with such a naive classifier.

Much of the published work in machine learning addresses relative or between-class imbalance. Techniques have been developed for learning in the presence of imbalanced data [19, 35].

Class imbalance can be a result of several factors and we list a few here.

1. *Nature of the problem*

In several domains, examples of some classes are naturally fewer in number compared to other classes within the same problem domain. In such cases, identifying more examples of the minority class may be close to impossible.

2. *Exploration of the Problem Space*

   In some scenarios such as processing streaming data, the distribution of incoming data could be such that examples of one of the classes could arrive later in the stream. As a result, early in the process, or at a given time interval, such a class might appear to be a minority. However, over time, it is possible for the dataset to be better balanced.

3. *Cost of Exploration*

   The cost of collecting data is usually high for most domains. For scientific domains, the data collection often involves long hours in a laboratory setting or on-field sensing. The cost is even higher when labeled data is required.

4. *Choice of Features*

   While the skewed class distribution causes the imbalance, in some scenarios, the choice of features or the set of recorded attributes may be bad enough to make learning harder than it is if the data processing was performed with different choice of features.

Class Imbalance Can be a Good Thing

In some real life classification tasks, having a skewed dataset, where $|C_2| >> |C_1|$, can be beneficial. Let us take the problem of keyword or query classification in search engines. In order to identify the category a search query might belong to, the queries are classified

using multiple binary classifiers. We might use a classifier for music, movie, travel, video games etc. Suppose we would like to build a classifier to identify if the search query is about a video game. An easy approach is to collect a set of search queries that led the user to click a video game webpage, the set of positive examples C_1, and search queries that did not lead to a video game webpage, the set of negative examples C_2. Since video game titles often have similar name or shared keywords, the n-grams formed by the words in the movie title can be used as the feature set.

In this case, it is desirable to have more negative examples than positive examples, or $|C_2| >> |C_1|$. That way, the classifier can precisely identify video game titles and have lower risk of misclassification.

Apart from showing that class imbalance is not necessarily evil, the example also highlights another important point. Traditionally, research on class imbalance has been concerned with the frequency of occurrence of the examples. However, as demonstrated by the example above, given a rich enough feature space and the requirement of the classifier to avoid misclassification, having a skewed dataset may not be as bad as generally believed.

Further, in some problems, the learning becomes easier when the dimensionality "d" is large. Several studies have studied binary classification problems that are "easier" when d is large [39, 52]. Dicker and Foster [23] show that the prediction is easier when the dimension of x_i is large since that provides more context in regression problems.

2.2.2 One-Shot Learning

One shot learning [26, 66] refers to learning an entire class with one example by utilizing preexisting knowledge derived from similar tasks learned in the past. It is a learning

scenario wherein a model is able to learn or generalize a concept or a category C_i using as few as one example from C_i, given some background knowledge. Existing approaches initially build a model using a large initial dataset, which acts as prior knowledge of the domain. When an example from an unseen class is presented, the model learns the class using the information supplied by the existing model and based on the new example's similarity to previously learned examples.

An older approach, Explanation Based Learning (EBL) [22], also learns from one example and in a sense is very similar to one-shot learning as it is known today. EBL requires a domain theory, which needs to be consistent and complete, a training example, a goal concept and an operationality criteria. EBL generalizes the training example such that the generalization is consistent with the goal concept and meets the operationality criteria. Among the various issues with EBL, defining the operationality criteria is not easy.

2.2.3 Dataset Shift

In real world applications, the conditions in which we use a statistical model differ from the conditions that existed when training the model for several reasons. Dataset shift deals with such differences by relating information in very similar or related environments or tasks to help in the prediction of one task given the data or prior knowledge in the others. As noted in [56], there are several flavors of dataset shift.

1. Simple covariate shift occurs when only the distributions of the covariates change and everything else is the same.

2. Prior probability shift occurs when only the distribution over the target variable changes and everything else stays the same.

3. Sample selection bias occurs when the distributions differ as a result of an unknown sample rejection process.

4. Imbalanced data is a form of deliberate dataset shift for computational or modeling convenience.

5. Domain shift involves changes in measurement. Source component shift involves changes in strength of contributing components.

2.2.4 Domain Transfer

Domain Transfer, or transfer learning is motivated by the fact that humans can apply knowledge learned previously to solve new problems that might arise in similar or related tasks, faster or with higher accuracy. In 2005, DARPA [21] defined transfer learning as the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks. Thus, there are two sets of tasks: a source task, from which the learning system learns and a target task, to which the acquired/learned knowledge is applied. The source and target tasks need not be the same, but are related in some sense. This is in contrast with traditional machine learning approaches which require the source and the target tasks to be the same.

Transfer learning has been categorized into various types depending on the types of tasks and the characteristics of the source and target domains. These include inductive transfer, multitask learning, domain adaptation, and unsupervised transfer learning.

2.3 Rare examples

Unbalanced data [35, 55, 62] and its effects have been a topic of interest for a long time in supervised learning. Almost any real world application suffers from the effects of unbalanced data. As noted in [35], there are different types of imbalances and three specific types of imbalances are more prominent:

1. Relative or between-class imbalance
2. Imbalance due to rare examples and within-class imbalance
3. A combination of imbalanced data and small sample size

Much of the published work in machine learning addresses relative or between-class imbalance. Techniques have been developed for learning in the presence of imbalanced data [19, 35] where the number of examples in class C_1 greatly exceeds the number of examples in class C_2. Further, some multilevel approaches [77] are known to aid prediction in the presence of data imbalance. Many approaches artificially "balance" the imbalanced datasets by sampling. Some of these approaches either deliberately or unintentionally compromise overall performance to achieve good approximation on rare examples [19, 35, 55].

Some have used the terms "imbalance" and "rare example" to refer to *Relative or between-class imbalance* only. Imbalance due to rare examples, on the other hand, has received less focus. This problem arises in domains which lack representative examples for sub-concepts within a class. Further, the imbalance due to rare examples also occurs when the example has features that are different from examples of the class it belongs to.

Relatively, there is more work on detecting rare events or rare examples, than there is on prediction tasks. Past research has concentrated on detecting rare cases from a set of examples, especially in streaming data [74]. Only a handful of the existing approaches study learning in the presence of rare examples [62]. There are some issues of both theoretical and practical importance that affect progress.

In our investigation, we have found that many of the existing algorithms are unable to provide good approximations for examples that are rare or under-represented in the training set. For example, it is important to predict the toxicity of a chemical that belongs to an unknown chemical class or functional group accurately while preserving, or even the overall accuracy of our model. Further, the presence of within-class variation and inter-class similarity affects the performance of a machine learning model. When compared to traditional techniques, our approach achieves better performance on such data without compromising the overall error rate.

Moreover, there are two more problems with existing definitions of imbalance and rareness. The existing approaches are based on:

- **Frequency only**: All definitions of imbalance are based on frequency of occurrence of the examples in the dataset. In a two-class classification problem, given two classes C_1 and C_2, the number of examples of C_1, represented as $|C_1|$, is significantly more than the number of examples of class C_2, represented as $|C_2|$.

  However, as discussed earlier in Section 2.2.1, frequency alone cannot be used to define imbalance. In datasets with within-class imbalance, this definition and approaches based on the definition, are more likely to have higher error rates. Classifiers must take features into account.

- **A 0-1 step function based threshold**: Generally, class imbalance and rareness

are considered as a 0-1 thresholds as illustrated in the Equation below.

$$R = \begin{cases} 0 & \text{for} \quad C_1 < T \\ 1 & \text{for} \quad C_1 >= T \end{cases} \tag{2.1}$$

Where, R is the degree of rareness or imbalance, T is the threshold to determine if the dataset is imbalanced. Typical values of T are 1%, 5%, ... 30%.

In the next section, we provide a quantitative representation of rareness.

2.3.1 Quantifying Rareness

Currently, imbalance and rareness are defined as a function of the data frequency. Data points that occur less frequently are considered rare. Likewise, in a classification problem, the dataset is considered unbalanced if one of the classes has fewer examples compared to the other classes. There is a general belief that an unbalanced dataset is hard to learn from, where the value of learning is measured by a combination of the ability of the learning algorithm to learn efficiently, and to predict new examples with high accuracy.

In practice, however, there are problems where a 75-25 imbalance is easy to learn, while the same imbalance can be hard to learn in other cases. Primary differentiating factors include the choice of features and the quality of the examples. Generally, if the features are descriptive of the dataset, the learning problem is easier, and we can generalize with higher accuracy. In several real life situations, obtaining such features may not be easy for several reasons such as those discussed in Section 2.2.

By definition of rareness, a given example is not only infrequent but is also different from other examples of the same class. By this, a single example which is similar to other examples in its class is not rare and this case is similar to typical online learning. If 5

out of 100 examples are different from the rest of the examples, they are rare to a high degree. If 50 out of 100 examples in a class are different from other examples in the same class, they are "rare" to a small (but non-zero) extent.

A distance measure such as Euclidian distance can be used to determine how much an example is different from, or deviates from the other examples. The distance of an example x_i is either measured from the centroid of set of examples belonging to a class C_k, or is measured pairwise between all x_i and all other examples. The choice of approach depends on the application. In order to determine rareness, we measure the distance of an example from the centroid.

Defining Rare Examples Based on the our observations, we can define rare examples or patterns as follows:

"An example x_i belonging to a class C in the given dataset D is considered rare if x_i deviates more from the centroid than other examples in class C, and the frequency of class C, F_C is small, i.e., $|F_C| \ll |D|$."

In order to illustrate this, let us consider the problem of detecting cancer. Let class C represent all known cases of cancer in the dataset. The number of examples of "cancer" is relatively low compared to the number of examples of "not cancer". Now, if x_i represents a new form of cancer that was recently detected, then x_i will be considered rare.

The definition of rareness can be further generalized to describe rareness in a hierarchy of classes. That is, if x_i is an example of a subclass of C that occurs with low frequency, and x_i is different from the rest of the examples, then it is considered rare. Let us consider the example of hand-written characters from Section 1.1. The objective is to learn to recognize hand-written digits given a dataset containing several images of digits 0-9. Each person has a different way of writing the digits. Such variations can occur due to

cultural differences, or can be result of personal habit. Let us consider the set of examples of the number 4. There are several variations or patterns of writing 4. For instance, some people leave the top "open" like in the seven segment display, while others use a "closed" top. If the example x_i represents one of these variations 4, then it is different from rest of the examples of the number 4. x_i is considered rare if the variation it represents occurs with low frequency with respect to the overall frequency of all examples representing the number 4.

Thus the definition covers the different kinds of imbalances: between class and within class imbalances; we described these earlier in Section 2.2.1.

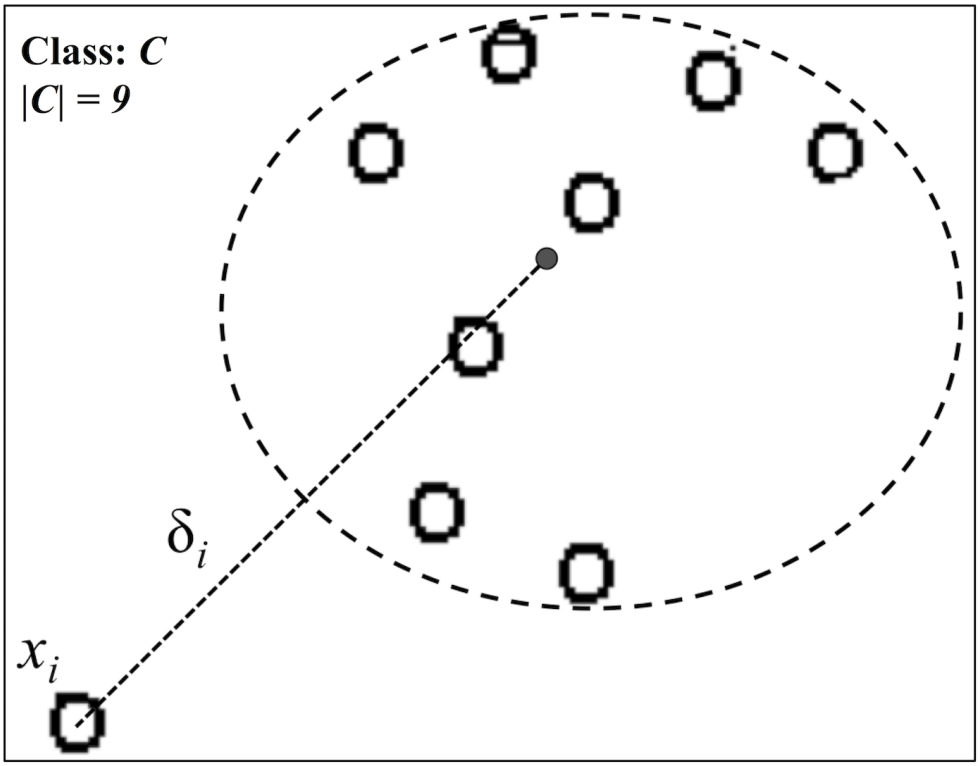


Figure 2.1: Example x_i, belongs to class C. δ_i is the deviation and F_C is the frequency of class C with respect to the dataset. δ_i is measured from the centroid of C. $F_C = 9/|D|$

The degree of deviation between the example x_i is denoted by $\delta_i \in \mathbb{R}$. F_C denotes the frequency of occurrence of the set V, where $x_i \in V$. Figure 2.1 illustrates the idea.

The rareness measure R_i for an example x_i is then obtained by a combination of δ_i and F_C. The user defines rareness based on the application. We use the following rationale to obtain R_i. We use T to represent the threshold of frequency.

1. $F_C \geq T$ and small values δ_i: The example in the dataset is related to the rest of the examples in the class, and the class has sufficient representation.

2. $F_C < T$ and small values δ_i: The examples in the dataset are related but the evidence of occurrence is low – an *infrequent pattern.*

3. $F_C = 0$: The example has not appeared yet in the dataset. δ_i is undefined in this case.

4. $F_C \geq T$ and large values of δ_i: In spite of high frequency of occurrence, the example may not be related to the others in the dataset.

5. $F_C < T$ and large values of δ_i: a *rare example.*

Using the above rationale, we derive the values of R. A simple equation such as the one shown below can help us realize the rationale in a 3-dimensional Euclidean space. We can express rareness of example x_i in class C with the following function:

$$R_i = \delta_i^m + (1 - F_C)^n \tag{2.2}$$

where m and n are constants. In general, we choose m and n to lie in the open interval (2, 10). The values of m and n determine the tradeoff between low frequency $(1 - F_C)$ or

high relationship (δ_i). We can determine the amount of rareness by appropriate choice of values m and n.

The 3-dimensional plot in Figure 2.2 describes the value of R (along the Z axis). In this 3d plot, $0 \leq \delta_i \leq 1$ and $0 \leq 1 - F_C \leq 1$. The same can be generalized to larger values as well.

The degree of rareness from Equation 2.2 serves two purposes.

- We can identify rare examples from a dataset such as those used in the experiments from Chapters 4 and 5.
- The degree of rareness can used as a parameter when learning a classification model in Chapter 5.

2.4 Challenges in Learning with Small Number of Examples

Learning with a small number of examples is not without challenges. Understanding some of these is key in solving the problem. Over the years, some of these challenges have been better understood, theories have been developed to effectively solve several other, while still others remains an open area for research. In this section, we highlight some of the primary pitfalls and challenges that one needs to remember when learning with a limited number of examples.

2.4.1 Bias-Variance Tradeoff

The trade-off between model complexity and mode accuracy is unavoidable regardless of the problem being solved. This is often called the Bias-Variance trade off, and represents

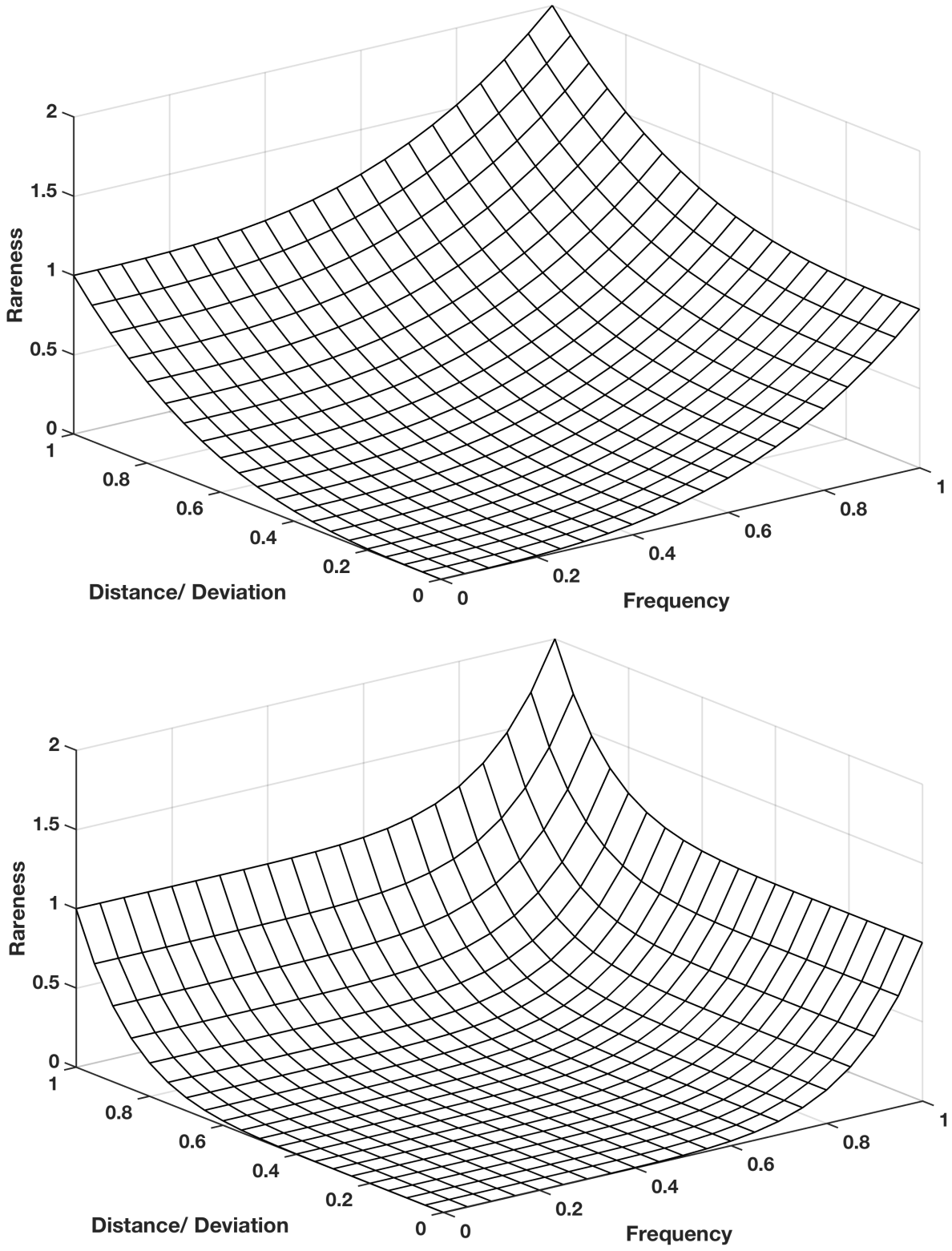


Figure 2.2: Value of R: δ_i -X axis, $(1 - F_C)$ - Y axis, R - Z axis. $m = n = 3$ and $m = n = 7$

one of the primary concepts in machine learning. In short, if the hypothesis space is too small and/or simple for the application, in general there will be high bias but low variance. On the other hand, if the hypothesis space is too large and/or complex for the application, in general there will be low bias but high variance.

When presented with a rich dataset containing sufficient examples to represent each concept, the model will be able to learn the concepts well. In such cases, however, we still run into the risk of learning a very complex model, or a model with very high variance. Predictive accuracy is low in such models as well, and the accuracy decreases with increase in model complexity.

In learning problems involving a small dataset, the model will have high bias. Since the data set to learn the concept is limited, the model has limited opportunity to fully capture the various nuances of the concept in its entirety. This in turn affects the predictive accuracy of the model since future examples are not guaranteed to be exact replicas of the data point or points used to learn the concept. Since we are interested in learning rare concepts, handling high bias is important, as is ensuring a balance between bias and variance.

A model that is able to simultaneously minimize bias and variance generally has the highest accuracy. In a learning problem, if t is the unknown correct value, $\hat{t}$ is the predicted value, and $E[\hat{t}]$ is mean predicted value over all possible training sets, then

$$\begin{aligned} Bias(\hat{t}) &= (E[\hat{t}] - t) \\ Variance(\hat{t}) &= E[(\hat{t} - E[\hat{t}])^2] \end{aligned} \tag{2.3}$$

Typically, in prediction problems, we minimize the Mean Squared Error, or MSE.

$$MSE = E[(\hat{t} - t)^2] \tag{2.4}$$

The above Equation 2.4 can be rewritten as follows:

$$\begin{aligned} MSE &= E[(\hat{t} - t)^2] \\ &= E[(\hat{t} - E[\hat{t}] + E[\hat{t}] - t)^2] \\ &= E[(\hat{t} - E[\hat{t}])^2 + (E[\hat{t}] - t)^2 + 2(E[\hat{t}] - t)(\hat{t} - E[\hat{t}])] \\ &= E[(\hat{t} - E[\hat{t}])^2] + E[(E[\hat{t}] - t)^2] + E[2(E[\hat{t}] - t)(\hat{t} - E[\hat{t}])] \\ &= E[(\hat{t} - E[\hat{t}])^2] + (E[\hat{t}] - t)^2 \\ &= Variance(\hat{t}) + Bias(\hat{t})^2 \end{aligned} \tag{2.5}$$

Note that because $(E[\hat{t}] - t)$ is a constant and $(\hat{t} - E[\hat{t}]) = 0$, the MSE simplifies to a sum of variance and the square of the bias. Thus, minimizing both bias and variance results in low error, or high accuracy. Figure 2.3 illustrates this relationship.

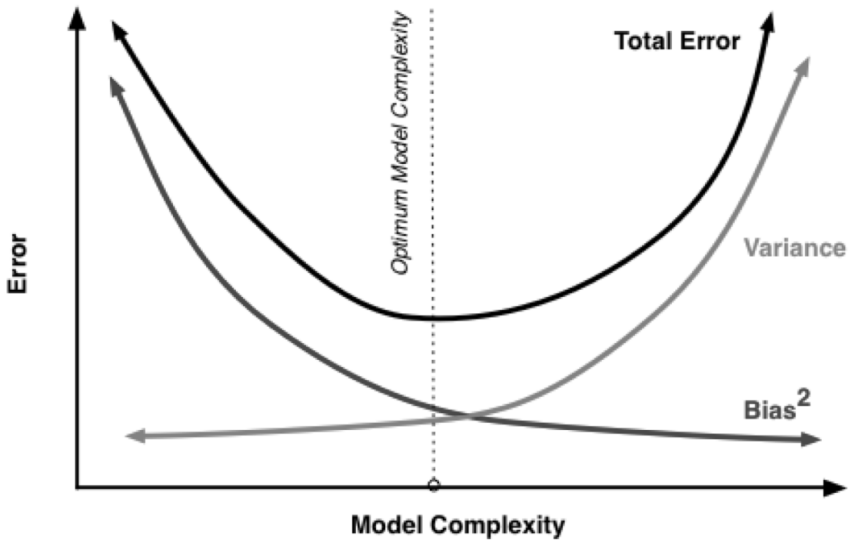


Figure 2.3: Relationship between $Error(MSE)$, $Variance$ and $Bias^2$

When learning from rare concepts, minimizing bias is a challenge. In some cases, as we obtain more examples to represent the rare concept, retraining the model is one way to reduce bias. In Chapters 4 and 5, we will discuss how our models are able to improve predictive accuracy with a simplest possible model, thus achieving a good balance between bias and variance.

2.4.2 Noise vs. Rare Data

In general, noisy examples and rare data points are very similar, and in some cases practically indistinguishable. There are several ways to handle noise in datasets.

- Treat noise as a rare example – we can treat the noisy example as a representative of an extremely rare class. In many applications, the rare example is contradictory to already known facts. While it is easy to discard the rare example as noise, but in doing so we lose information carried by that example.

- Ignore the noisy example – Probably the simplest thing to do is to ignore noise, and learn the other concepts in the dataset. While this is a safe option when learning a simple model, we must have a good mechanism to identify or isolate noise from the rest of the data. In some cases, the learning algorithms themselves are robust to noise.

- Correct the noise – Correcting noise requires either reacquiring one or many of the features for the datapoint, which can often be a tedious and expensive venture.

2.4.3 Obtaining Sufficient Examples, and Good Quality Class Labels

A rare example, by definition, occurs less frequently than other examples in the problem space, and is often different from the already known examples. In such cases, an expert's supervision is required in order to obtain the correct target value. Techniques such as Active Learning and Semi Supervised learning are often employed to obtain the class labels. However, due to the nature of the problem, this challenge remains.

2.5 Related Concepts

There are several concepts related to rare examples, and it is important to understand their similarities and differences with respect to our work. In particular, we would like to highlight three of the most prominent concepts.

2.5.1 Long Tail

The Long Tail is a well known phenomenon wherein a small number of generic objects/entities/words appear very often and most others appear more rarely. Long tail is a term used in variety of domains ranging from media, online business of search and advertising, economic models and marketing. The "tail" is the part of a distribution having several examples or occurrences with low frequency and the "head" is the part showing occurrences with higher frequency. A typical "long tail" distribution is shown in Figure 2.4.

Long tail distributions are common in various domains. Some examples include search engines, estimating user base, market forecasting and supply-demand problems.

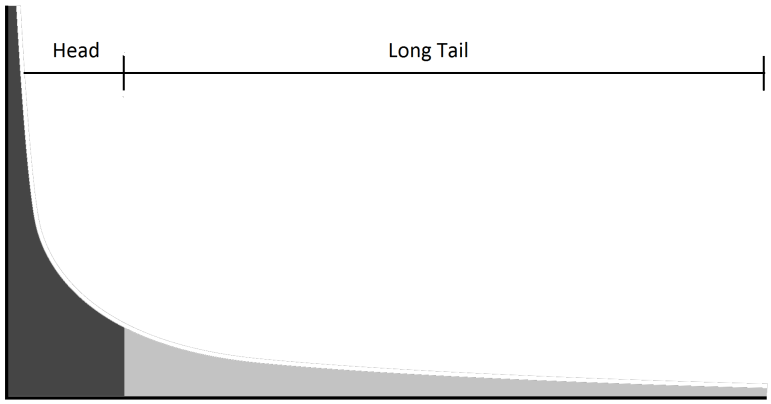


Figure 2.4: Long Tail

In search engines, the distribution representing the frequency of searches for keywords or queries is considered long tail. That is, a small number of keywords or queries are searched for with very high frequently in a given time period and a large number of other keywords or queries are searched for with relatively low frequency. The former set of keywords constitutes the head and the latter set of keywords forms the long tail. For instance, during a major sporting event like the super bowl, keywords such as NFL, Super Bowl, NFC, AFC, Super Bowl MVP, Super Bowl Ads etc., would be part of the head. The tail might consist of queries such as "Discovery of the 9th planet".

Related concepts include the Power law, Zipf's law [28], and Pareto distribution. Power laws deal with the frequency of distribution of items, and can be observed in a wide range of applications. A power-law probability distribution is a distribution whose density function (or mass function in the discrete case) has the form $P(X > x) \sim x^{-\alpha+1}$, where $\alpha > 1$.

Zipf's law gives the relationship between the size of words and their frequency. According to Zipf's law, the frequency of any item is inversely proportional to its rank in

the frequency table for the domain. One application is the study of how terms are distributed across documents. If the most frequent term occurs f_1 times, then the second most frequent term occurs half as many times ($f_2 = f_1/2$), the third most frequent term a third as many occurrences ($f_3 = f_1/3$), and so on. In general, $f_i = f_1/i$ for i=1,2,. . . , n. Thus frequency decreases very rapidly with rank. Zipf's law is a discrete distribution, and can be represented by a Zeta Distribution.

The Pareto distribution is similar to Zipf's law but is a continuous distribution. Pareto's law is given in terms of the cumulative distribution function (CDF) and it represents the number of items larger than x is an inverse power of x: $P[X > x] \sim x^{-}k$. The Pareto principle, or the 80-20 principle is a special case of the Pareto distribution.

2.5.2 Cold Start

The cold start problem is common in inference engines and recommendation systems where the system cannot draw meaningful inferences about users items about which sufficient information is not yet available. Recommendation systems identify common patterns in people's behavior which are then used to make predictions and recommendations of new items to a user. However, such methods perform poorly when presented with new users with minimal information.

One might argue that if additional features are known, the cold start problem can be avoided. For instance, demographic or geographic information of a user would reduce the effects of cold start. In many cases, additional information is not available due to concerns like privacy and cost of obtaining such information.

There are several approaches to solve the cold start problem. One common approach is to use the best estimate of values for the new example. The estimate for the new

example x is obtained by using the values of the previously known examples that are similar to x. When very little to no information is known about the new example, the estimate for x is determined based on the problem. For example, when a new user creates an account in e-commerce web sites like Amazon or E-Bay, the most frequently bought items are shown as recommendations. More personalized recommendations are presented as the user makes more purchases.

Auto insurance companies face similar problems as well. The insurance rates of a car depend on, among other things, the cost of repairing the car if it were to get into an accident, the likelihood of a car getting into an accident and so on. When a new car is released in the market, determining such factors is difficult and the insurance company often has to come up with the best estimate. The estimate will be revised later as the car has been in the market for a while and more data is available. The initial estimate must not be too low or too high, and the revised estimate must not lead to an enormous increase in the insurance premium rates. The correctness of the estimate depends on the problem domain as well.

Recently, active learning has been identified as an approach to solving the cold start problem [36]. With active learning, we could potentially distinguish between informative and noisy data points

2.5.3 Outliers and Anomalies

Outliers or Anomalies are unusual occurrences of data and typically lie outside the distribution the data. Most approaches are based on distance measures, finding similarities, or computing nearest neighbors [1, 34, 51, 64, 67]. Generally, outliers are considered to be of less useful when finding patterns in the data.

However, in several applications such as fraud detection, identifying anomalies is a primary task. Fraudulent activities exhibit behaviors that deviate from the normal or legitimate activities. Standard measures for evaluating outlier detection problems include

- Precision and Recall or Detection rate
- False alarm or false positives
- ROC Curve between detection rate and false alarm rate

2.6 Rare Data and Transfer Learning

Inductive transfer, or transfer learning focuses on using knowledge gained while learning one problem and applying it to a different but related problem.

Among others things and depending on the type of learning setting, transfer learning might require fewer labeled examples to learn compared to a traditional learning approach, and the source and target domains need not be the same.

In recent years, a variety of approaches have been studied under transfer learning [50] and hierarchical or, multilevel models models [6, 17, 31]. These techniques have been shown to perform better than traditional approaches in different problem domains. The primary aim of these approaches is to learn a set of tasks in such way that the performance on related tasks is improved. These approaches also aim to improve prediction in the presence of fewer training examples. However, there are issues that need to be addressed. In this section, we discuss the effects of imbalanced data in such learning problems.

Imbalance can affect performance in hierarchical models and transfer learning tasks in at least three different ways:

1. Affects one-shot learning

2. Potentially lead to negative transfer

3. May be unable to generalize a particular class due to lack of variation in examples

We highlight the first two issues below. Arriving at a solution to such problems is not easy. We believe that it is not easy, even for humans, to learn with high accuracy in such a setting. But, these will be interesting problems to study. Similar observations have been made about the effect of rare data in learning in general [74]. We highlight issues specific to transfer learning in the discussion below.

2.6.1 Imbalance and One-Shot Learning

When an example belonging to a previously unseen class is presented, the algorithm learns the class using the information supplied from the existing information based on the new example's similarity to previously learned examples. The new class, thus learned, can be considered to be novel. One shot learning [26, 66] refers to learning an entire class with one example by utilizing preexisting knowledge derived from similar tasks learned in the past. It is a learning scenario where a model is able to learn or generalize a class or a category C_i using just one example from C_i, given some background knowledge. Existing approaches initially build a model using a large initial dataset, which acts as prior knowledge of the domain. When an example from an unseen class is presented, the model learns the class using the information supplied by the existing model, and based on the similarity of the new example to previously learned examples.

The problem arises when one of the following happen:

1. When the initial training data is imbalanced

2. The training data is not available in batch instead is available one-by-one in real

time

One shot learning assumes presence of adequate knowledge at training time. When one of the aforementioned problems arise, there is very little variation and the resulting models are either biased or offer inadequate information for learning new concepts. To be practically applicable, models must be developed that can handle the cases highlighted above.

2.6.2 Imbalance and Negative Transfer

Negative transfer has been identified as one of the major problems for transfer learning techniques [50]. Negative transfer leads to reduced learning performance in the target domain. This might arise due to several reasons already noted, including presence of outliers [57] and dissimilarities between the tasks [50]. However, very little research has been published on this topic [4, 9, 10, 57, 65].

It is known that imbalanced data can affect performance of learning algorithms. As noted in [35], imbalance due to rare examples and within-class imbalance can affect performance in learning tasks regardless of between-class imbalance. Rare examples are not outliers but examples whose classes do not have sufficient number of examples. However, the rare examples act as outliers in the dataset and in turn affect performance much like outliers by causing negative transfer.

Chapter 3

Background Concepts

In this section, we discuss the background for the relevant past research on learning with a small number of examples and imbalance (Section 3.1) and research on Multilevel Models (Section 3.2). Some of the concepts presented in this chapter will be used in future chapters.

3.1 Class Imbalance Learning

Learning with imbalanced datasets has been a research problem for several years, and it continues to draw interest as more applications present us with datasets with skewed distribution of samples from different classes. The performance of most standard learning algorithms is significantly less when presented with an imbalanced learning problem. The standard algorithms expect balanced class distributions or equal misclassification costs. When presented with an imbalanced dataset, these algorithms are unable to fit the characteristics of data, resulting in poor classification accuracy.

The most common type of imbalance is **Between-Class Imbalance**. An imbalanced

dataset contains significantly more examples in one class than in the other classes. For example, in a binary classification problem with two classes "A" and "B", class "A" could out-number class "B" by a large factor. The under-represented class is referred to as the **Minority Class**. The same intuition can be applied to multi class datasets, where the minority class is a set of one or more classes. Such datasets occur in variety of applications such as fraud detection, cancer research, internet search, bioinformatics and more.

The cost of misclassification can be significantly high depending on the application. In cancer research, the number of cancerous cases is significantly less than the number of non-cancerous cases. However, misclassifying a cancerous patient as non-cancerous patient is very serious. In fact, it can be more serious than classifying a non-cancerous patient as cancerous. While both are bad, the former is more serious. Traditional machine learning techniques are not effective in such problem settings for two main reasons.

- Traditional machine learning techniques require the datasets to have similar sample counts.
- Traditional algorithms assign equal weight to any misclassification, and are not capable of representing the cost of misclassification correctly.

3.1.1 Approaches

In this section, we describe techniques that have been proposed for learning with imbalanced data. The approaches can be classified into the following types, based on the underlying technique used. They are:

- Sampling Based Approaches – These approaches modify the dataset to artificially balance the dataset such that the majority and minority class have similar counts.

- Cost Sensitive Learning – These approaches account for the cost of misclassification when learning.

- Kernel Methods – Support vector machines and kernel based learning can be applied to learning with imbalance by appropriate choice of re-representation to get an ideal separation.

- Active Learning Based Methods – These are hybrid approaches which sample the dataset to obtain labels for the most informative examples, which in turn result in high accuracy during classification.

Sampling Based Approaches Sampling methods are the simplest approaches to improving classification accuracy for imbalanced datasets. Sampling methods create a balanced distribution for a given imbalanced dataset by either under sampling the majority class or oversampling the minority class. The resulting balanced dataset can be used as inputs to traditional classifiers for learning. It has been shown that applying such sampling techniques has improved classification accuracy for most imbalanced data sets.

There are several ways of sampling the datasets, and three techniques are common.

- **Random Sampling** – Random sampling is probably the first technique to be applied when trying to learn from imbalanced data. Random sampling can be used to either undersample the majority class or oversample the minority class.

  Undersampling removes data from the original data set, more specifically from the majority class. We continue to remove samples at random from the majority class until we have removed sufficient examples to balance the dataset. Undersampling gives us a simple method for adjusting the balance of the original data set. This technique has one serious problem. Removing examples from the majority class may

cause the classifier to miss some informative data points pertaining to the majority class.

Oversampling augments the minority set by replicating the selected examples and adding them to minority set. In this way, the number of total examples in the minority class is increased and the class distribution balance can be achieved. Oversampling can lead to overfitting since it appends replicated data to the original data set.

- **Informed Sampling** – Informed sampling overcomes the information loss introduced in the random sampling method described above. Typical informed sampling techniques use ensemble models by independently sampling several subsets from the majority class and developing multiple classifiers based on the combination of each subset with the minority class data.

- **Synthetic Sampling or Data Generation** – This approach creates artificial data based on the feature space similarities between existing minority examples. SMOTE (Synthetic Minority Over-Sampling Technique) is a common and powerful sampling technique that has shown good results in various applications.

Cost Sensitive Learning Cost-sensitive learning methods consider the costs associated with misclassifying examples [43, 44]. Cost-sensitive learning uses different cost matrices to describe the misclassification costs for a given data example. Cost-sensitive learning yields superior results when compared to sampling methods in many learning problems.

The definition of the cost matrix is fundamental to the cost-sensitive learning techniques. The cost matrix is a numerical representation of the penalty of classifying exam-

ples from one class to another. In a binary classification scenario, the cost can be defined as follows:

- There is no cost for correct classification of either class
- The cost of misclassifying minority examples is higher than the cost of misclassifying majority class examples.

Based on the above scheme, the objective of cost-sensitive learning is to develop a hypothesis that minimizes the overall cost on the training data set, which is usually the Bayes conditional risk.

There are many different ways of implementing cost-sensitive learning, and the techniques fall under three categories.

1. Apply misclassification costs to the data set as a form of data space weighting and these techniques can be thought of as cost-sensitive bootstrap sampling approaches. The misclassification costs are used to select the best training distribution.
2. Apply cost-minimizing techniques to ensemble methods and standard learning algorithms are integrated with ensemble methods to develop cost-sensitive classifiers.
3. Apply cost-sensitive functions or features directly into classification paradigms.

In addition, adaptive boosting, cost-sensitive decision tree, and cost sensitive neural networks have also been developed to learn imbalanced data. While cost sensitive learning approaches are effective, determining a cost representation of a given domain can be challenging and in some cases impossible.

Kernel Methods Kernel-based learning paradigm using Support Vector Machines (SVMs), can be very effective when learning with imbalanced data sets [2, 58, 75]. SVMs use support vectors, or examples near concept boundaries, to maximize the separation margin between the support vectors and the hyperplane boundary. In this process, SVMs minimize the total classification error.

When applied naively, since SVMs minimize total error, they are biased toward the majority class.If there is a lack of data representing the minority concept, there could be an imbalance of representative support vectors resulting in degradation of performance.The optimal hyperplane separating the classes will be biased toward the majority class in order to minimize the high error rates of misclassifying the more prevalent majority class. Thus, SVMs will learn to classify the majority class resulting in minimal error rate across the data set.

Integration of sampling and ensemble techniques to the SVM can improve the performance. With active learning [24], we sample the dataset to obtain labels for the most informative examples, which in turn result in high accuracy during classification. The informative examples can act as support vectors for the model.

3.1.2 Evaluation and Metrics for Imbalance Learning

Let us consider a two-class classification problem, with classes C_1 and C_2. Without loss of generality, let $|C_1| \gg |C_2|$. In other words, C_2 is the minority class. The performance of a classifier on a given dataset for this problem can be evaluated using traditional approaches including *Accuracy* and *Error Rate*.

$$Accuracy = \frac{TP + TN}{Total\ Number\ of\ Examples} \tag{3.1}$$

where, TP is True Positive or the number of examples of C_1 correctly classified as C_1, and TN is True Negative or the number of examples of C_2 correctly classified as C_2.

$$ErrorRate = 1 - Accuracy \tag{3.2}$$

However, there is a problem; these metrics do not reflect the reality well. Suppose the dataset has 100 examples with $|C_1| = 95$ and $|C_2| = 5$. Achieving 95% accuracy is as simple as assigning the label C_1 to every example in the dataset. Taken as a raw number, 95% is a large number and in most cases, we will be content with such a high accuracy. In case of this problem, however, the misclassification of C_2 was accounted for when evaluating the classifier. All examples of C_2 were misclassified, and in most applications, including cancer prediction, this can be a very bad situation.

Precision and *Recall* are two metrics that are closely related to accuracy but are more effective in demonstrating the performance of the classifier, especially when dealing with imbalanced data. These two metrics are not affected by changes in data distribution. Precision and recall numbers must be reported together and using only one does not present the full picture. While recall does not show how many examples are incorrectly labeled as positive, precision does not portray how many positive examples are labeled incorrectly.

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

where, FP is False Positive or the number of examples of C_2 incorrectly classified as C_1.

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

where, FN is False Negative or the number of examples of C_1 incorrectly classified as C_2.

F-measure is another metric for measuring performance and it provides more insight into the performance of a classifier. F-measure combines prevision and recall into a single measure of classifier effectiveness.

$$F - measure = \frac{(1+\beta)^2 \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision} \tag{3.5}$$

3.2 Multilevel Models

Many of the models that have been studied in the last few years are, in one way or another, multilevel models. In statistical data modeling, *Multilevel Models* are viewed as a generalization of regression models. In other words, multilevel models are an improvement over classical regression. Such models can also be used for various tasks including prediction and causal inference. Typically, multilevel models have been used to handle clustered or grouped data, and hierarchical data. However, there have been many instances where such models have been used for non-hierarchical data also. Multilevel modeling allows relationships to be simultaneously assessed at several levels.

Over the years, the terms hierarchical model, random effects model and mixed effects model have been used to describe multilevel models. Whether or not these terms can be used interchangeably is debatable. For instance, in their book [31], Gelman and Hill state that the usages "mixed" and "random" to be misleading. Part of the reason is that there is no single authoritative definition of these terms.

In general, hierarchical or multilevel models can be thought of as follows: Consider a set of N input examples each derived from a body of high dimensional data. The N objects are partitioned into basic classes or categories. These basic classes are in-turn

partitioned into classes or groups and so on. Characteristics or processes occurring at the nth step of analysis can influence characteristics or processes at subsequent steps. Typically the outputs from earlier steps are used as inputs in the subsequent steps. Constructs are defined at different levels, and the hypothesized relations between these constructs operate across different levels.

In our work, we discuss how multilevel models can be used to solve various machine learning problems. Before discussing the approaches, we present a brief survey on Hierarchical Models (Section 3.2.1) and Bayesian Nonparametric approaches (Section 3.3), which can be considered as different types of multilevel models. These sections provide some relevant related work for the approaches we discuss in later sections.

3.2.1 Hierarchical Models

Hierarchical modeling is an increasingly popular approach to modeling data and is known to outperform classical regression in predictive accuracy [29, 31]. Hierarchical models developed by others [6, 16, 60, 68, 73, 78] have good overall generalization, but much less attention has been given to their ability to deal with rare and unseen examples.

Approaches including Hierarchical Bayesian Models [30, 31] and Hierarchical Kernel Learning [6] are being widely studied. More recently, research has looked into utilizing the power of kernels with multilevel models. Such models have been able to achieve good generalization across various datasets [7].

Hierarchical approaches such as VQSVM [77] have been developed to address the presence of imbalanced data in classification tasks. However, not much is known about presence of rare examples in the test data used for VQSVM.

In the recent past, hierarchical models have been discussed in various forms spanning

various applications. A large amount of such work can be found in computer vision literature [16].

Hierarchical models have found limited use in prediction problems in the presence of rare examples. The foremost question to address is: given the power of hierarchical models, can they be applied to prediction in the presence of the problems discussed earlier? As we shall discuss in Chapter 4, our current work has shown promising results in this direction.

Much multilevel modeling work has focused on batch learning and supervised learning. However, some applications require online learning. Online learning often poses all four of the problems mentioned in the previous section. We must investigate the performance of multilevel models in such applications.

Bouvrie et. al [16] discuss invariant properties of multilevel models. More insight is provided in [17] where the author discusses multilevel models in the context of speech and text data. Gelman et.al [29] discuss the powers of hierarchical models. While research has demonstrated that the hierarchical models are very effective as a generalization technique for both classification and regression, little is still known about the theory and much of the research has been empirical.

Moreover, much of the hierarchical modeling work is now focused on enabling transfer learning [4, 10, 50, 66]. Thus, it is important to study the capabilities of hierarchical models in greater detail.

At this point, it is important to note the difference, or rather the lack of difference between the terms "hierarchical" and "multilevel" models. There exists a considerable confusion in the community in the usage of these terms and they have been used interchangeably in the past. In recent times, the term "hierarchical models" has also been used to address feature hierarchies and hierarchies of class labels (e.g. in topic modeling

and document classification).

Characteristics of Hierarchical Models

For a really useful analysis of given a dataset with observations, we need to separate observations into groups, and understand the relationship between the groups. This is also referred to as "sharing statistical strength". Hierarchical modeling affords such sharing in a Bayesian setting. The parameters of the hierarchical model are shared among groups, and the randomness of the parameters induces dependencies among the groups.

Hierarchical Bayesian Models Hierarchical Bayes models are a popular tool in machine learning.

In a hierarchical Bayes model, the model is specified over several levels where each level is formed by a new distribution of data. Suppose we have data about some random variable Y from m different populations with n observations for each population. Let y_{ij} represent observation j from population i.

Suppose $y_{ij} \sim f_1(\theta_i)$, where f_k is a known distribution and θ_i is a vector of parameters for population i such that $\theta_i \sim f_2(\Theta)$. Here, Θ may also be a vector $\Theta \sim f_3(a, b)$ and is called the hyperprior.

The parameters a and b for the hyperprior may be known a priori and represent our prior beliefs about Θ or, we can also assign a probability distribution for these quantities as well, and proceed to another layer of hierarchy. This sequence of parameters and priors constitutes a hierarchical model. Typically, at the stopping point of the hierarchy, i.e., at the root of the hierarchy, all prior parameters are assumed to be known. Empirical Bayes approaches use the observed data to estimate these priors as well. Further, the priors can be either parametric or nonparametric, depending on unknown parameters in the model.

Both parametric and nonparametric hierarchical Bayes models have been used in recent times. These span a variety of applications concerned with computer vision and natural language processing.

While many of approaches exist, not much has been accomplished towards solving problems pertaining to scalability, feature selection, and applicability in the presence of rare data.

3.3 Bayesian Nonparametrics

Among the biggest challenges in learning and data analysis are the choice of number of clusters in clustering, number of classes to use in a mixture model, and factors in factor analysis. In the classical clustering problem, we assume the existence of k clusters, each associated with a parameter θ_k. The goal of inference is to draw the value of the parameters from the observations. The parameter space $\mathbf{T}$ is finite, i.e., $\mathbf{T} \subset \mathbb{R}^d$. Models following such an approach are called *Parametric Models*, and most traditional machine learning approaches are parametric models.

Parametric models work best in problem settings where significant knowledge is available. Assumptions about the application can be drawn from the prior knowledge. Some of the benefits of parametric algorithms are:

1. Simplicity: These methods are easy to understand and interpret results.

2. Speed: Parametric models are very fast to learn from data.

In general, parametric models do not require as much training data as nonparametric models, since the prior knowledge accounts for most variations in the data and can work well even if the fit to the data is not perfect. However, that is not always the case,

and modeling under uncertainty is not effective with parametric models. As we shall see in later chapters, nonparametric models can be used to learn with rare examples and perform better than parametric models.

A nonparametric Bayesian model [27, 32, 49] is a Bayesian model whose parameter space has infinite dimension. To define a nonparametric Bayesian model, we have to define a probability distribution (the prior) on an infinite-dimensional space.

Bayesian Nonparametric models adapt the model complexity to fit the data. In contrast, parametric models use a predetermined set of parameters to determine the complexity of the model. For instance, a parametric mixture model for clustering requires the number of clusters k to be specified when learning from the data. The Bayesian nonparametric approach estimates the number of clusters needed for the observed data and allows new clusters to be created to fit future data. Thus, the number of clusters in a Bayesian nonparametric model based clustering can grow or shrink as more data is presented to the model, in turn changing the model complexity.

Some of the benefits of Non-Parametric algorithms are:

1. Flexibility: Capable of adapting the complexity if the model to fit the data.

2. Performance: Have higher performance on prediction tasks compared to parametric models.

3. Power: Makes no assumptions about the underlying function and does not rely on prior knowledge about the problem domain.

Bayesian Nonparametrics vs K-Nearest Neighbors Some traditional approaches like k-nearest neighbors and locally weighted regression are considered to be nonparametric. K-nearest neighbors has been applied to detecting outliers [34, 64, 67]. However,

the problem of choosing of k, the number of nearest neighbors, still remains. Further, defining the distance metric has a significant impact on the performance of the model.

3.3.1 Dirichlet Process

The Dirichlet process (DP), was first developed by Ferguson in 1973 [27]. The DP is a distribution over probability measures or probability space Θ, and draws from a Dirichlet process can be interpreted as random distributions. The Dirichlet process prior is very often applied to Bayesian nonparametric models. The availability of computationally efficient methods for posterior simulation is a primary reason for the extensive use of these models.

A distribution G drawn from a DP is denoted as

$$G \sim DP(\alpha, G_0) \tag{3.6}$$

Where, $\alpha > 0$ is the concentration or strength parameter and G_0 is the base distribution over Θ. The probability measure G assigns probability $G(A)$ to every measurable set $A \subset \Theta$ such that for each measurable finite partition $A_1, A_2, \ldots, A_n$, which is given by:

$$(G(A_1), \ldots, G(A_n)) \sim Dirichlet(\alpha_0 G_0(A_1), \ldots, \alpha_0 G_0(A_n)) \tag{3.7}$$

The base distribution G_0 is the mean of the DP. Thus, for every measurable set A, the base distribution is the mean or expected value of the probability measure:

$$E[G(A)] = G_0(A)$$

And, the concentration parameter can be thought of as an inverse variance:

$$V[G(A)] = \frac{G_0(A)(1 - G_0(A))}{(\alpha + 1)}$$

Thus, larger the value of α, the smaller the variance, and the greater is the strength of the prior, or greater is the mass concentrated around the mean. In other words, as $\alpha \to \infty$, $G(A) \to G_0(A)$.

The Dirichlet process prior is a conjugate prior for distributions over Θ. In other words, the posterior over G is also a Dirichlet Process. Since G is a distribution over Θ, a sequence of independent draws from G, $\theta_1, \theta_2, \ldots, \theta_n$ take values from Θ.

Let $\delta_x(A)$ denote the measure for a point x over a measurable set A. It is given by

$$\delta_x(A) = \begin{cases} 0 & \text{for} \quad x \notin A \\ 1 & \text{for} \quad x \in A \end{cases} \tag{3.8}$$

Then, according to Ferguson [27], the posterior Dirichlet process has a concentration parameter $\alpha + n$ and a base distribution $\hat{G_0}$

$$\hat{G_0} = \frac{\alpha G_0 + \sum_{j=0}^{n} \delta_{\theta_i}}{\alpha + n}$$

Thus, the Posterior Dirichlet Process can be written as:

$$G|\theta_1, \theta_2, \ldots, \theta_n \sim Dirichlet(\alpha + n, \frac{1}{\alpha + n}(\alpha G_0 + \sum_{j=0}^{n} \delta_{\theta_i})). \tag{3.9}$$

The posterior base distribution is a weighted average between the prior base distribution G_0 and the empirical distribution $\sum_{j=0}^{n} \delta_{\theta_i}$, where the weight of the prior base distribution is proportional to α, and that of the empirical distribution is proportional to

the number of observations k. This is consistent with the fact that α denotes the strength of the prior. As the number of observations grows, the posterior resembles the empirical distribution.

The posterior can be used to describe the posterior predictive distribution for a new item θ_{n+1}.

$$\begin{aligned} p(\theta_{n+1}|\theta_1, \theta_2, \ldots, \theta_n) &= E[G(A)|\theta_1, \theta_2, \ldots, \theta_n] \\ &= \tfrac{1}{\alpha+n}(\alpha G_0 + \textstyle\sum_{j=0}^{n} \delta_{\theta_i}) \end{aligned} \tag{3.10}$$

Which is the same as the posterior base distribution.

The sequence of predictive distributions for $\theta_1, \theta_2, \ldots, \theta_n$ is called the Blackwell-MacQueen urn scheme, or the Polya's Urn Scheme [12, 25, 54]. This scheme describes that the Dirichlet distribution has the following two properties:

1. It is a discrete distribution.

2. It exhibits clustering property.

The unique values of $\theta_1, \theta_2, \ldots, \theta_n$ induce a partitioning of the dataset into a set of n clusters. This is referred to as the clustering property, and the Chinese Restaurant Process (CRP) shows the clustering effect based on Dirichlet process more explicitly. CRP provides a very useful representation when doing inference in Dirichlet process mixture models (DPMM). We discuss the CRP in Section 3.3.2, and the Dirichlet Process Mixture Models in Section 3.3.3

The Dirichlet Process Mixture Models are interesting in our work in the context of classification models discussed in Chapter 5. We use the clustering property of the DP models to group the data appropriately.

3.3.2 Chinese Restaurant Processes

The Chinese Restaurant Process (CRP) [3, 32, 49, 53] defines a distribution over partitions. The CRP is based on a metaphorical Chinese restaurant with an infinite number of tables a sequence of customers entering the restaurant and sitting down. The customers entering the restaurant are seated at any of the already occupied tables or are seated at the first empty table. using the following scheme:

1. The first customer enters and sits at the first table.

2. The n^{th} customer enters and sits at

   the first unoccupied table with probability

$$P_{newtable} = \frac{\alpha}{n-1+\alpha} \tag{3.11}$$

   an already occupied table k with probability

$$P_{oldtable} = \frac{n_k}{n-1+\alpha} \tag{3.12}$$

where α is a positive real number, called the concentration parameter, and n_k is the number of people at table k. When new customers enter the restaurant, they sit at each of the occupied tables with probability proportional to the number of previous customers sitting there, and at the next unoccupied table with probability proportional to α. At any point in this process, the assignment of customers to tables defines a random partition.

If c_n is the table assignment of the nth customer, the probability that the nth customer

is assigned table k is given by:

$$P(c_n = k|c_1, c_2, \ldots, c_{n-1}) \propto \begin{cases} \frac{n_k}{n-1+\alpha} & \text{for} \quad \textit{previously occupied table } k \leq K \\ \frac{\alpha}{n-1+\alpha} & \text{for} \quad \textit{next unoccupied table } K+1 \end{cases} \tag{3.13}$$

Equation 3.13 shows how the predictive posterior would behave under the CRP scheme. One can see that this is analogous to Equation 3.10, where $c_i \equiv \theta_i$. We can rewrite Equation 3.10 as:

$$p(\theta_n|\theta_1, \theta_2, \ldots, \theta_{n-1}) \sim \frac{1}{\alpha + n - 1}(\alpha G_0 + \sum_{j=1}^{K} n_k \delta_{\theta_i}) \tag{3.14}$$

Where K is the total number of tables or clusters.

Thus the CRP demonstrates the clustering property of the Dirichlet process. Each customer is equivalent to a new data point and each table assignment is equivalent to cluster assignment. Thus each new data point (i.e., customer) inherits the class label (i.e., table number) of the cluster (i.e., table) it is assigned to.

A schematic of CRP is shown in Figure 3.1.

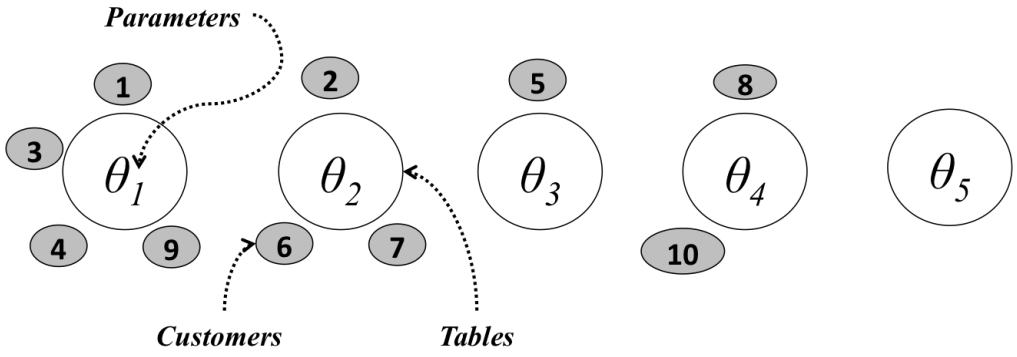


Figure 3.1: Chinese Restaurant Process. Customers are distributed in tables according to the probabilities in Equations 3.11 and 3.11

Effect of Concentration Parameter

The concentration parameter affects the number of customers in each table. Each customer is assigned a table according to the probabilities in Equations 3.11 and 3.12. Based on the probabilities, at lower values of α, new customers are assigned tables with more elements c_k, and at higher values, customers occupy less crowded, or even new tables. By extension, at very large values of α, each customer occupies their own table. More generally, the larger α is, the smaller the variance, and the DP will concentrate more of its mass around the mean.

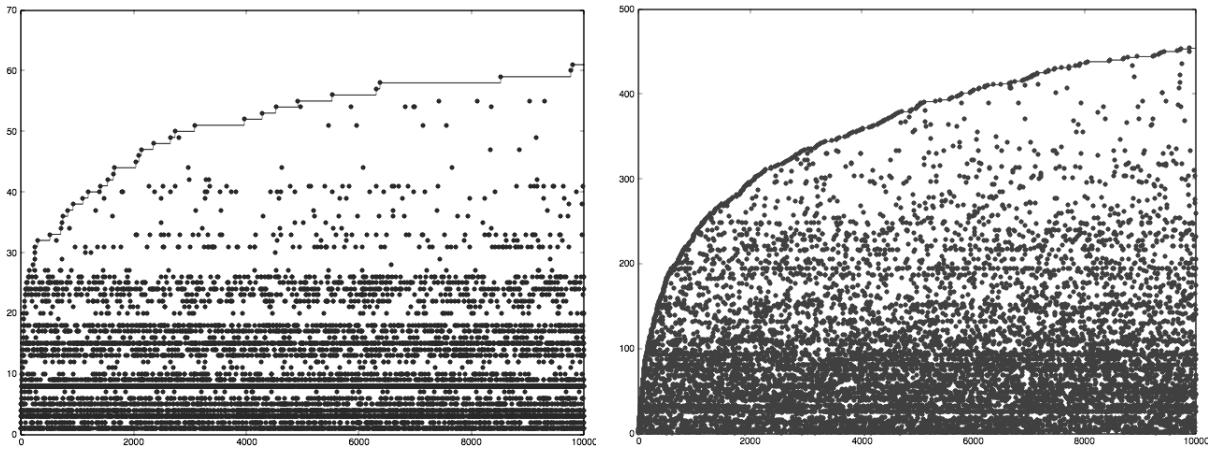
The concentration parameter is also called the strength parameter, referring to the strength of the prior and the base distribution when using the DP as a nonparametric prior in a Bayesian nonparametric model, and the mass parameter, as it influences the distribution of observations.

Thus, by increasing the value of α, it is more likely that an element is given its own cluster, and the number of clusters increases.

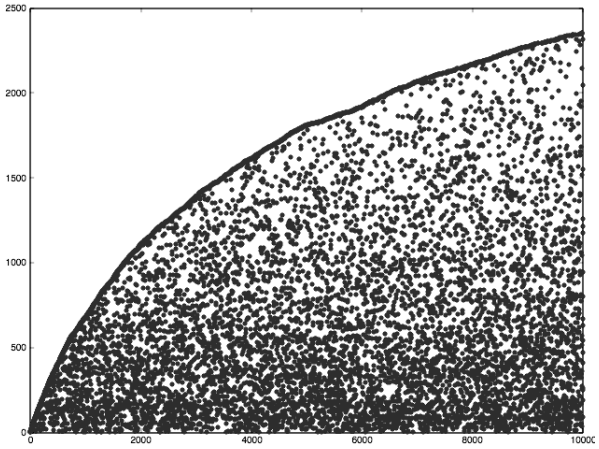
The effect of α on the table assignments is shown in Figure 3.2

3.3.3 Dirichlet Process Mixture Model (DPMM)

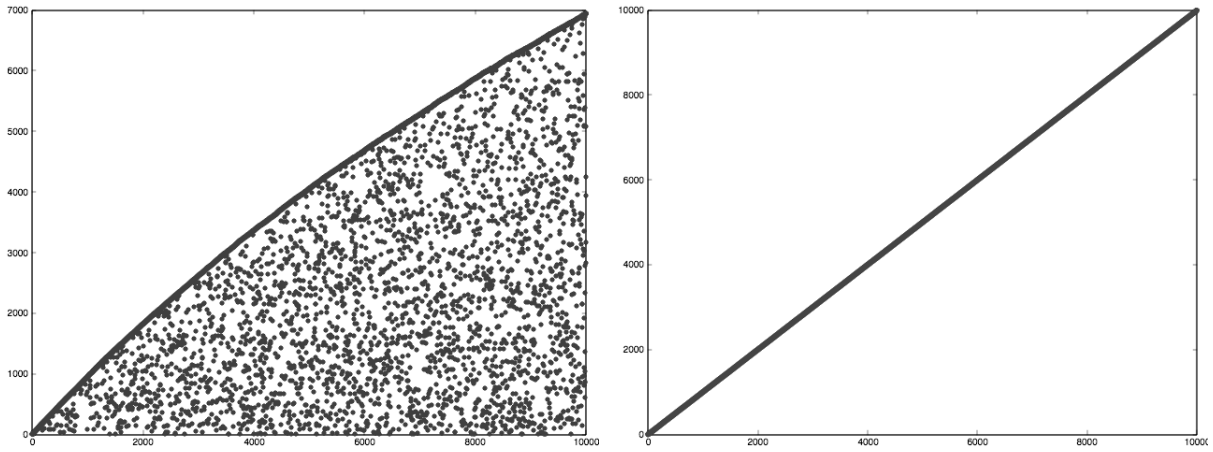
As discussed in earlier sections, the Dirichlet process exhibits the clustering property, and clustering data using mixture models is the most common application of Dirichlet process. The model is based on the assumption that there is an infinite number of possible clusters, a finite number of which is are to fit the observed data. The posterior provides a distribution over the number of clusters, the assignment of data to clusters, and the parameters associated with each cluster. Thus, unlike traditional parametric mixture models, nonparametric models are capable of adapting model complexity, and identify



(a) $\alpha = 10$ (b) $\alpha = 100$



(c) $\alpha = 1000$



(d) $\alpha = 10000$ (e) $\alpha = 100000000$

Figure 3.2: Variation of number of elements in each cluster for different values of α. For smaller values of α, crowded tables attract more customers, whereas larger valuers of α yield a more uniform distribution of the customers across the tables.

the number of clusters K from the data.

Let $X = x_1, x_2, \ldots, x_n$ be a set of independent data points, and $\Theta = \theta_1, \theta_2, \ldots, \theta_n$ be a set of latent parameters drawn from G. Data points x_i associated with the same θ_i belong to the same cluster, $f(x_i|\theta_i)$ denotes their density, and F is the set of the densities. We obtain a hierarchical model of the form:

$$\begin{aligned} x_i|\theta_i &\sim F(\theta_i) \\ \theta_i|G &\sim G(\Theta) \\ G|\alpha, G_0 &\sim DP(\alpha, G_0) \end{aligned} \tag{3.15}$$

The plate model of Equation 3.15 is shown in Figure 3.3.

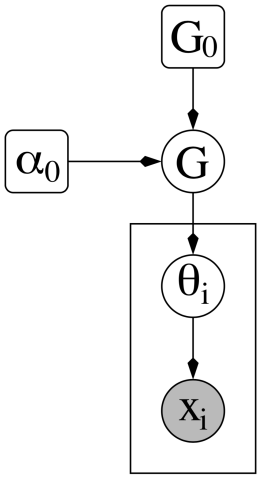


Figure 3.3: Plate Model for the Dirichlet Process Mixture Model (DPMM)

Inference in DPMM

The Bayesian Non-Parametric mixture model based on the CRP uses a generative probabilistic process of a collection of observed data. The basic step in the modeling is the computation of the posterior. The posterior distribution of the latent variables given the observations gives us a distribution of the data.

Markov Chain Monte Carlo (MCMC) methods are widely used for inference methods in Bayesian Models, and are popular for inference with Bayesian Nonparametric models as well. MCMC methods define a Markov chain on the hidden variables that has the posterior as its distribution. By drawing samples from this Markov chain, we can obtain samples from the posterior. Gibbs sampling is a simple form of MCMC sampling, where the Markov chain is constructed using the conditional distribution of each hidden variable given the other variables and the observations.

Gibbs sampling can be used with CRP mixtures effectively mainly due to the exchangeability property of CRP. According to the property, the order in which the customers sit does not affect the probability of the final distribution. Therefore, each observation can be considered to be the last one and Equations 3.11 and 3.12 can be used as a term in the conditional distribution along with the likelihood of the observations under each partition. Using conjugate priors makes the inference process easy since the posterior has the same distribution as the prior.

Metropolis-Hastings is another method used for MCMC sampling. It is one of the easiest methods for handling non-conjugate priors. For a survey of Gibbs Sampling and other MCMC methods for CRP mixture models, see [25, 48].

MCMC methods, although guaranteed to converge to the posterior with enough samples, have two limitations:

1. MCMC methods can be slow to converge and may require many iterations before convergence, especially for large datasets.

2. It is difficult to assess convergence.

Variational inference [14] is a popular alternative. This approach is based on the idea of approximating the posterior with a simple family of distributions and searching for the closest member of that family. While variational methods are not guaranteed to estimate the true posterior, they are faster than MCMC and it is easier to assess convergence.

Both MCMC and variational strategies for posterior inference enable searching the space of models and finding optimal parameters simultaneously using the data provided. In addition to inferring the posterior, MCMC can also be used to estimate or update the hyper-parameters including the concentration parameter α and the base distribution. Gorur et.al [33] discuss the estimation of the base distribution. [48] presents a deeper discussion on estimation of hyper-parameters.

3.3.4 Hierarchical Dirichlet Processes

A hierarchical Dirichlet process [71], is a nonparametric Bayesian approach to modeling grouped data. Each group is associated with a mixture model, and the Hierarchical Dirichlet process links these mixture models. The model enables sharing of mixture components between groups. Similar to a Dirichlet Process Mixture Model, we define a Hierarchical Dirichlet Process prior, which can be used to group data. The data is assumed to be subdivided into a set of groups, and we wish to find clusters within each group to capture latent structure in the data belonging to that group.

In order to illustrate the problem better, Teh et. al [71] provide an example from the field of information retrieval. One common problem in IR is modeling the relationships

among documents, and the bag of words approach is widely used. This problem is also referred to as topic modeling [13], where the words in a document belong to a number of topics or latent clusters. The topics are modeled as a probability distribution on the words in the dictionary. Several documents could share the same topic. For instance, for articles from a local newspaper the topics may be 'NC State', 'Wolfpack', 'Raleigh', 'Election', 'Hurricane'. At the same time, articles from a newspaper in another location could be assigned topics such as 'Election', 'Hurricane', 'Football'.

The process defines a set of random probability measures G_j, one for each group, and a global base distribution or random probability measure G_0. The global measure G_0 is distributed as a Dirichlet process with concentration parameter γ and base probability measure H. Essentially, the hierarchical Dirichlet process is aimed at modeling the base distribution in a Dirichlet process.

$$G_0|\gamma, G_0 \sim DP(\gamma, H) \tag{3.16}$$

The random measures G_j are conditionally independent given G_0, with distributions given by a Dirichlet process with base probability measure G_0:

$$G_j|\alpha_0, G_0 \sim DP(\alpha_0, G_0) \tag{3.17}$$

Thus, there are three hyperparameters for the hierarchical Dirichlet process: baseline probability measure G_0, and the concentration parameters γ and α_0. The baseline measure H provides the prior distribution for the factors θ_{ji}. The distribution G_0 varies based on the prior H, and γ gives the amount of variability. The actual distribution G_j over the factors in the jth group deviates from G_0, with the amount of variability governed by α_0.

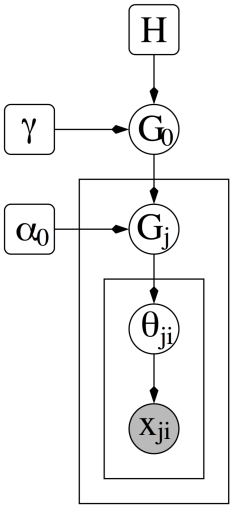


Figure 3.4: Hierarchical Dirichlet Processes Mixture Model (HDPMM)

A hierarchical Dirichlet process can be used as the prior distribution over the factors for grouped data. For each j let $\theta_{j1}, \theta_{j2}, \ldots$ be i.i.d. random variables distributed as G_j. Each θ_{ji} is a factor corresponding to a single observation x_{ji}. The likelihood is given by:

$$\begin{aligned} \theta_{ji}|G_j &\sim G_j \\ x_{ji}|\theta_{ji} &\sim F(\theta_{ji}) \end{aligned} \tag{3.18}$$

This likelihood defines a hierarchical Dirichlet process mixture model and the graphical model is shown in Figure 3.4.

The hierarchical Dirichlet process can be extended to more than two levels by specifying more hyperparameters. The base measure H can be drawn from a DP, and more levels of the hierarchy can be added recursively.

The hierarchical Dirichlet process extends the Dirichlet process, and exhibits the clustering property. The HDP is used extensively in topic modeling applications, where the number of topics can be unbounded and are learned from data. This is similar to Latent

Dirichlet allocation [15], and the HDP mixture model is a nonparametric generalization.

The Chinese restaurant process analog for hierarchical Dirichlet processes is referred to as the Chinese restaurant franchise. In the Chinese restaurant franchise, the CRP metaphor is extended to allow multiple restaurants to share a set of dishes. In a restaurant franchise with a shared menu across the restaurants, at each table of each restaurant one dish is ordered from the menu by the first customer to sit at the table, and all subsequent customers who sit at that table share that dish. Multiple tables in multiple restaurants can serve the same dish. In the hierarchical Dirichlet process, the values of the factors are shared both between the groups as well as within the groups.

Teh et. al [71] describe three related MCMC sampling schemes for the hierarchical Dirichlet process mixture model for inferring the posterior.

1. Gibbs sampler based on the Chinese restaurant franchise

2. An augmented representation involving both the Chinese restaurant franchise and the posterior for G_0

3. A variation on the second sampling scheme with streamlined bookkeeping, called Posterior sampling by direct assignment

Each of the three techniques has its merits and limitations, in terms of ease of implementation and convergence speed. While sample by direct assignment is easier to implement, the Chinese restaurant franchise based approaches offer improved performance on convergence speed.

Thus, a hierarchical Dirichlet process can be used for effectively modeling groups of data, where each group is characterized by a mixture model and mixture components can be shared between groups.

Chapter 4

Similarity Based Multilevel Model

Unseen or rare examples, within-class variations and inter-class similarities have been a major topic of discussion in recent years and a variety of approaches have been identified to work under different conditions. However, lacking are approaches that can deal with all three problems at once. In this section, we discuss an approach to regression tasks called Similarity Based Multilevel Model (SBMM), in which the training examples are first grouped by similarity and a group of heterogeneous models is applied to each of these groups. This approach enables better prediction on unseen or rare examples when compared to existing approaches. Further, our approach demonstrates an application of multilevel modeling to complex data. Our approach was found to provide better prediction than common approaches on rare unseen examples without affecting the overall performance on prediction tasks.

4.1 Introduction

In our approach, we constructed a 2-level model as shown in Figure 4.1. Consider an input dataset $D = \{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}$ where $X = \{x_1, x_2, \ldots x_n\}$ is the set of examples and $Y = \{y_i, y_2, \ldots, y_n\}$ is the set of target values where $y_i \in \mathbb{R}$.

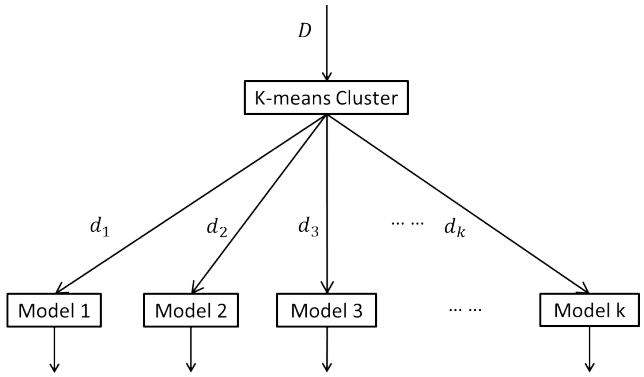


Figure 4.1: Similarity Based Multilevel Model (SBMM)

At the first level, the dataset is divided into k clusters $d_1, d_2, \ldots, d_k$. Clusters so formed contains examples that exhibit similar properties or attributes. Thus, two examples $\langle x_m, y_m \rangle$ and $\langle x_n, y_n \rangle$ are placed in a cluster d_i if their attributes x_m and x_n are similar. In k-means, the dissimilarity is measured using the Euclidean distance between two examples. Thus similar examples have smaller distances compared to dissimilar examples. In the reminder of this section, we discuss how each of these clusters is used to train a mixture of heterogeneous models and how such a model can be used to predict the target value $\hat{y_i}$ for an unseen example $\langle x_i \rangle$.

When we are presented with an example $\langle x_i \rangle$ for prediction, we assign it to its nearest cluster d_i and use the corresponding model to approximate the target value $\hat{y}$.

4.1.1 Algorithm

Let us consider the training set $D = (X, Y)$. We obtain k clusters $d_1, d_2, \ldots, d_k$ as the first step. Each of these clusters is used as a training set for a mixture of heterogeneous models M_i, where $i = 1, 2, 3, \ldots, k$. For each example $\langle x_m, y_m \rangle$ in cluster d_i, the model M_i predicts the target value $\hat{y}$.

The performance of each model M_i is measured using the Root Mean Squared Error (RMSE) value. If there are n data points in a cluster d_i,

$$RMSE_{M_i} = \sqrt{\frac{\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}{n}} \tag{4.1}$$

The choice of k is critical for this approach. We choose an appropriate value of k such that

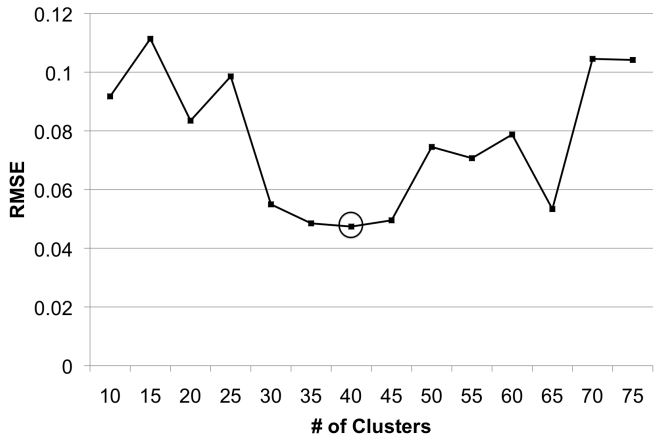
1. the overall RMSE for SBMM is minimum during validation;

2. the resulting model M_i has an acceptable balance of bias and variance.

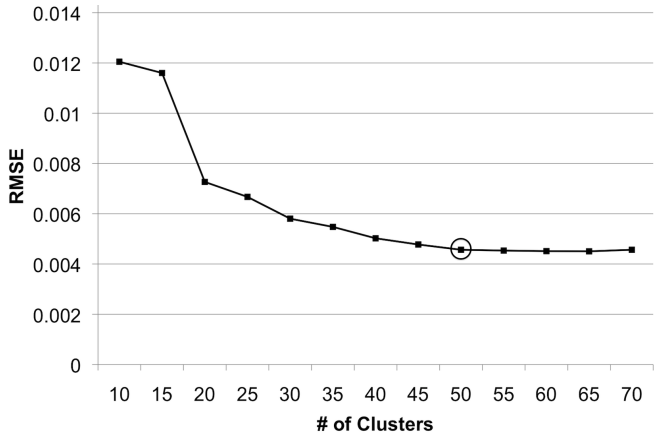
Ideally, we would prefer that the within-cluster sum of squared error (SSE) be small and would choose the value of k corresponding to the knee of the curve of k plotted against SSE [70]. However for more complex datasets, such a simple choice of k is either not possible or ineffective. In such datasets we identify a value of k such that it corresponds to a low value of RMSE during validation.

Several approaches have been developed over the years to determine the value of k in order to find natural clusters in data. These include using hierarchical clustering [40], information theoretic approach [69], silhouettes [70], elbow method [70], and cross-validation . Each of these approaches have their respective advantages and limitations.

The goal of the multilevel model is ultimately to achieve good approximation on



(a) Value of 'k' corresponding to the lowest RMSE



(b) Value of 'k' corresponding to the 'knee' of the curve

Figure 4.2: Two types of curves for RMSE vs Number of clusters

unseen samples and not to identify clusters. Therefore, we extend the cross-validation-based approach so that the number of clusters is guided by the cross-validation RMSE from our heterogeneous mixture of models since our training set is labeled.

We start with $k = 1$ and hierarchically increase the number of clusters until one of the following two cases occurs.

1. In some datasets, the value of RMSE obtained during validation initially decreases

with increase in k before RMSE either starts increasing or shows irregular behavior as shown in Figure 4.2a. A good choice of k is the value corresponding to the lowest RMSE.

2. In some datasets, the value of RMSE generally decreases with increase in k as shown in Figure 4.2b. However, beyond a certain point, the RMSE remains a constant. We stop at the "knee"; that is, the point after which RMSE becomes a constant.

As one would expect, a very high value of k results in fewer data points in each cluster d_i, resulting in a model with high bias and low variance. The two cases described above ensure that the model does not overfit and ensures a good balance between bias and variance.

For a new example $\langle x_i \rangle$ that has been assigned to its nearest cluster d_i, we use the trained model M_i to approximate the target value $\hat{y}_i$. The error in prediction of unseen examples is measured as $|(y_i - \hat{y}_i)|$.

One practical concern is feature selection and dimensionality. Each of the datasets we discuss has a different number of attributes. For data with high dimensions, we observed that the presence of highly correlated predictors affected the performance. In order to avoid such predictors, we applied Relief attribute selection [63] in order to remove correlated attributes.

Another concern is time taken to train the model. As one would expect, the time taken to train the model depends on the number of clusters. Figure 4.3 shows the average trend for the time taken to train the model. The major contributing factor to this time is the time taken to divide the datasets into clusters. As k increases, the number of examples per cluster decreases and the average training time of the heterogeneous classifiers also decreases.

Figure 4.3: Time taken to train the model with increase in number of clusters.

4.1.2 Heterogeneous Models

SBMM uses a mixture of heterogeneous models for the prediction task. As we shall see later, the mixture of models with SBMM performs better than each of the models taken individually and also better than other approaches to regression.

When using a mixture of models, the choice of combining results from the different models is a major task. Various approaches to combining heterogeneous models models for classification tasks have been identified [8]. In addition, approaches for multilevel mixture of models have also been developed [37, 38, 47].

For a regression task, we use the the validation error for combining the results. The aim of regression is to reduce the validation error. Now, the target value for an example from the test set is predicted as a weighted average of outputs from different models. There are 6 weights, denoted as w_i, one for each of the six models and the weight w_0 corresponds to the model with the lowest RMSE and w_5 corresponds to the one with the highest RMSE. The weights w_i are assigned such that:

$$w_i = \begin{cases} 0 & \text{for} \quad RMSE_i >= H \\ e^{-i} & \text{for} \quad RMSE_i < H \end{cases} \tag{4.2}$$

where, $RMSE_i$ is the error of the model corresponding to ith smallest error value and H is the harmonic mean of the errors. Harmonic mean is used since it avoids the effect of models with large error values and gives preference to the models with the lower error values in the mixture of models. For a model with the lowest error, the weight is $w_0 = 1$ and other models with error less than the harmonic mean get weights $w_i < 1$.

The above step normalizes for high bias and results in the best approximation for any data instance. This is in a sense similar to majority voting strategy used for combining results in classification [8], where a majority vote determines the class of an unseen example. In our approach, a set of models with low error values for a particular cluster determines the target value of an unseen example that is assigned to the cluster.

4.1.3 Within-Class Variations, Inter-Class Similarity and Rare Examples

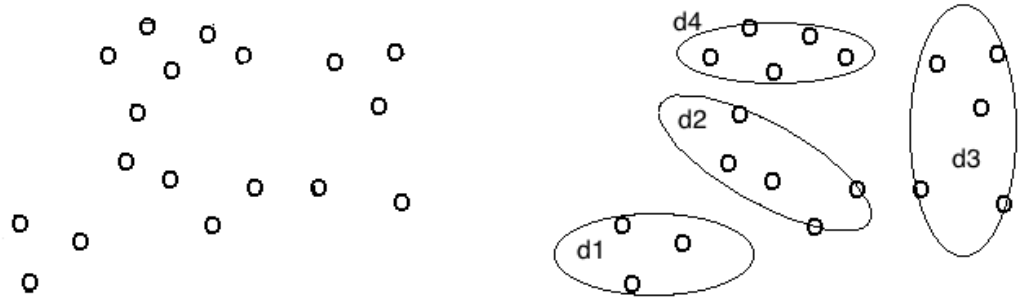
The three complexities – Within-Class Variations, Inter-Class Similarity and Rare Examples – pose issues in regression. In classification tasks, approaches to solve each of the three problems have previously been identified, but no one has described a overall solution in the presence of all three problems together.

In case of regression tasks, this problem has received even less attention. Yet there are cases when predicting target values of unknown or entirely unseen examples becomes a major task. In most of these cases, the examples belong to some category but their behavior tends to resemble examples of another category. Examples of such problems include

predicting toxicity of chemicals and predicting the market price of a new commodity. Our approach is an attempt to solve all three of our problems in regression tasks.

When we cluster the data, similar examples are grouped together. At this point, even examples that are similar across different categories are grouped into the same cluster. Thus, instead of avoiding the Within-Class Variations and Inter-Class Similarity, we utilize the presence of the two complexities. Now, when we train our mixture models on each of these clusters, the difference between examples actually improves learning.

Given the dataset in Figure 4.4a, the output at the end of clustering is shown in Figure 4.4b where d_i represents the clusters. The mixture of models learns from each cluster. When a new datapoint is given to SBMM, the data point is assigned to its closest cluster and the mixture of models corresponding to that cluster predicts the value for the point.



(a) Dataset before clustering (b) After clustering with $k = 4$

Figure 4.4: Data Clusters

The cluster structure obtained in our approach need not correspond to the natural groups that exist or to the categories in data. A rare example has features dissimilar to examples from the same category. When clustering, the rare example is assigned to a cluster based on similarity. The model corresponding to that cluster predicts the target

value of the example. As one can observe from our results in Section 4.2, this approach is not fail safe and there are some wrong predictions.

4.1.4 Comparison to Local Models

The general behavior of the SBMM is similar to that of local regression models such as Locally Weighted Regression [20, 45]. Locally Weighted Regression is a non-parametric approach and retains the training data for use each time a prediction is made. This makes it a memory-based, lazy learning method. The model performs regression around a point of interest using only training data that are "local" to that point. The clustering step in SBMM simulates the "local" effect in the model. However, SBMM uses a heterogeneous mixture of models and the model is pre-trained.

Also, locally weighted regression, especially *loess* [20] is a linear regression model. Such models are not very effective in handling nonlinearity that occur locally. Further, research has shown the effectiveness of multilevel mixture models for tasks involving more complex data [47].

4.2 Experiments and Results

In the previous section, we discussed the working of our approach using the sample application of toxicity prediction. In this section we present experimental results for the Similarity Based Multilevel Model. We evaluated the approach against different datasets and compared the performance with existing approaches to regression.

First, we show that the performance is similar to existing approaches and significantly better than most approaches in many cases. Then, we present results showing how our approach was able to provide improved approximation on rare examples over existing

approaches.

4.2.1 Toxicity Prediction

In this section we present an application of our approach to predicting chemical toxicity. Our discussion is based on the datasets provided by Cadaster [18] for the Toxicology Prediction Challenge as a part of 19th International Conference on Artificial Neural Networks (ICANN), 2009. The aim of toxicity prediction is to build *in silico* models that can predict the environmental toxicity of chemicals, measured against a micro-organism. The growth inhibition of ciliated protozoan *T. pyriformis*, $log(IGC50)^{-1}$ [72, 79], was used to represent the toxicity.

The dataset consisted of around 2250 attributes each describing various properties of various organic chemicals. Using these attributes, the task was to estimate the toxicity of various chemicals. The dataset consisted of both aromatic and aliphatic chemicals. Within each, some chemicals belonged to a distinct congeneric group such as alcohols, acids and amines while some others contained a mixture of these groups.

We split the entire dataset into clusters using simple k-Means. For each of the clusters obtained, we trained the heterogeneous mixture of models. For both the validation set and the test set, we determined the cluster to which each molecule belongs. Then using the corresponding model for that cluster, we predicted the toxicity value of each molecule in the test set. The number of clusters was chosen to be $k = 10$ since training on these clusters had the lowest error values for the given data.

Existing approaches to regression use the dataset as it is. Alternatively, due to the categorical nature of the dataset, one might take each chemical class and predict the toxicity for each class independent of the others, since the behavior of compounds is

thought to be closely related to their chemical classes. However, as we discuss in the next section, both these methods performed no better than our approach.

Using the models built on the training set, we evaluated the test set. The challenge in the test set was that many compounds were complex and consisted of a mixture of chemical classes that were previously unseen.

The best results where obtained when the number of clusters was $k = 10$. So, with the 10 models built, one for each cluster, we determine the best cluster for each of the test examples and apply the corresponding model. The results for the best RMSE values obtained by these approaches are shown in Table 4.1.

Since the application was based on the toxicity prediction challenge, we consider it meaningful to compare our results against the top results from the competition. It must be noted that our approach was not a part of the competition.

As can be observed, out results were non-significantly different from the method with lowest RMSE. However, our approach closely approximates more rare examples than other approaches.

Table 4.1: Comparison of RMSE values for Toxicity Data

Approach	Validation	Test Set
Ridge regression	0.353	0.741
Gaussian Processes	0.41	0.756
Least Squares SVM	0.395	0.765
SBMM (our approach)	**0.408**	**0.766**
SMO Reg	0.424	0.789
Neural Network	0.4	0.794

Table 4.1 also shows the values of other top contestants in the challenge obtained from Cadaster. One interesting fact was that most contestants used an ensemble or combination of different models. Some contestants also grouped the data based on different criteria. These are similar to our approach.

However, it must be noted that the contestants were permitted to use their own descriptors for the competition apart from those supplied in the datasets while we used only the attributes provided in the dataset.

4.2.2 Other Datasets

Apart from toxicity prediction, we evaluated the performance on other datasets as well.

We used the following datasets from the UCI repository [5]: Auto, Communities and Crime, Parkinson's Tele-monitoring, and Wisconsin Breast Cancer (WPBC). The Bank and PumaDyn datasets were obtained from the Delve Repository [59]. Table 4.2 below presents a description of the datasets:

Table 4.2: Description of Datasets

Dataset	# of Attributes	# of examples
Abalone	13	4177
Auto	26	205
Bank-8FM	8	8192
Bank-32nh	32	8192
Crime	128	1994
PumaDyn	32	8192
Tele-monitoring	22	5875
WPBC	32	198

Each of the datasets have different characteristics. Datasets from the Delve repository exhibit non-linearity and some datasets, such as the Community and Crime Data, contain missing values.

4.2.3 Experiments

We present a comparison of our approach with the results obtained from standard techniques to regression. We compare our results with SVM (with polynomial kernels), Gaussian process (with polynomial kernels), Regression tree (M5P), Ridge regression and Local Regression (LOESS).

Table 4.3 shows the results of our experiments. The RMSE values shown are validation errors. During a validation run, we compute the error as predicted by the the models corresponding to the cluster assignment for each example. The numbers reported are the overall RMSE obtained for n examples during validation. While Table 4.3 shows the raw error values, Table 4.4 shows the normalized error values for easy comparison. The values are normalized with respect to the performance of SBMM.

The choice of k depends on the overall prediction error, bias and variance in the models. Overall, we observed that an increase in value of k resulted in a decrease in the RMSE for prediction. However, beyond a certain value of k, the error either increases or shows irregularities. Figure 4.2 shows graphs depicting the variation of error with increasing value of k.

Earlier, we described the mixture of heterogeneous models we used in our approach. From this list, we compared the performance of individual models with SBMM. Our approach showed significant improvement in performance compared to the models taken individually. This empirically justifies our choice of mixture models against using a single

model for each cluster.

Table 4.3 shows good performance on such data as well. The values shown are not normalized. For example, in the Auto imports dataset, the task was to predict the cost of the vehicle and the values are in range of 5000 to 36000, hence the high error value.

Table 4.3: RMSE on UCI and Delve regression datasets: Comparing the performance of our approach with other approaches to regression

Dataset	k	SBMM	Ridge	Regression Tree	Gaussian Process	SVM-Poly	Local Regression
Abalone	120	0.803	0.707	1.819	1.650	1.504	1.332
Auto	10	746.940	1124.785	1528.613	1705.278	1795.231	1702.063
Bank-8FM	150	0.022	0.032	0.030	0.690	0.040	0.025
Bank-32nh	180	0.077	0.660	0.750	0.640	0.689	0.630
Crime	40	0.033	0.054	0.128	0.124	0.132	0.062
PumaDyn	50	0.004	0.005	0.007	0.024	0.027	0.007
Tele-monitoring	120	0.017	0.030	0.035	0.047	0.045	0.045
Toxicity Data	120	0.408	0.353	0.400	0.410	0.395	0.400
WPBC	7	0.730	0.704	2.832	2.790	2.985	1.540

Table 4.4: Normalized Error: error values normalized with respect to the performance of SBMM

Dataset	k	SBMM	Ridge	Regression Tree	Gaussian Process	SVM-Poly	Local Regression
Abalone	120	1	0.8804	2.2653	2.0548	1.8730	1.6588
Auto	10	1	1.5059	2.0465	2.2830	2.4034	2.2787
Bank-8FM	150	1	1.4545	1.3636	31.3636	1.8182	1.1364
Bank-32nh	180	1	8.5714	9.7403	8.3117	8.9481	8.1818
Crime	40	1	1.6364	3.8788	3.7576	4.0000	1.8788
PumaDyn	50	1	1.2500	1.7500	6.0000	6.7500	1.7500
Tele-monitoring	120	1	1.7647	2.0588	2.7647	2.6471	2.6471
Toxicity Data	120	1	0.8652	0.9804	1.0049	0.9681	0.9804
WPBC	7	1	0.9644	3.8795	3.8219	4.0890	2.1096

Table 4.5: Number of rare examples correctly predicted within 1% error margin

Dataset	# new	SBMM	Ridge	Regression Tree	Gaussian Process	SVM-Poly	Local Regression
Abalone	120	105	89	50	61	60	80
Auto	15	12	9	3	2	3	8
Bank-8fm	120	112	85	87	53	73	85
Bank-32nh	120	104	75	40	56	60	80
Crime	120	111	87	32	36	46	60
PumaDyn	120	116	98	98	78	68	96
Tele-monitoring	120	114	91	86	54	56	90
Toxicity Data	120	105	105	45	76	60	60
WPBC	15	12	8	7	7	6	7

4.2.4 Performance on Rare Examples

The purpose of SBMM is to provide the best approximation for rare examples in the presence of different variations and similarities in the data without reducing the overall performance on regression.

In the previous sections, we discussed how our approach was significantly similar in performance to existing approaches. In this section, we present the performance on rare examples.

Table 4.3 shows the error rates for different datasets. The error rates for a few datasets are significantly lower than other approaches. This gives us an indication of overall performance and one would expect the performance on rare examples to be good as well. Table 4.5 confirms our expectation. We present the number of examples that were correctly predicted within a 1% margin of error for each regression model.

None of the datasets have an annotation of which examples are rare or new. So, Table 4.5 is an estimate of the performance. We identified the rare examples or cases where the example is a rare representative of its kind using our definition of rareness from Section 2.3.1. This takes frequency of occurrence of the sample and how different the data point is from the rest of the dataset to compute the rareness of the dataset. The test set was obtained by both random sampling and by hand picking rare cases from the dataset.

The following is a description of how we determined whether an example is rare. The Toxicity data [18] consisted of a pre-existing test set consisting of 120 examples. All these were previously unknown or unseen. The examples also included chemicals whose congeneric classes were not present in the training set (e.g., the chemical class Ester). In the Crime data, examples corresponding to certain states and/or communities were considered rare since they were not represented when training. In the Auto imports data,

the type of car and a few of its features were provided. For this dataset, an unknown example would be one that corresponds to an unknown body type for a particular brand. This simulates a real life case where we are required to predict the target price of a new car in the market. In the Abalone dataset, a class distribution was already given.

While it is easy to identify rare examples in data consisting of categorical information, it is not as easy in datasets consisting only of real valued attributes. In such datasets, we considered a set of attributes which could be easily divided into disjoint ranges. Each of such groups of value ranges were treated as categories. For example, in the WPBC dataset, the Lymph Node Status was considered to provide categorical information since it was integer valued and the values ranged from 1 to 20. For the PumaDyn, we considered groups of values of angular positions as categories. We used ranges of time since recruitment into the trial for the Tele-monitoring dataset.

4.3 Discussion

We have presented a simple multilevel model that can learn well in the presence of rare examples. The model can also correctly predict the target value of higher number of rare examples compared to traditional approaches. The comparisons shown above illustrate how a multilevel model can be effective in learning in the presence of imbalance compared to non-multilevel models.

The model presented above uses a clustering based approach which is very similar in construction to a variety of existing models for transfer learning [4, 10, 66] and local models [20, 45]. The major difference here is in the choice of cluster size and how it attempts to reduce over-fitting. As mentioned earlier, the clusters are formed by hierarchically splitting each cluster at each time step until model M_i corresponding to the

cluster i has the least validation error. Also, we ensure that each cluster has a significant proportion of the data in it. These ensure that there are enough examples in each cluster, thus adding more variance to learn effectively.

One important observation to make here is that the cluster structure obtained in our approach need not correspond to the natural groups that exist or to the categories in data. For example, in the chemical toxicity dataset, each cluster need not consist of examples that belong to the same congeneric class. Rather, the cluster members are similar in behavior. While some applications benefit from a hierarchical structure that corresponds to the natural hierarchy, our approach enables learning in the presence of imbalanced data, an advantage which many existing approaches do not offer.

While we have empirically shown the performance of the model and briefly described the intuition behind why the model works, more work is required to make this approach better. The notion of rare example is considered to be subjective by some researchers. A more specific definition of imbalance due to rare example will greatly aid further research. Further, we must explore other methods for identifying or selecting rare cases such as perplexity.

Further, we must study the effects of feature selection on prediction with rare data, and more so in the presence of imbalanced data. Our model does not perform explicit feature selection and we believe that improving the set of features used in the final prediction can affect overall performance.

Various approaches are being developed to learn in the presence of less data. One-shot learning [66] is one such genre of approaches where an entire concept is learned from one example given enough domain knowledge. A rare example could be that first example presented to learn a concept. On the other hand, presence of rare examples in such learning settings can make learning harder [61]. It is therefore important to further

explore the effects of rare examples in prediction problems and develop approaches that can improve accuracy.

Chapter 5

Classification Models for Learning Rare Data

Models for learning in the presence of rare data have mainly been restricted to learning imbalanced data. As discussed in earlier chapters, the concept of rareness extends well beyond imbalance. There are several approaches, with different applications and varying degrees of success to learn imbalanced data.

Most of these techniques are based on the traditional notion: In a two-class classification problem, given two classes C_1 and C_2, the number of examples of C_1, represented as $|C_1|$, is significantly more than the number of examples of class C_2, represented as $|C_2|$.

Further, approaches for learning with rare or imbalanced data often account for the frequency of occurrence of examples only. Similarity between examples based on the features known at the time of classification are more important. An example is rare not only if it is infrequent, but also when its features are different from those seen earlier. Also, a rare example can belong to an existing class or can be a representative of an unseen class. When learning, it is important to consider all of these factors as well.

In this chapter, we describe an approach to learning with rare data that takes into account the definition of rareness from Section 2.3.1. We are concerned with learning new or rare concepts so that future predictions on such concepts is better, given sufficient domain knowledge, or examples of related examples. In Section 5.2, we discuss the approach based on similarity based learning that extends the nonparametric bayesian approach discussed in Chapter 3.

Our approach can be extended to other learning settings including one-shot learning and learning with imbalanced data. We can also extend the approach to hierarchically classify data, which as we shall discuss, is a more powerful technique to learning new information. We discuss all of these in Section 5.3.

5.1 Introduction

In the previous chapter, we discussed an approach to learning based on similarities which was mainly applied to predicting continuous valued outputs for unseen examples. It turns out that a similar approach performs poorly in case of a classification problem. The issue arises when handling new examples belonging to classes different from already known classes.

5.1.1 Approaches to Classification with Rare Examples

There are several ways of modeling a classifier when learning data with rare examples. Each of the approaches has its advantages and limitations, and can be applied to solve different problems as appropriate.

Learn the rare example as an member of an existing class

Ideally, we would prefer that a rare example belonging to class k be learned as a member of the same class. Most traditional classification algorithms tend to take this approach.

While simpler to learn, the resulting model has higher variance and can result in higher misclassification rate.

Learn the rare example as a member of a new class

In some applications, it is advantageous to learn the rare example x_i as a new class k', or at least as a member of a subclass of an existing class k. Assigning a new class label will enable us to recognize that the new example is different from the other examples in the class.

If our classifier has the ability to classify hierarchically, we can classify x_i as a subclass of an existing class k. Such a classification is very powerful since it carries more information about the examples and is easier to expand in the future. The only limitation of hierarchical classifiers is that obtaining class labels is expensive and requires an expert labeler. Active learning or crowd-sourced labeling are often used in such cases.

Ignore the rare example

Rare examples often resemble outliers or anomalies. Like anomalies, rare examples differ from other examples in the dataset, and could be ignored or discarded when curating the dataset, either intentionally or by mistake.

5.2 Similarity Based Classification Model

Consider a classification problem: Given a dataset D with N labeled training examples. Each example belongs to one of K classes, and each class $k \in \{1, \ldots, K\}$ contains n_k labeled examples.

Assuming we have built a model to fit the dataset D with N labeled examples, our task is to learn the new labeled training example (X_{n+1}, k). Ideally, the new training example X_{n+1} will belong to any of the existing classes, i.e., $k \in \{1, \ldots, K\}$.

$$D = (x_1, k_1), (x_2, k_2), \ldots, (x_n, k_n) \tag{5.1}$$

However, if X_{n+1} is rare, its addition to an existing class could potentially increase the classification error for that class. In such cases, it might be wise to reassign the example with a new class label k'. In Section 5.2.1, we describe an algorithm based on Dirichlet Processes that can learn the new rare example.

Reassigning the class label brings up an interesting problem. We lose the information that both classes k and k' are the same. A hierarchical classification will help us solve both the problems:

1. Adding X_{n+1} will not affect the classifier accuracy
2. We will know that both k and k' are related, and the information about their relationship is preserved in the hierarchy.

In a hierarchical classification, two or more sub-classes belong to a super class, and all the classes have labels associated with them. In some datasets, the class hierarchy is known a priori. When such a hierarchy is unknown, the help of a human expert can be sought. We describe an algorithm in Section 5.3.2.

5.2.1 Algorithm

The algorithm consists of the following major steps:

For each example x_i in a pool of labeled examples:

1. Estimate the value of α. (see Section 5.2.2)

2. Select the class assignment based on the cluster assignment obtained from Dirichlet Process Mixture Model. (see Section 3.3.3)

$$\begin{aligned} x_i|\theta_i &\sim F(\theta_i) \\ \theta_i|G &\sim G(\Theta) \\ G|\alpha, G_0 &\sim DP(\alpha, G_0) \end{aligned} \tag{5.2}$$

3. Estimate the distribution over the class labels. (see Section 5.2.3)

   (a) probability of belonging to an existing class in the current model $P(k_{x_i} = k)$

   (b) probability of belonging to an unknown/ news class $P(k_{x_i} = k')$

4. Update the model with the new labelled example x_i.

   If the example gets a new class label, update the model accordingly.

These tasks are detailed in the following subsections. Estimation of the concentration parameter α is a major step in the algorithm. The value of α influences how the cluster assignments are handled. Since we are optimizing for learning rare classes, we prefer that the value of α is based on the degree of rareness defined in Section 2.3.1, and the value is sufficient to influence the assignment of a new cluster for the rare class. We discuss how to obtain α in Section 5.2.2.

5.2.2 Choosing the Concentration Parameter

The first steps in the learning task involve choosing the concentration parameter.

Typically, the concentration parameter is set a priori, or a hyperprior is defined over α. A Gamma distribution, $\Gamma(a, b)$ is a typical hyperprior, and the value is estimated using Gibbs sampling [25].

In our approach, however, the concentration parameter is defined based on the characteristics of data. In particular, we define the value of α using the scheme from Section 2.3.1. Recall from Section 2.3.1 that we use the following equation to describe the degree of rareness.

$$R_i = \delta_i^m + (1 - F_C)^n$$

We derive the value of α from R_i. The higher the rareness factor, the greater will be the value of α. As the value of α increases, the Dirichlet process prior favors new clusters to be formed, each used to represent a rare example.

For values of R_i, where $0 \leq R_i < \theta$, where θ is the threshold, or tolerance of rareness in our system, we will restrict ourselves to use a simple clustering scheme where $\alpha = \Gamma(a, b)$. For values $R_i \geq \theta$, α will take larger values.

$$\alpha = \begin{cases} 1 & \text{for} \quad 0 \leq R_i < \theta \\ R_i & \text{for} \quad R_i \geq \theta \end{cases} \tag{5.3}$$

5.2.3 Classification

We start by computing the probability that the given example x_i belongs to an already known class k or a new class k'. Computing the latter is harder since the example is new and nothing much is known about the new class.

A generative model with a Dirichlet Process prior can be used to solve this problem. Dirichlet processes are used for density estimation and topic modeling. But unlike typical clustering algorithms such as k-means, a Dirichlet Process based clustering does not require a pre determined number of classes.

As we may recall from Chapter 3, Dirichlet Process is denoted as $DP(\alpha, G_0)$, where α is the concentration parameter and G_0 is its base measure. Given the problem of learning from a rare example, we could either add the new example to an existing class or assign a new class label to the rare example, as appropriate. Such a decision is made easy with the use of the Dirichlet process, and more specifically, the marginal posterior: the Chinese Restaurant Process (CRP).

Each example in the dataset is likened to a customer arriving at a restaurant, and the restaurant is assumed to contain infinite number of tables. All customers sitting at a table are expected to share the same dish. The table is analogous to a cluster and the dish to a class label.

Assuming there are already some customers dining at a few tables, when a new customer arrives at the restaurant, the customer has 2 options:

1. Sit at one of the already occupied table and share the dish being served to the table with a probability

$$\frac{\alpha}{n-1+\alpha} \tag{5.4}$$

2. Occupy a new table, and eat a new dish.

$$\frac{n_k}{n-1+\alpha} \tag{5.5}$$

Where α is the concentration parameter, n_k is the number of different tables occupied

after n customers have arrived. Note that $1 \leq n_k \leq n$.

This choice is similar to the two options our approach requires for class assignment of rare examples.

That is, a customer may sit at a previously occupied table with a probability proportional to the number of customers already sitting at it, or at a new table with probability proportional to the concentration parameter.

For the dataset D, set of classes K, the probability that an example x_i belongs to a class k or a new class k' can be computed using the Dirichlet process mixture model from Section 3.3.3 and the posterior defined in Section 3.3.2:

$$p(\theta_n|\theta_1, \theta_2, \ldots, \theta_{n-1}) \sim \frac{1}{\alpha + n - 1}(\alpha G_0 + \sum_{j=1}^{K} n_k \delta_{\theta_i})$$

The probabilities $P(k_{x_i} = k)$ and $P(k_{x_i} = k')$ are computed using Gibbs Sampling as described in [48].

5.2.4 Experiments and Results

The similarity based classification model is intended to learn a dataset with rare examples. The model should not only correctly predict future examples from rare classes, but should also have good predictive accuracy on non-rare class examples.

We used a few of the most well known datasets from the UCI repository to demonstrate the performance of our approach, as compared to other approaches. Table 5.1 shows the list of the datasets used in this work.

The datasets we have chosen are from different applications domains ranging from health care to optical character recognition (OCR). They have varying number of attributes, different number of examples, and different number of target classes. A few of

Table 5.1: Description of Datasets

Dataset	# of Attributes	# of examples	# of Classes
Abalone	8	4177	29
Breast Cancer Wisconsin	9	683	2
Pima - diabetes	8	768	2
Handwritten Digits Recognition	64	5620	10
Pen-Based Handwritten Digits	16	10992	10
Statlog (Landsat Satellite)	36	6435	6

the classes are under-represented. Two data sets, breast cancer and pima-diabetes, are binary data sets from the medical domain. Handwritten Digits Recognition and Pen-Based Recognition of Handwritten Digits are data sets containing the information of handwritings of numerical information from 0 to 9. The satimage data set contains the multi-spectral values of pixels in 3×3 neighborhoods in a satellite image. The Abalone dataset is based on the population biology of Abalone (Haliotis species) in Tasmania and the objective is to predict the age of abalone from physical measurements.

In Section 5.3, we use other publicly available datasets to demonstrate how our approach can be extended to other related learning settings such as one-class classification and hierarchical classification.

We have compared our approach against the performance of SVN, k-Nearest Neighbor, Random Forest and an Artificial Neural Network. We demonstrate the working of our approach using two experiments.

- Learn the model, and predict data similar to non-rare examples.

  The objective of this experiment is to show that the performance of the model is not worse than similar, traditional models. We train the model with the dataset, and evaluate the performance using cross-validation.

Table 5.2 shows the error rates for different datasets. This gives us an indication of overall performance and one would expect the performance on rare examples to be good as well.

- Use existing data as background knowledge, learn rare examples, and predict future rare examples.

  For the second experiment, we use the model learned from the first experiment, and train the model with rare examples. We choose the rare examples based on the definition from Section 2.3.1. We hold back a handful of examples from each rare class to be used for measure predictive performance. In datasets with a lack of rare examples, we simulate the effect by manually removing a sufficient examples from certain classes to simulate rareness. The results of the experiment are shown in Table 5.3. The extreme case of one-shot learning can also be demonstrated using this approach. We present a more in-depth discussion of one-shot learning later in Section 5.3.1.

Among the datasets, the Abalone dataset is noteworthy, primarily due to the distribution of the examples. It is a multi-class dataset with 29 classes, and is very imbalanced. The majority class has 689 examples whereas five classes have exactly one example in the training data. So, for the first experiment, we considered examples from classes with at least 100 examples. We did not use classes with exactly one example in either experiment. Samples from all remaining classes were used in the second experiment.

As we can see in Table 5.2, the Similarity Based Classification Model performs at least as well as other models when presented with a well-balanced dataset. In case of Abalone, which had a significant imbalance, SBCM outperformed the other models. This demonstrates how the model is able to handle class imbalance as well.

Table 5.3 shows the results from the second experiment. We provide the model with a small proportion of examples from the set of rare examples, and used the rest to test. For handwritten and pen-based digit recognition datasets, we treated examples from one digit as "rare". The table shows the results when the model was trained with 25% of the data, and the remaining 75% was used as the test set. SBCM is able to perform better than traditional models. In case of multi-class datasets, the training and examples consisted of samples from all classes, including a few from the majority classes as well.

Table 5.2: Precision and Recall on UCI and Delve Classification datasets: Comparing the performance of our approach with other common approaches

Dataset		SBCM	SVM	kNN	Random Forest	Neural Network
Abalone						
	Precision	0.790	0.651	0.682	0.735	0.705
	Recall	0.777	0.689	0.680	0.738	0.696
	$F-measure$	0.776	0.669	0.681	0.736	0.700
Breast Cancer						
	Precision	0.982	0.978	0.961	0.982	0.974
	Recall	0.965	0.972	0.956	0.965	0.963
	$F-measure$	0.974	0.975	0.958	0.974	0.968
Pima - Diabetes						
	Precision	0.769	0.769	0.735	0.742	0.748
	Recall	0.773	0.773	0.738	0.746	0.751
	$F-measure$	0.771	0.771	0.736	0.744	0.749
Handwritten						
	Precision	0.983	0.973	0.970	0.855	0.816
	Recall	0.973	0.953	0.978	0.845	0.821
	$F-measure$	0.978	0.963	0.974	0.850	0.818
Pen-Based						
	Precision	0.982	0.981	0.962	0.923	0.897
	Recall	0.980	0.987	0.955	0.905	0.877
	$F-measure$	0.981	0.984	0.958	0.914	0.887
Statlog						
	Precision	0.881	0.871	0.841	0.857	0.859
	Recall	0.879	0.875	0.831	0.854	0.850
	$F-measure$	0.880	0.873	0.836	0.855	0.854

Table 5.3: Learn a rare example and testing on more examples from the "rare class"

Dataset	$\#Train$	$\#Test$	SBCM	SVM	kNN	Random Forest	Neural Network
Abalone	53	150	0.720	0.507	0.640	0.640	0.620
Breast (W)	12	48	0.750	0.625	0.688	0.646	0.667
Pima	18	54	0.778	0.685	0.667	0.722	0.648
Handwritten	140	422	0.853	0.801	0.820	0.822	0.829
Pen-Based	270	830	0.884	0.865	0.863	0.869	0.871
Statlog	62	180	0.833	0.761	0.767	0.767	0.722

5.3 Extensions

The approach discussed in Section 5.2 can be extended to a variety of learning settings. In particular, we demonstrate three related problems:

1. **One-shot learning**

   One-shot learning is an extreme case of learning with rare examples. We are presented with exactly one example x to learn a concept or class C. It is reasonable to assume that we have some relevant prior knowledge learned from previous examples in the same problem domain. Traditional machine learning approaches have relied upon learning a concept from several positive examples. For a variety of reasons, the focus has recently shifted towards learning from fewer examples; the major reason being cost of labeling or annotating the examples.

2. **Hierarchical Learning**

   Hierarchical Learning is very powerful since it carries more information about the examples and is easier to expand in the future when compared to traditional classification approaches. Sub-class or parent-child relationships between examples can be captured through the hierarchy. Given the ability to obtain rich labeled data, hierarchical classifiers can learn meaningful information.

3. **Class Imbalance Learning** Extending a rare class classifier to learning with imbalanced data is straightforward since class imbalance is a kind of rareness where one entire class occurs less frequently compared to another class in the learning problem.

Since some of the datasets presented in Section 5.2.4 were also imbalanced, we restrict our discussion below to one-shot learning and hierarchical learning. We demonstrate the

extension of the similarity based approach to one-shot learning in Section 5.3.1 using hand gesture recognition as a case study. In Section 5.3.2, we discuss the extension of the model to hierarchical classification.

5.3.1 One-shot Learning – Case Study: Hand Gesture Recognition

We demonstrate the extension of our similarity based classifier for rare examples to one-shot learning with a case study of learning hand gestures. The dataset is from a recent challenge to develop a gesture recognizer for Microsoft Kinect[1]. The goal of the challenge was to use one-shot learning motivated by the ability of humans to learn from just one example compared to a machine, which requires several examples to generalize a concept. It must be noted that the approach and the results reported in this paper were not presented to the challenge.

The dataset from the challenge consisted of videos recorded by various volunteers performing different gestures spanning multiple domains including sports, dance, music, and sign language. A training set with labels for each gesture was provided. The challenge is to develop a model using the examples from this initial training set such that when a single new labeled gesture is provided, the model must learn the new gesture. When more examples of the new gesture are presented for labeling, the model must accurately recognize them.

Existing approaches to learning from few examples and one-shot learning operate on static images [66]. There has been much less related work on deep learning or one-shot learning architectures for video and multimodal input.

[1] `http://gesture.chalearn.org/`

Much like any learning problem, identifying appropriate features is a significant challenge in learning gestures. Learning and Recognizing gestures have been a topic of study for many years. Most commonly used approaches include Hidden Markov Models, Neural Networks, and others. A recent survey on the topic lists some interesting approaches [46]. The existing approaches have shown varying degrees of accuracy. However, most of these approaches are not applicable for learning, or generalization to occur with fewer examples, much less for one-shot learning. The approaches are based on traditional setting where the target and the source domains are typically drawn from the same distribution and the models less powerful in terms of generalization across domains.

Similarity Based One-Shot Learning

Our approach to learning the gestures is based on constructing a bag of features covering significant aspects of various gestures. The features include scale and pose invariant features such as HOG or Hu Moments for each frame, and the pixels of frames. Once the bag is constructed, each feature is associated with the set of gestures that contain the feature. When a new gesture is presented to the learning system, the best features contained in the gesture are identified. Whenever an unlabeled gesture is presented, the classification occurs based on the list of features that are contained in the new gesture.

More formally, a video V_i consists of a sequence of features $F_i = f_{i1}, f_{i2}, \ldots, f_{in}$. Each f_{ij} corresponds to a location of the hand in 3D space. If two videos, V_i and V_j consist of a particular feature, f_{ik} and f_{jk} respectively, we build a representative feature f'_k that can be thought of as a "average" feature. In order to build a set of all such f'_k's, we identify all possible features from the videos and group the images by similarity. Each f'_k is characterized by the mean and variance of its corresponding features from all videos. Each representative feature maintains a list of gestures it is a part of.

Note that the feature generalization step is unsupervised. Figure 5.1 shows some examples of features considered in our approach. Along with the position of the hands, we captured the approximate location of the head in order to situate the hands better in space.

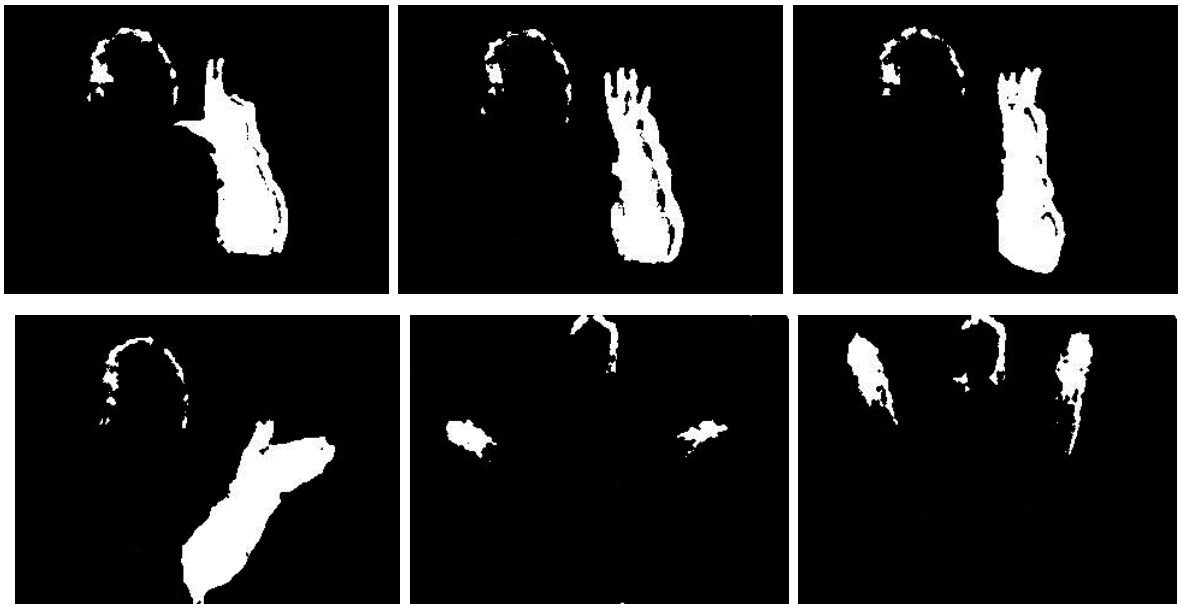


Figure 5.1: Bag of Features for Gesture Recognition

Once the representative features for each gesture have been identified, we built a model that can map each sequence of features to a gesture. In order to do this, we treat the features in the gestures like words in documents. We built a multilevel model, not very different from existing hierarchical models [4, 10, 61, 66, 76]. Along with classifying gestures, we also identify gestures that are similar to each other.

When a new gesture is presented, it is associated with the gesture group with the maximum similarity and the new gesture's features are also represented. The one-shot

learning in our approach is similar to the work by [66].

Evaluation and Observations

We evaluated the performance of the approach on the dataset provided during the initial development phase only. The gestures were either whole hand-based or finger-based. Also, we used predictive accuracy as our measure instead of that used by the challenge.

We obtained an accuracy of 66% on the validation set. Most of the errors can be attributed to the fact that temporal relationships were not explicitly represented. Further, we did not use the depth component of the dataset provided for this preliminary study. Including the depth component with the RGB component of the data posed a challenge since they act as multiple inputs to the system. Also, including multiple sources of information, even though they are of the same type, is hard.

The bag of features is not a new idea and has been applied to a variety of tasks in the past. Also, the results obtained are not as impressive as one would expect. However, the application of this approach to gesture recognition led us to some conclusions and observations.

The approach is based on robust features that account for minor variations. This resulted in the actual problem of learning gestures becoming easy task in terms of learning time. It requires only learning the associations between the feature sets and the classes. Much of the generalization depends on the power of constructing the bag of features rather than the ability of an algorithm that learns the gestures.

Further, as one might observe, the actual order of features was not accounted for in the model. This highlights the short coming in existing models for one-shot or deep learning. Existing models account for temporal relationships among features to a very limited extent. Further, learning such temporally dependent features is an important

future work.

Any approach that performs favorably must have the following characteristics:

1. Robustness to sparsity
2. Hierarchical or Multilevel, similarity based feature learning
3. Improved featured recognition leading to better prediction results using simpler prediction models
4. Feature learning step receives feedback about the relevance of learned features.

5.3.2 Extending to Hierarchical Classification

Hierarchical classification is a powerful approach to learn concepts. The approach is similar to hierarchical clustering and each cluster is assigned a class label derived from a taxonomy. Hierarchical topic models are classical examples of hierarchical classification, where each node in level l represents a concept, and its children in level $l + 1$ represent subtopics. Such a hierarchy carries more information about the concepts, and is very helpful in inference. One problem with such classifiers, of course, is finding the correct class labels for each node in the hierarchy. Obtaining labels from experts and crowd-sourcing are common approaches.

Recall the discussion about approaches to classification with rare examples from Section 5.1.1. Our hypothesis was that if we learn the rare example as a new class, or as a member of an existing class, hierarchical classifier are powerful. The capabilities of hierarchical models can be attributed to the following properties:

1. Adding X_{n+1} will not affect classifier accuracy: Since neither the existing class labels nor the class memberships are changed, classifying a new example will not

be affected by the assigning the rare example with a new class label.

2. We will know that both k and k' are related, and the information about their relationship is preserved in the hierarchy.

The approach descried earlier in Section 5.2 can be extended to accommodate a hierarchical structure. Hierarchical Dirichlet process [71] is an extension of the dirichlet process.

Chapter 6

Discussions and Conclusion

We have presented our work that explores multilevel models in the context of learning with small number of examples and with multiple features. Overall, we presented two approaches: one for regression (Chapter 4) and another for classification (Chapter 5). In both cases, we showed that the multilevel models learn rare examples better than traditional models.

Traditionally, rareness and class imbalance are treated as a function of frequency of occurrence of the examples alone. However, as we discussed in Chapter 2, this is not sufficient to capture all problem settings. In order to have a meaningful discussion about rare examples, it is important to use an improved definition. In this work, we defined rareness as a function of both the frequency of occurrence, and the relationship or difference between the examples in the given feature space. We used this degree of rareness in two ways: (1) to identify rare examples from a dataset such as those used our experiments, and (2) as a parameter when learning a classification model.

Our work focuses on labeled datasets over structured datasets, which contain either discreet or continuous valued feature set. For the purposes of this work, we did not

consider predicting time series or unstructured data. Further, the toxicity prediction and hand-gesture recognition datasets had a multimodal feature set; they used features of different types. We normalized them into a feature set that can be handled by our models.

Multilevel models are capable of handling high dimensional data. Successful learning in the presence of a small number of examples requires the learner to incorporate strong contextual information into the learning algorithm. Thus it is not only enough to learn with a large feature space, but our models must be able to learn the importance or weights of the features as well. If the features were not selected carefully, the model can learn incorrect concepts. In other cases, the choice of features can make the learning problem hard and decrease the predictive accuracy.

Also, in case of the classification tasks, we have demonstrated that the models discussed in this work can handle multi-class datasets.

Our work presented here finds applications in a variety of problem domains, and also leads to several next steps for further research. We discuss some of these in the following section.

6.1 Future Work

6.1.1 Extensions to other problems

1. *Recommendation Systems and Time series data* In our work, we have concentrated on classification and regression tasks in the traditional sense. The problem of learning with rare data is prevalent in recommendation systems and time series datasets as well. One direct extension of our work will be to develop models for such problems

as well.

2. *One-shot learning* is an extreme case of learning where an entire category of data is learned using only one example, utilizing the existing domain knowledge (See Section 2.2.2). There are a variety of issues that require attention with respect to one-shot learning.

   Existing approaches for one-shot learning may also require cluster reassignments depending on the nature of new data that needs to be learned. Existing approaches such as [66] first build a hierarchical model using existing data, which serves as background knowledge. Any update to the background knowledge can result in the rearrangement of the hierarchical model constructed. In many practical applications on an industrial scale, retraining is often less costly due to the large computational power available. However, it is important to study if online learning is feasible. More specifically, we must identify if cluster reassignments are less costly than retraining in terms of time and computational cost, without compromising the overall efficiency of the model.

6.1.2 Feature Selection and Representation Techniques

In its current state, we need a good feature selection process. We use the original set of attributes in the clustering step and the same feature set is used to train the mixture of heterogeneous models. This is neither efficient nor effective. Most datasets have large number of features, which is true for most tasks in artificial intelligence. It is important to note that the number of examples in each cluster is much lower than the number of attributes. The resulting data is sparse and the models tend to overfit. These factors can reduce the effectiveness of the models.

With respect to feature/ attribute selection, we address the following two problems.

1. **Improving feature selection, as a step towards improved performance**

   Features or attributes play a vital role in classification. It is important to study the role of feature selection when learning in the presence of the complexities.

   In this work, we plan to incorporate efficient approaches to select features such that the accuracy of classification is improved, especially when using hierarchical models.

   In hierarchical models, the feature selection problem is more complex. It is important to choose appropriate features at different levels to obtain the best results. Existing approaches to selecting features are more ad hoc and based on certain assumptions. One most common assumption is that among the categories in a hierarchical structure are usually of the type generalization-specialization [41, 76] . The lower level categories are supposed to have the same general properties as the higher level categories plus additional more specific properties.

   Xiao et.al [76] present the following example: "For classification between documents on sports and computer science, the frequency of the word *computer* is a very indicative feature. However, between the two subclasses compiler and operating system in computer science, the word *parsing* can be much more indicative than *computer*."

   While this is a valid assumption in most cases, identifying such features is anything but standard across applications.

   As mentioned earlier, SBMM, in its current state, does not include a good feature selection step. We believe that a better feature selection approach can give us

better guarantees on the performance. The result of the work will either confirm our hypothesis or show that the approach is independent on the feature selection process. Both these results are significant.

2. **Obtaining a set of features that can enable robust learning with multi-modal Data**

   One of the major problems is obtaining a set of features that can enable robust learning. The problem gets harder with increased complexity of the data. One common complexities is the presence of multiple genres or modes of features. Here are some examples.

   Computer Vision: When analyzing a video, a variety of features can be considered, depending on the task. *Temporal features* capture the variations across time and *spatial features* capture changes in the location of objects. Further, if we consider a frame, different features describe different parts of the image. Additionally, variations in size, rotation, and scaling add invariance in features in both image and video analysis.

   Chemical Toxicology: Chemicals have different properties and features, each falling under one or the other of various categories. Some features are based on the structure of an organic compound, some on the electrochemical properties such as conductivity, resistance etc., and some others based on reaction with other chemicals. In the dataset described in Chapter 4, there were at least four different sets of features. While most of these features are present as simple attribute-value pairs, the structure of the chemical compound is presented as a graph.

   Recently, various approaches have been proposed to tackle some of these problems. However, the solutions are less than impressive in terms of their ability to scale,

ability to handle multi-modal data and the overall effectiveness of the approaches.

In Chapters 4 and 5, we presented our work so far and some observations that will enable future research. Incorporating deep learning methods for unsupervised feature representation in a multilevel model can be one solution to the problem.

6.1.3 Other Theoretical and Practical Issues

Most multilevel models have been restricted to not more than 3 to 4 layers. Most models use a predetermined number of layers and their performances are reported for 3 to 4 layers. For most models, the extension to more levels is trivial. Fixing the number of layers before learning is not always desirable. The number of layers depends on the problem domain, the data and the task to be completed. For instance, for pattern recognition tasks with a multilayer neural model, the number of layers is relative easy to determine. Typically, the first layer corresponds to pixels and the subsequent layers correspond to line segments or edges, small regions or blobs, and so on.

This remains a practical concern and an open research question in the field. Bengio [11] lists more than 25 open problems, most of which are related to deep learning but can be extended to other parts of machine learning as well. Specifically, we are interested in knowing if it is possible to determine a good depth. The ultimate goal is to determine if there a depth that is mostly sufficient to attain human-level performance of AI tasks.

As we discussed in Section 2.6, presence of rare data can affect the performance of one-shot learning. This is not an easy problem to solve. Showing an experimental evidence of such negative effect is a feasible task.

REFERENCES

[1] Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 37–46, New York, NY, USA, 2001. ACM.

[2] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. *Applying Support Vector Machines to Imbalanced Datasets*, pages 39–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[3] D. Aldous. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII 1983*, pages 1–198. Springer, 1985.

[4] Andreas Argyriou, Andreas Maurer, and Massimiliano Pontil. An algorithm for transfer learning in a heterogeneous environment. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, ECML PKDD '08, pages 71–85, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] A. Asuncion and D.J. Newman. UCI machine learning repository, 2010.

[6] Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. *CoRR*, abs/0809.1493, 2008.

[7] Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*, pages 105–112. MIT Press, 2008.

[8] Dennis Bahler and Laura Navarro. Methods for combining heterogeneous sets of classifiers. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Workshop on New Research Problems for Machine Learning. AAAI Press/ The MIT Press, 2000.

[9] Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99, December 2003.

[10] Shai Ben-David and Reba Schuller. Exploiting task relatedness for mulitple task learning. In *Computational Learning Theory*, pages 567–580, 2003.

[11] Yoshua Bengio. *Learning Deep Architectures for AI*. Now Publishers Inc., Hanover, MA, USA, 2009.

[12] David Blackwell and James B. MacQueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

[13] D. Blei, A. Ng, and M. Jordan. Hierarchical Bayesian models for applications in information retrieval. In J. Bernardo, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*, volume 7, pages 25–44. Oxford University Press, 2003.

[14] David M. Blei and Michael I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Anal.*, 1(1):121–143, 03 2006.

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[16] Jake Bouvrie, Lorenzo Rosasco, and Tomaso Poggio. On invariance in hierarchical models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 162–170, 2009.

[17] Jake V. Bouvrie. *Hierarchical Learning: Theory with Applications in Speech and Vision*. PhD thesis, Massachusetts Institute of Technology, 2009.

[18] Cadaster. http://www.cadaster.eu/, 2009.

[19] Nitesh V. Chawla. Data mining for imbalanced datasets: An overview. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 875–886. Springer, 2010.

[20] William S. Cleveland and Susan J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):pp. 596–610, 1988.

[21] DARPA. Transfer learning proposer information pamphlet (pip). *Defense Advanced Research Projects Agency, Information Processing Technology Office*, 2005.

[22] Gerald DeJong. Generalizations based on explanations. In *IJCAI81, the Seventh International Joint Conference on Artificial Intelligence*, pages 67–69, 1981.

[23] Lee H Dicker and Dean P Foster. One-shot learning and big data with n=2. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 270–278. Curran Associates, Inc., 2013.

[24] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 127–136, New York, NY, USA, 2007. ACM.

[25] Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[26] L. Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[27] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, March 1973.

[28] P. Sargant Florence. *The Economic Journal*, 60(240):808–810, 1950.

[29] Andrew Gelman. Multilevel (Hierarchical) Modeling: What It Can and Can't Do. *Technometrics*, 2005.

[30] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2 edition, July 2003.

[31] Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press, 2006.

[32] Samuel J. Gershman and David M. Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1 – 12, 2012.

[33] Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *J. Comput. Sci. Technol.*, 25(4):653–664, July 2010.

[34] V. Hautamaki, I. Karkkainen, and P. Franti. Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 430–433 Vol.3, Aug 2004.

[35] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21:1263–1284, 2009.

[36] Neil Houlsby, Jose M. Hernandez-lobato, and Zoubin Ghahramani. Cold-start active learning with robust ordinal matrix factorization. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 766–774. JMLR Workshop and Conference Proceedings, 2014.

[37] Wenxin Jiang and Martin A. Tanner. Hierarchical mixtures-of-experts for exponential family regression models: Approximation and maximum likelihood estimation. *The Annals of Statistics*, 27(3):pp. 987–1011, 1999.

[38] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[39] S. Jung, A. Sen, and J.S Marron. Boundary behavior in high dimension, low sample size asymptotics of pca. In *Journal of Multivariate Analysis*, pages 190–203, 2012.

[40] Yunjae Jung, Haesun Park, Ding-Zhu Du, and Barry L. Drake. A decision criterion for the optimal number of clusters in hierarchical clustering. *J. of Global Optimization*, 25:91–111, January 2003.

[41] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In Editor Fisher, Douglas H, editor, *Proceedings of ICML97 14th International Conference on Machine Learning*, pages 170–178. Morgan Kaufmann, July 1997.

[42] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[43] Dragos D. Margineantu. When does imbalanced data require more than cost-sensitive learning? In *AAAI Technical Report*, volume WS-00-05, 2000.

[44] Kate McCarthy, Bibi Zabar, and Gary Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? In *Proceedings of the 1st international workshop on Utility-based data mining*, UBDM '05, pages 69–77, New York, NY, USA, 2005. ACM.

[45] Tom M. Mitchell. *Machine learning*. McGraw Hill series in computer science. McGraw-Hill, 1997.

[46] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(3):311–324, 2007.

[47] Bengt Muthén and Tihomir Asparouhov. Multilevel regression mixture analysis. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 172(3):639–657, 2009.

[48] Radford M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[49] Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 81–89. Springer, 2010.

[50] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November 2008.

[51] Dan Pelleg and Andrew Moore. Active learning for anomaly and rare-category detection. In *In Advances in Neural Information Processing Systems 18*, pages 1073–1080. MIT Press, 2004.

[52] Malay Ghosh Peter Hall, Yvonne Pittelkow. Theoretical measures of relative performance of classifiers for high dimensional data with small sample sizes. In *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, volume Vol. 70, No. 1, pages 159–173, 2008.

[53] J. Pitman. Combinatorial stochastic processes. *Ecole d'Eté de Probabilités de Saint-Flour XXXII*, 2002.

[54] Jim Pitman. Some developments of the blackwell-macqueen urn scheme. *Lecture Notes-Monograph Series*, 30:245–267, 1996.

[55] Foster Provost. Learning with imbalanced data sets 101. In *AAAI'2000 Workshop on Imbalanced Data Sets*, 2000.

[56] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[57] Piyush Rai and Hal Daumé III. Multitask learning via mixture of linear subspaces. In *NIPS Workshop on Transfer Learning by Learning Rich Generative Models*, Whistler, Canada, 2010.

[58] Bhavani Raskutti and Adam Kowalczyk. Extreme re-balancing for svms: A case study. *SIGKDD Explor. Newsl.*, 6(1):60–69, June 2004.

[59] C E Rasmussen, R M Neal, G E Hinton, D van Camp, M Revow, Z Ghahramani, R Kustra, and R Tibshirani. The delve manual version 1.1. In *http://www.cs.utoronto.ca/ delve*, 1996.

[60] Stephen W. Raudenbush and Anthony S. Bryk. *Hierarchical linear models: Applications and data analysis methods*. Sage, 2nd edition, 2002.

[61] Srinath Ravindran and Dennis Bahler. On imbalanced data, hierarchical models and transfer learning. In *Workshop on Challenges in Learning Hierarchical Models: Transfer Learning and Optimization*. Neural Information Processing Systems, 2011.

[62] Srinath Ravindran and Dennis Bahler. Multilevel regression models for learning in the presence of rare data. In *11th International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 107–112. IEEE, 2012.

[63] Marko Robnik-Sikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In Douglas H. Fisher, editor, *Fourteenth International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1997.

[64] Jonathan Root, Jing Qian, and Venkatesh Saligrama. Learning efficient anomaly detectors from K-NN graphs. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015.

[65] Michael T. Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G. Dietterich. To transfer or not to transfer. In *Workshop on Transfer Learning*. Neural Information Processing Systems, 2005.

[66] Ruslan Salakhutdinov, Josh Tenenbaum, and Antonio Torralba. One-Shot Learning with a Hierarchical Nonparametric Bayesian Model. In *MIT-CSAIL-TR-2010-052*, 2010.

[67] Kumar Sricharan and Alfred O. Hero, III. Efficient anomaly detection using bipartite k-nn graphs. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 478–486, USA, 2011. Curran Associates Inc.

[68] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1331–1338, Washington, DC, USA, 2005. IEEE Computer Society.

[69] Catherine A. Sugar and Gareth M. James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):pp. 750–763, 2003.

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

[71] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[72] Igor V Tetko, Iurii Sushko, Anil Kumar Pandey, Hao Zhu, Alexander Tropsha, Ester Papa, Tomas Oberg, Roberto Todeschini, Denis Fourches, and Alexandre Varnek. Critical assessment of qsar models of environmental toxicity against tetrahymena pyriformis: focusing on applicability domain and overfitting by variable selection. *J Chem Inf Model*, 48(9):1733–46, Sep 2008.

[73] Pavan Vatturi and Weng-Keen Wong. Category detection using hierarchical mean shift. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 847–856, New York, NY, USA, 2009. ACM.

[74] Gary M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl.*, 6:7–19, June 2004.

[75] Gang Wu and Edward Y. Chang. Class-boundary alignment for imbalanced dataset learning. In *In ICML 2003 Workshop on Learning from Imbalanced Data Sets*, pages 49–56, 2003.

[76] Lin Xiao, Dengyong Zhou, and Mingrui Wu. Hierarchical classification via orthogonal transfer. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 801–808, New York, NY, USA, June 2011. ACM.

[77] Ting Yu, Tony Jan, Simeon J. Simoff, and John K. Debenham. A hierarchical vqsvm for imbalanced data sets. In *IJCNN*, pages 518–523. IEEE, 2007.

[78] Ting Yu, Simeon Simoff, and Tony Jan. Vqsvm: A case study for incorporating prior domain knowledge into inductive machine learning. *Neurocomput.*, 73:2614–2623, August 2010.

[79] Hao Zhu, Alexander Tropsha, Denis Fourches, Alexandre Varnek, Ester Papa, Paola Gramatica, Tomas Oberg, Phuong Dao, Artem Cherkasov, and Igor V Tetko. Combinatorial qsar modeling of chemical toxicants tested against tetrahymena pyriformis. *J Chem Inf Model*, 48(4):766–84, Apr 2008.