# Rare Category Analysis

**Jingrui He**

May 2010
CMU-ML-10-106

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon

# Rare Category Analysis

Jingrui He

May 2010
CMU-ML-10-106

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

**Thesis Committee:**
Jaime Carbonell, CMU, Chair
John Lafferty, CMU
Larry Wasserman, CMU
Foster Provost, NYU

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This research was sponsored by an IBM Fellowship, SRI International subcontract 55-000691; IBM grant W0550432 and National Science Foundation grant ELA-9873009. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

**Keywords:** majority class, minority class, rare category, supervised, unsupervised, detection, characterization, feature selection

*To my parents, Fan He and Rongzhen Zhai.*

# Abstract

In many real world problems, rare categories (minority classes) play an essential role despite of their extreme scarcity. For example, in financial fraud detection, the vast majority of the financial transactions are legitimate, and only a small number may be fraudulent; in Medicare fraud detection, the percentage of bogus claims is small, but the total loss is significant; in network intrusion detection, malicious network activities are hidden among huge volumes of routine network traffic; in astronomy, only 0.001% of the objects in sky survey images are truly beyond the scope of current science and may lead to new discoveries; in spam image detection, the near-duplicate spam images are difficult to discover from the large number of non-spam image; in rare disease diagnosis, the rare diseases affect less than 1 out of 2000 people, but the consequences can be very severe. Therefore, the discovery, characterization and prediction of rare categories or rare examples may protect us from fraudulent or malicious behaviors, provide the aid for scientific discoveries, and even save lives.

This thesis focuses on rare category analysis, where the majority classes have a smooth distribution, and the minority classes exhibit a compactness property. Furthermore, we focus on the challenging cases where the support regions of the majority and minority classes overlap each other. To the best of our knowledge, this thesis is the first end-to-end investigation of rare categories.

Depending on the availability of the label information, we can perform either supervised or unsupervised rare category analysis. In the supervised settings, our first task is rare category detection, which is to discover at least one example from each minority class with the help of a labeling oracle. Then given labeled examples from all the classes, our second task is rare category characterization. The goal here is to find a compact representation for the minority classes in order to identify all the rare examples with high precision and recall. On the other hand, in the unsupervised settings, we do not have access to a labeling oracle. Here we propose to co-select candidate examples from the minority classes and the relevant features, which benefits both tasks (rare category selection and feature selection). For each of the above tasks, we have developed effective algorithms with theoretical guarantees as well as good empirical results.

In the future, we plan to apply rare category analysis on rich data, such as medical images, texts / blogs, Electronic Health Records (EHR), web link graphs, stream data, etc; we plan to build statistical models for the rare categories in order to understand how they emerge and evolve over time; we plan to study complex fraud based on rare category analysis; we plan to make use of transfer learning to help with our analysis; we also plan to build a complete system for rare category analysis.

## Acknowledgments

Looking back upon the many years I spent in school, I feel greatly indebted to the numerous people who have made me the person I am today.

I would like to thank Jaime Carbonell for being my advisor. I could not have hoped for a better advisor to guide me through my PhD studies. He is smart, professional, fun and knows everything. Every time I came to him with a new idea, he was always able to sharpen my thoughts and point out possible directions which turned out to be very fruitful. I would like to thank Christos Faloutsos for being both a wise mentor and a great friend. He shared with me a lot of his advices and experiences so that I would not take detours in my career. I would like to thank John Lafferty, Larry Wasserman, and Foster Provost for serving on my thesis committee. Their comments and suggestions really helped me improve my thesis work. And I would like to thank Avrim Blum for invaluable and insightful discussions.

I would also like to thank all my collaborators during my internship. These include: Hong-Jiang Zhang, Mingjing Li, and Lei Zhang from Microsoft Research Asia, who set up a very high standard for me at the beginning of my research life; Bo Thiesson from Microsoft Research, who constantly encouraged me to pursue one step further; Rick Lawrence and Yan Liu from IBM Research, the discussions with whom really expanded my horizon.

During my undergraduate studies in Tsinghua University, I was greatly inspired by the faculty in Automation Department, including Nanyuan Zhao, Changshui Zhang, Zongxia Liang, Yuanlie Lin, Yanda Li, Shi Yan, Mei Lu, to name a few. Their enthusiasm towards science and rigorous attitude towards research have a long impact on my own career. They deserve my deepest appreciation.

I am also grateful to Yanxi Liu. She interviewed me 5 years ago and her nice comments got me into CMU, the best place for studying computer science. Diane Stidle, who is always there for the students. Whenever I have a problem, her name is the first one I can think of to ask for help. Michelle Pagnani, who is always able to squeeze my meeting into Jaime's busy schedule. And all my friends, including Zhenzhen Kou, Fan Guo, Lei Li, Rong Yan, Xiaojing Fu, Jin Peng, Xiaomeng Chang, and many more.

Special gratitude goes to my grandparents, whose warm smiles always blessed me; my parents, who have the strongest belief in me all the time; Hanghang Tong, who has been nothing but a great husband; and my daughter Emma, who is the most naughty girl ever. Actually this thesis is a gift to her for her upcoming one year's birthday, though she could not understand a single word at this time.

# Contents

# Chapter 1

# Introduction

Imbalanced data sets are prevalent in real applications, i.e., some classes occupy the majority of the data set, a.k.a., the majority classes; whereas the remaining classes only have a few examples, a.k.a., the minority classes or the rare categories. For example, in financial fraud detection, the vast majority of the financial transactions are legitimate, and only a small number may be fraudulent [Bay *et al.*, 2006]; in Medicare fraud detection, the percentage of bogus claims is small, but the total loss is significant; in network intrusion detection, new malicious network activities are hidden among huge volumes of routine network traffic [Wu *et al.*, 2007][Vatturi & Wong, 2009]; in astronomy, only 0.001% of the objects in sky survey images are truly beyond the scope of current science and may lead to new discoveries [Pelleg & Moore, 2004]; in spam image detection, near-duplicate spam images are difficult to discover from the large number of non-spam images [Wang *et al.*, 2007]; in health care, the rare diseases affect less than 1 out of 2000 people, but the consequences are severe. Compared with the majority classes, the minority classes are often of much greater interest to the users. The main focus of my research is rare category analysis, which refers to the problem of analyzing the minority classes in an imbalanced data set. In this thesis, we plan to address this problem from different perspectives.

## 1.1 Motivation

When dealing with highly imbalanced data sets, there are a number of challenges. For example, in financial fraud detection, we may want to discover new types of fraud transactions from a large number of unexamined financial transactions with the help of a domain expert. In a more abstract way, given an imbalanced data set, our goal is to discover a few examples from the minority classes when we have access to a labeling oracle. Due to the extreme scarcity of the new types of fraud transactions compared with the normal transactions, simple methods such as random sampling would result in a huge number of label requests from the domain expert, which can be very expensive. Therefore, to reduce the labeling cost, we need more effective methods for discovering the new types of fraud transactions. Take rare disease diagnosis as another example. Given a small number of patients with a specific rare disease, how can we characterize this rare disease based on a subset of the medical measurements? With this characterization, we hope to help domain experts better understand the mechanism of this disease, distinguish it from other diseases for the best treatment, and identify potential patients with the same disease. The major challenge here is the insufficiency of label information, which might be alleviated by leveraging the information of health people as well as patients with similar diseases. Yet another example is in Medicare fraud detection. Due to the large number of medical claims submitted to the computer system for Medicare services, it is impossible for a domain expert to examine each of them and report suspicious ones. Therefore, an automated program that is able to detect complex fraud patterns with a high accuracy will greatly reduce the demand for human labor. The major challenge here is the lack of label information, which might be compensated by studying the properties of known fraud patterns. All the above challenges are associated with rare category analysis, where the main theme is to provide powerful tools for analyzing the rare categories in real applications.

## 1.2 Problem Definition

In rare category analysis, we are given an imbalanced data set, which is unlabeled initially. Depending on the availability of the label information, rare category analysis can be performed in the supervised or unsupervised fashion.

In supervised rare category analysis, we have access to a labeling oracle, which is able to provide us with the label information of any example with a fixed cost. In this case, rare category analysis can be divided into the following two tasks.

1. **Rare category detection**: in this task, we start from *de-novo*, and propose initial candidates of each minority class to the labeling oracle (one candidate in each round) in an active learning fashion, hoping to find at least one example from each minority class with the least total label requests. This task serves as the initial exploration step of the data set, and generates a set of labeled examples from each class, which can be used in the second task.
2. **Rare category characterization**: in this task, we have labeled examples from each class (both majority and minority classes), which are obtained from the first task, as well as a set of unlabeled examples as input. The goal is to find a compact representation for the minority classes in order to identify all the rare examples with high precision and recall. This task serves as the exploitation step of the data set, and constructs a reliable classifier, which can be used to identify future unseen examples from the minority classes.

In unsupervised rare category analysis, we do not have such a labeling oracle, and the goal here is to address the following two problems.

1. Rare category selection: selecting a set of examples which are likely to come from the same minority classes;
2. Feature selection: selecting the features that are relevant to the minority classes.

## 1.3 General Assumptions

When dealing with examples with feature representations, we make the following general assumptions throughout this thesis.

1. **Smoothness assumption:** the underlying distribution of each majority class is sufficiently smooth.
2. **Compactness (clustering) assumption:** the examples from the same minority class form a compact cluster in the feature space or feature subspace;

An example of the underlying distribution where these assumptions are satisfied is shown in Fig. 1.1. It shows the underlying distribution of a one-dimensional synthetic data set. The majority class has a Gaussian distribution with a large variance; whereas the minority classes correspond to the two lower variance peaks.

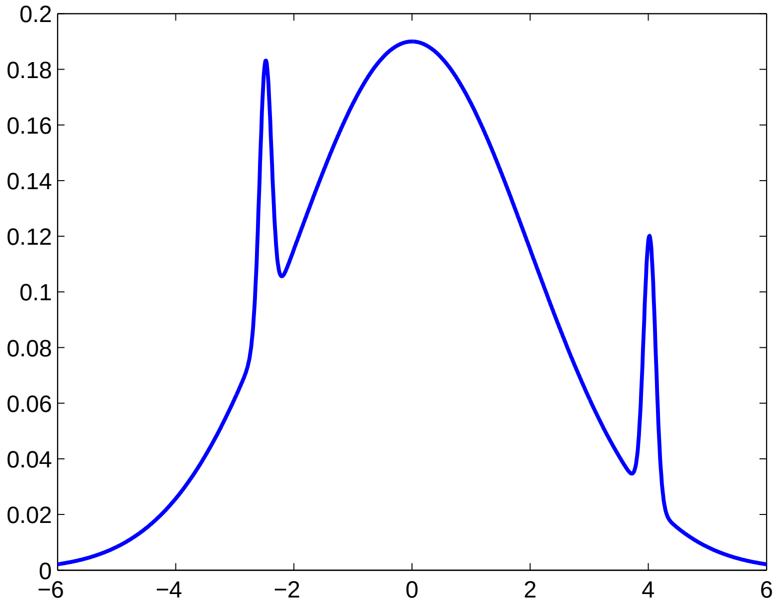


Figure 1.1: Underlying distribution of a one-dimensional synthetic data set: the majority class has a Gaussian distribution with a large variance; whereas the minority classes correspond to the two lower variance peaks.

The purpose of these assumptions is to make the minority classes identifiable. For example, if there are only one majority class and one minority class, and both have the same underlying distribution, for the task of rare category detection, no algorithms will perform better than random sampling. On the other hand, if the distribution of the majority class is very bumpy, and each bump is as narrow and sharp as the distribution of the minority class, then the minority classes can not be discovered very efficiently either. Empirical studies support these assumptions. Furthermore, according to the experimental results shown in the following chapters, our algorithms developed based on these assumptions often achieve good performance for rare category analysis.

Furthermore, in this thesis, we focus on the cases where the support regions of the majority and minority classes overlap with each other in the feature space (although some algorithms work well in both the separable and non-separable cases). The overlapping (non-separable) phenomenon can be observed in many real applications. For example, the guileful fraudulent people often try to camouflage their transactions within the normal transactions so that they can bypass the current fraud detection algorithms [Chau *et al.*, 2006]; in spam image detection, 'advanced' spam images are deliberately made like normal. Compared with the cases where the majority and minority classes are separable / near-separable from each other, which are targeted by most existing work in rare category analysis, such as [Fine & Mansour, 2006] and [Pelleg & Moore, 2004], the overlapping (non-separable) cases are more realistic and more challenging, but have not been well studied before.

## 1.4 Thesis Outline

### 1.4.1 Rare Category Detection

I plan to address the following questions with respect to rare category detection.

1. How to detect the rare categories in an unlabeled, imbalanced data set with the help of a labeling

oracle?

2. How to detect the rare categories for graph data, or relational data?
3. How to do rare category detection with the least prior information about the data set?

The first question is fundamental in rare category detection. Given an unlabeled, imbalanced data set, a naive way for finding examples from the minority classes is to randomly select examples from the data set to be labeled by the oracle until we have identified at least one example from each minority class. A major drawback of this random sampling strategy is the following. If a minority class is extremely rare, say its proportion in the data set is only $0.001\%$, in order to discover this minority class, the number of label requests by random sampling would be very large. Therefore, we need more effective methods to address the rare category detection challenge.

The second question is similar to the first one except that here we are interested in graph data (relational data) instead of data points with feature representations. Since graph data is very common in real applications, how to adapt the rare category detection algorithms for data points with feature representations to this data type is of key importance.

The third question aims at doing rare category detection with less prior information about the data set, and the ultimate goal is prior-free rare category detection, i.e., the algorithm is given no prior information about the data set, such as the number of classes, the proportions of different classes, etc. This question is more difficult than the first two, and yet quite important, since in real applications, given an unlabeled data set, it is sometimes difficult to estimate the number of classes in the data set a prior, not to mention the proportions of different classes.

Although both Rare category detection and traditional active learning proceed by actively selecting examples to be labeled by an oracle, they are different in the following two aspects. First, in rare category detection, initially we do not have any labeled examples; whereas in traditional active learning, initially we have labeled examples from all the classes as input. Second, in rare category detection, our goal is to discover at least one example from each minority class with the least label requests from the oracle; whereas in traditional active learning, the goal is to improve the performance of the current classifier with the least label requests from the oracle. Furthermore, rare category detection is a bottleneck in reducing the overall sampling complexity of active learning [Balcan *et al.*, 2006, Dasgupta, 2005]. That is, the sampling complexity of many active learning algorithms are dominated by the initial stage of finding at least one example from each class, especially the minority classes. Therefore, effective rare category detection algorithms could help reduce the sampling complexity of active learning by a large margin.

The major difference between rare category detection and outlier detection is the following. In rare category detection, the examples from the same minority class are often self-similar, potentially forming compact clusters in the feature space; and we assume that the support regions of the majority and minority classes are NOT separable from each other, which is more challenging than the separable cases. On the other hand, in anomaly detection (outlier detection), each anomaly (outlier) is a single data point; the anomalies (outliers) are typically scattered in the feature space; and the anomalies (outliers) are often separable from the normal points.

### 1.4.2 Rare Category Characterization

Rare category characterization follows rare category detection. Given labeled examples from all the classes as input, the goal is to identify all the rare examples from the known minority classes with high precision and recall. Here, our question is: how to characterize the minority classes with a compact representation?

Rare category characterization is of key importance in many applications, such as text retrieval where the number of documents relevant to a particular query is very small, and the goal is to retrieve all of them on the top of the ranked list returned to the user.

Rare category characterization bears similarity but also some differences with imbalanced classification. In both tasks, the data set is imbalanced, and we have labeled training examples from each class. However, in imbalanced classification, the goal is to construct a classifier that optimizes a discriminative criterion for both the majority and the minority classes, such as balanced accuracy, G-mean, etc [Chawla, 2009]; whereas in rare category characterization, we only focus on the minority classes, and aim to identify all (or nearly all) the rare examples from the unlabeled data set with high precision and recall. On the other hand, in rare category characterization, we are able to obtain a compact representation for the minority classes, which can be used to improve the interpretability of the learning results to domain experts who have no knowledge in machine learning and data mining; whereas in imbalanced classification, such representations are not provided.

### 1.4.3 Unsupervised Rare Category Analysis

In unsupervised rare category analysis, we do not have access to any labeling oracle. Under certain assumptions, we can perform both rare category selection, which is to select a set of examples that are likely to come from the minority classes, and feature selection, which is to select a set of feature relevant to the minority classes.

Rare category selection is similar to rare category detection in that both start *de-novo*, i.e., no label information is available at the beginning. However, in rare category detection, the labeling oracle gives the label information of the candidate example selected in each round, and the learning algorithm can adjust its model based on this information and propose new candidates, hoping to find at least one example from each minority class with only a few label requests; whereas in rare category selection, no label information is available during the training stage, and the goal is to select a set of examples which are likely to come from the minority classes.

In many imbalanced data sets, the examples from the same minority class are close to each other in some dimensions, and far apart in others. For example, in financial fraud detection, small-amount-probing type of fraud transactions are similar to each other in terms of the small amount being stolen in each transaction; however, different transactions may occur in different locations, at different time, etc. Therefore, to better describe the minority classes, we need to identify the relevant features.

Notice that existing co-clustering algorithms would fail in our settings due to the following two reasons. First, we are dealing with the cases where the majority and minority classes overlap in the feature space; whereas existing co-clustering algorithms mainly work in the separable / near-separable cases. Second, in our applications, we hope to find the features relevant to the minority classes; whereas existing co-clustering algorithms would simply ignore the rare examples due to their scarcity, and the selected features would be relevant to the majority classes only.

## 1.5 Main Contributions

To the best of our knowledge, this thesis is the first end-to-end investigation of rare categories in imbalanced data sets in both the supervised and unsupervised settings[1]. To be specific, our main contributions can be summarized below.

1. We distill different tasks in rare category analysis, namely rare category detection and rare category characterization in the supervised settings; rare category selection and feature selection in the unsupervised settings.

[1] Existing work on imbalanced data sets by researchers such as Gary Weiss and Nitesh Chawla mainly focuses on the supervised settings when labeled examples from all the classes are given as input.

2. For rare category detection, we develop different algorithms for data with feature representations and graph data, given full prior information, partial prior information, or even no prior information. For each of the proposed algorithms, we provide theoretical justification as well as empirical evaluations, showing the effectiveness of these algorithms.
3. For rare category characterization, we propose an optimization problem which captures the idea of the minimum-radius hyper-ball for the minority classes. Then we develop an effective algorithm to find its solution based on projected subgradient method. Experimental results show that its performance is better than some existing methods for imbalanced classification on benchmark data sets.
4. For unsupervised rare category analysis, we propose to co-select the rare examples and the relevant features of the minority classes, which benefits both tasks. To this end, we design an optimization framework, which is well justified theoretically, and an optimization algorithm based on augmented Lagrangian method. The performance of this algorithm is evaluated by experiments on some benchmark data sets.

## 1.6 General Notation

Given a set of $n$ unlabeled examples $S = \{x_1, \ldots, x_n\}$, $x_i \in \mathbb{R}^d$, which come from $m$ distinct classes, i.e., $y_i \in \{1, \ldots, m\}$. Without loss of generality, assume that $\sum_{i=1}^n x_i = \vec{0}$ and $\frac{1}{n}\sum_{i=1}^n (x_i^j)^2 = 1$, where $x_i^j$ is the $j^{\text{th}}$ feature of $x_i$. For the sake of simplicity, assume that there is one majority class with prior $p_1$, which corresponds to $y_i = 1$, and all the other classes are minority classes with priors $p_2, \ldots, p_m$, $p_1 \gg p_c$, $c \neq 1$. Notice that in real applications, we may have multiple majority classes. If all of them satisfy the smoothness assumption introduced in Section 1.3, they can be seen as a single class for the purpose of rare category analysis. This is because if a certain example is from one of the majority classes, we do not care which majority class it comes from.

Let $f_c$ denote the probability density function (pdf) of class $c$, where $c = 1, \ldots, m$. Based on our discussion in Section 1.3, $f_1$ for the majority class should satisfy the smoothness assumption; whereas $f_c$, $c = 2, \ldots, m$ for the minority classes should satisfy the compactness assumption.

Table 1.1 summarizes the general notation used in this thesis.

Table 1.1: Notation

| Symbol | Definition |
|---|---|
| $S$ | The set of unlabeled examples |
| $n$ | The number of examples in $S$ |
| $m$ | The number of classes in $S$ |
| $x_i$ | The $i^{\text{th}}$ unlabeled example |
| $x_i^j$ | The $j^{\text{th}}$ feature of $x_i$ |
| $d$ | The dimensionality of the feature space |
| $y_i$ | The class label of $x_i$ |
| $f_c$ | The probability density function of class $c$, $c = 1, \ldots, m$ |

Other more specific notation will be introduced where needed.

# Chapter 2

# Survey and Overview

In this chapter, we review related work in the following directions, including active learning, imbalanced classification, anomaly detection (outlier detection), rare category detection, unsupervised feature selection, clustering, and co-clustering.

## 2.1 Active Learning

The key idea behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns [Settles, 2010]. In active learning, we assume that the class labels are obtained from a labeling oracle with some cost, and under a fixed budget, we hope to maximally improve the performance of the learning algorithm. According to [Settles, 2010], there are three main settings in active learning: membership query synthesis, stream-based selective sampling, and pool-based sampling.

Many early active learning algorithms belong to membership query synthesis, such as [Angluin, 1987] [Angluin, 2001] [Cohn *et al.*, 1996]. One major problem with membership query synthesis is that the synthesized queries often have no practical meanings, and thus no appropriate labels. On the other hand, with stream-based active learning and pool-based sampling, the queries always correspond to real examples. Therefore, their label information can be readily provided by the oracle.

In stream-based selective sampling, given an unlabeled example, the learner must decide whether to query its class label or to discard it. For example, in [Cohn *et al.*, 1992], Cohn et al compute region of uncertainty, and query examples within in; in [Dagan & Engelson, 1995], Dagan et al proposed committee-based sampling, which evaluates the informativeness of an example by measuring the degree of disagreement between several model variants and only queries the more informative ones.

On the other hand, in pool-based sampling, queries are selected from a pool of unlabeled examples. Its major difference from stream-based selective sampling is the large amount of unlabeled data available at query time, which reveals additional information about the underlying distribution. For example, Tong et al [Tong *et al.*, 2001] proposed an active learning algorithm that minimizes the size of the version space; Mccallum [Mccallum, 1998] modified the Query-by-Committee method of active learning to use the unlabeled data for density estimation, and combined with EM to find the class labels of the unlabeled examples.

It should be mentioned that in traditional active learning, initially we have labeled examples from all the classes in order to build the very first classifier, which can be improved by actively selecting the training data. On the other hand, in rare category detection, initially we do not have any labeled examples, and the goal is to discover at least one example from each minority class with the least label requests. Combining rare category detection and traditional active learning, it has been noticed in [Balcan *et al.*, 2006] and [Dasgupta, 2005] that if the learning algorithm starts *de-novo*, finding the initial labeled examples from each class (i.e., rare category detection) becomes the bottleneck for reducing the sampling complexity. Furthermore, in supervised rare category analysis, following rare category detection, the second task is rare category characterization, which works in a semi-supervised fashion. In this task, in order to get a more accurate representation of the minority classes, we can make use of active learning to select the most informative examples to be added to the labeled set.

## 2.2 Imbalanced Classification

In imbalanced classification, the goal is to construct an accurate classifier that optimizes a discriminative criterion, such as balanced accuracy, G-mean, etc [Chawla, 2009]. Existing methods can be roughly categorized into 3 groups [Chawla, 2009], i.e., sampling-based methods [Kubat & Matwin, 1997][Chawla *et al.*, 2002], adapting learning algorithms by modifying objective functions or changing decision thresholds [Wu & Chang, 2003] [Huang *et al.*, 2004], and ensemble based methods [Sun *et al.*, 2006][Chawla *et al.*, 2003]. To be specific, in sampling-based methods, some methods under-sample the majority classes. For example, the one-sided sampling strategy proposed in [Kubat & Matwin, 1997] employs Tomek links [Tomek, 1976] followed by closest nearest neighbor [HART, 1968] to discard the majority class examples that lie in the borderline region, are noisy or redundant. In contrast, some sampling-based methods over-sample the hard examples. For example, the DataBoost-IM method proposed in [Guo & Viktor, 2004] generates synthetic examples according to the hard examples identified during the boosting algorithm; the SMOTEBoost algorithm proposed in [Chawla *et al.*, 2003] applies the SMOTE algorithm [Chawla *et al.*, 2002] to create new examples from the minority class in each boosting round. Furthermore, some methods combine over-sampling the minority class and under-sampling the majority class. For example, the SMOTE algorithm combined with under-sampling [Chawla *et al.*, 2002] was proven to outperform only under-sampling the majority class and varying the loss ratios; in [Tang *et al.*, 2009], different rebalance heuristics were incorporated into SVM modeling to tackle the problem of class imbalance, including over-sampling, under-sampling, etc.

Imbalanced classification and rare category characterization bear similarity but also some difference. On one hand, both tasks need labeled examples from all the classes as input. On the other hand, imbalanced classification and rare category characterization have different goals as well as different output. To be specific, in rare category characterization, we only focus on the minority classes, and aim to identify all (or nearly all) the rare examples from the unlabeled data set with high precision and recall; whereas in imbalanced classification, the goal is to construct a classifier that optimizes a discriminative criterion for both the majority and minority classes. Furthermore, in rare category characterization, we are able to obtain a compact representation for the minority classes, which can be provided to domain experts for better understanding of the learning results; whereas in imbalanced classification, such representations are not provided.

## 2.3 Anomaly Detection (Outlier Detection)

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [Chandola *et al.*, 2009]. Anomalies are often referred to as outliers. According to [Chandola *et al.*, 2009], the majority of anomaly detection techniques can be categorized into classification based, nearest neighbor based, clustering based, information theoretic, spectral, and statistical techniques. For example, in [Barbará *et al.*, 2001], the authors propose a method based on a technique called pseudo-Bayes estimators to enhance an anomaly detection systems's ability to detect new attacks while reducing the false alarm rate as much as possible. In [Ramaswamy *et al.*, 2000], the authors propose a novel formulation for distance-based outliers that is based on the distance of a point from its $k^{\text{th}}$ nearest neighbor. Then they rank each point on the basis of its distance to its $k^{\text{th}}$ nearest neighbor and declare the top $n$ points in this ranking to be outliers. In [Yu *et al.*, 2002], the authors propose the *FindOut* algorithm, which is an extension of the *WaveCluster* algorithm [Sheikholeslami *et al.*, 1998] in which the detected clusters are removed from the data and the residual instances are declared as nomalies. In [He *et al.*, 2005b], the authors formally define the problem of outlier detection in categorical data as an optimization problem from a global viewpoint based on entropy minimization, and present a local-search heuristic based algorithm for efficiently finding feasible solutions. In [Dutta *et al.*, 2007], the authors describe distributed algorithms for doing Principal Component Analysis (PCA) using random projection and sampling based techniques. Using the approximate principal components, they develop a distributed outlier detection algorithm based on the fact that the last principal component enables identification of data points which deviate sharply from the 'correlation structure' of the data. And in [Aggarwal & Yu, 2001], the authors discuss a new technique for outlier detection which is especially suited to very high dimensional data sets. The method works by finding lower dimensional projections which are locally sparse, and cannot be discovered easily by brute force techniques because of the number of combinations of possibilities.

In general, anomaly detection finds *individual* and *isolated* examples that differ from a given class in an unsupervised fashion. Typically, there is no way to characterize the anomalies since they are often different from each other. There exist a few works dealing with the case where the anomalies are clustered [Papadimitriou *et al.*, 2003]. However, they still assume that the anomalies are separable from the normal data points. On the other hand, in rare category detection, each rare category consists of a group of points, which form a compact cluster in the feature space and are self-similar. Furthermore, we are dealing with the challenging cases where the support regions of the majority and minority classes overlap with each other.

## 2.4 Rare Category Detection

Here, the goal is to find at least one example from each minority class with the help of a labeling oracle, minimizing the number of label requests. Up till now, researchers have developed several methods for rare category detection. For example, in [Pelleg & Moore, 2004], the authors assumed a mixture model to fit the data, and experimented with different hint selection methods, of which Interleaving performs the best; in [Fine & Mansour, 2006], the authors studied functions with multiple output values, and used active sampling to identify an example for each of the possible output values; in [Dasgupta & Hsu, 2008], the authors presented an active learning scheme that exploits cluster structure in the data, which was proven to be effective in rare category detection; and in [Vatturi & Wong, 2009], the authors proposed a new approach to rare category detection based on hierarchical mean shift, where a hierarchy is created by repeatedly applying mean shift with an increasing bandwidth on the data. Different from most existing work on rare category detection, which assume that the majority and minority classes are separable / near-separable from each other in the feature space, in Chapter 3 of this thesis, we target the more challenging cases where the support regions of different classes are not separable. Furthermore, besides empirical evaluations of the proposed algorithms, we also proved their effectiveness theoretically; whereas most existing algorithms do not have such guarantees.

## 2.5 Unsupervised Feature Selection

Generally speaking, existing feature selection methods in the unsupervised settings can be categorized as wrapper models and filter models. The wrapper models evaluate feature subsets based on the clustering results, such as the FSSEM algorithm [Dy & Brodley, 2000], the mixture-based approach which extends to the unsupervised context the mutual-information based criterion [Law *et al.*, 2002], and the ELSA algorithm [Kim *et al.*, 2000]. The filter models are independent of the clustering algorithm, such as the feature selection algorithm based on maximum information compression index [Mitra *et al.*, 2002], the feature selection method using distance-based entropy [Dash *et al.*, 2002], and the feature selection method based on Laplacian score [He *et al.*, 2005a].

In unsupervised rare category analysis, one of the problems we want to address is feature selection, i.e., selecting a set of features relevant to the minority classes. In our settings, since the class proportions are very skewed, the general-purpose wrapper and filter methods would fail by selecting the features primarily relevant to the majority classes. Therefore, we need new feature selection methods that are tailored for rare category analysis.

## 2.6 Clustering

According to [Glenn & Fung, 2001], clustering refers to the grouping together of similar data items into clusters. Existing clustering algorithms can be categorized into the following 2 main classes [Glenn & Fung, 2001]: parametric clustering and non-parametric clustering. In general, parametric methods attempt to minimize a cost function or an optimality criterion which associates a cost to each example-cluster assignment. It can be further classified into 2 groups: generative models and reconstructive models. In generative models, the basic idea is that the input examples are observations from a set of unknown distributions. For example, in Gaussian mixture models [Reynolds & Rose, 1995], the data are viewed as coming from a mixture of probability Gaussian distribution, each representing a different cluster; in C-means fuzzy clustering [Dunn, 1973], the membership of a point is shared among various clusters. On the other hand, reconstructive methods generally attempt to minimize a cost function. For example, K-means clustering forms clusters in numeric domains, partitioning examples into disjoint clusters [Duda *et al.*, 2000]; in Deterministic Annealing EM Algorithm (DAEM) [Hofmann & Buhmann, 1997], the maximization of the likelihood function is embedded in the minimization of the thermodynamic free energy, depending on the temperature which controls the annealing process. For nonparametric methods, two good representative examples are the agglomerative and divisive algorithms, also called hierarchical algorithms [Johnson, 1967], that produce dendrogram.

In unsupervised rare category analysis, another important problem we want to address is rare category selection, i.e., selecting a set of examples which are likely to come from the minority classes. General-purpose clustering algorithms do not fit here because the proportions of different classes are very skewed and the support regions of the majority and minority classes overlap with each other. In this case, general-purpose clustering algorithms tend to overlook the minority classes and generate clusters within the majority classes. Therefore, we need to develop new methods for rare category selection which leverage the property of the minority classes.

## 2.7 Co-clustering

The idea of using compression for clustering can be traced back to the information-theoretic co-clustering algorithm [Dhillon *et al.*, 2003], where the normalized non-negative contingency table is treated as a joint probability distribution between two discrete random variables that take values over the rows and columns. Then co-clustering is defined as a pair of mappings from rows to row clusters and from columns to column clusters. According to information theory, the optimal co-clustering is the one that minimizes the difference in mutual information between the original random variables and the mutual information between the clustered random variables. The algorithm for minimizing the above criterion intertwines both row and column clustering at all stages. Row clustering is done by assessing closeness of each row distribution, in relative entropy, to certain 'row cluster prototypes'. Column clustering is done similarly, and this process is iterated until it converges to a local minimum. It can be theoretically proved that the proposed algorithm never increases the criterion, and gradually improves the quality of co-clustering.

Although the information-theoretic co-clustering algorithm can only be applied to bipartite graphs, the idea behind this algorithm can be generalized to more than two types of heterogeneous objects. For example, in [Gao *et al.*, 2007], the authors proposed the CBGC algorithm. It aims to do collective clustering for star-shaped inter-relationships among different types of objects. First, it transforms the star-shaped structure into a set of bipartite graphs; then it formulates a constrained optimization problem, where the objective function is a weighted sum of the Rayleigh quotients on different bipartite graphs, and the constraints are that clustering results for the same type of objects should be the same. Follow-up work includes the high order co-clustering [Greco *et al.*, 2007]. Another example is the spectral relational clustering algorithm proposed in [Long *et al.*, 2006]. Unlike the previous algorithm, this algorithm is not restricted to star-shaped structures. It is based on a general model, the collective factorization on related matrices. This model clusters multi-type interrelated objects simultaneously based on both the relation and the feature information. It exploits the interactions between the hidden structures of different types of objects through the related factorizations which share matrix factors, i.e., cluster indicator matrices. The resulting spectral relational clustering algorithm iteratively updates the cluster indicator matrices using the leading eigenvectors of a specially designed matrix until convergence. More recently, the collective matrix factorization proposed by Singh *et al.* [Singh & Gordon, 2008a] [Singh & Gordon, 2008b] can also be used for clustering k-partite graphs.

Other related work includes (1) GraphScope [Sun *et al.*, 2007], which uses a similar information-theoretic criterion as cross association for time-evolving graphs to segment time into homogeneous intervals; and (2) multi-way distributional clustering (MDC) [Bekkerman *et al.*, 2005] which is demonstrated to outperform the previous information-theoretic clustering algorithms by the time the algorithm was proposed.

At the first glance, one may apply co-clustering algorithms to simultaneously address the problem of rare category selection and feature selection. The problem here is similar to the one mentioned in Section 2.6. That is, due to the extreme skewness of the class proportions and the overlapping support regions, general-purpose co-clustering algorithms may not be able to correctly identify the few rare examples or the features relevant to the rare categories; whereas our proposed algorithm for co-selecting the rare examples and the relevant features addresses this problem by making use of the clustering property of the minority classes.

# Chapter 3

# Rare Category Detection

In this chapter, we focus on rare category detection, the first task in the supervised settings. In this task, we are given an unlabeled, imbalanced data set, which is often non-separable, and have access to a labeling oracle, which is able to give us the class label of any example with a fixed cost. The goal here is to discover an least one example from each minority class with the least label requests.

The main contributions of this chapter can be summarized as follows.

**Algorithms with Theoretical Guarantees.** To the best of our knowledge, we propose the first rare category detection algorithms with theoretical guarantees;

**Algorithms for Different Data Types.** For data with feature representations and graph data (relational data), we propose different algorithms that exploit their specific properties. These algorithms work with different amount of prior information.

The rest of this chapter is organized as follows. In Section 3.1, we introduce the detection algorithms with prior information for data with feature representation. The prior-free algorithm is introduced in Section 3.2. Then in Section 3.3, we present our detection algorithms for graph data, or relational data, given full prior information or partial prior information. Finally, in Section 3.4, we give a brief summary of rare category detection.

## 3.1 Rare Category Detection with Priors for Data with Features

In this section, we propose prior-dependent algorithms for rare category detection in the context of active learning, which are designed for data with feature representations. We typically start *de-novo*, no category labels, though our algorithms make no such assumption. Different from existing methods, we aim to solve the difficult cases, i.e., we do not assume separability or near-separability of the classes. Intuitively, our algorithms make use of nearest neighbors to measure local density around each example. In each iteration, the algorithms select an example with the maximum change in local density on a certain scale, and ask the oracle for its label. The algorithms stop once they have found at least one example from each class (given the knowledge of the number of classes). When the two assumptions in Section 1.3 are satisfied, the proposed algorithms will select examples both on the boundary and in the interior of the minority classes, and are proven to be effective theoretically.

The rest of the section is organized as follows. In Subsection 3.1.1, we introduce our algorithm for the binary cases and provide theoretical justification. In Subsection 3.1.2, we discuss about the more general cases where there are more than one minority classes in the data set. Finally, Subsection 3.1.3 provides some experimental results.

### 3.1.1 Rare Category Detection for the Binary Cases

#### Algorithm

First let us focus on the simplest case where $m = 2$. Therefore, $p_1 = 1 - p_2$, and $p_2 \ll 1$. Here, we assume that we have an estimate of the value of $p_2$ a priori. Next, we introduce our algorithm for rare category detection based on nearest neighbors, which is presented in Alg. 1. The basic idea is to find maximum changes in local density, which might indicate the location of a rare category.

The algorithm works as follows. Given the unlabeled set $S$ and the prior of the minority class $p_2$, we first estimate the number $K$ of minority class examples in $S$. Then, for each example, we record its distance from the $K^{\text{th}}$ nearest neighbor, which could be realized by kd-trees [Moore, 1991]. The minimum distance over all the examples is assigned to $r'$. Next, we draw a hyper-ball centered at each example with radius $r'$, and count the number of examples enclosed by this hyper-ball, which is denoted as $n_i$. $n_i$ is roughly in proportion to the local density. To measure the change of local density around a certain point $x_i$, in each iteration of Step 3, we subtract $n_k$ of neighboring points from $n_i$, and let the maximum value be the score of $x_i$. The example with the maximum score is selected for labeling by the oracle. If the example is from the minority class, stop the iteration; otherwise, enlarge the neighborhood where the scores of the examples are re-calculated and continue.

Before giving the theoretical justification, here, we give an intuitive explanation of why the algorithm works. Assume that the minority class is concentrated in a small region and the probability density function (pdf) of the majority class is locally smooth. Firstly, since the support region of the minority class is very small, it is important to find its scale. The $r'$ value obtained in Step 1 will be used to calculate the local density $n_i$. Since $r'$ is based on the minimum $K^{\text{th}}$ nearest neighbor distance, it is never too large to smooth out changes of local density, and thus it is a good measure of the scale. Secondly, the score of a certain point, corresponding to the change in local density, is the maximum of the difference in local density between this point and all of its neighboring points. In this way, we are not only able to select points on the boundary of the minority class, but also points in the interior, given that the region is small. Finally, by gradually enlarging the neighborhood where the scores are calculated, we can further explore the interior of the support region, and increase our chance of finding a minority class example.

**Algorithm 1** Nearest-Neighbor-Based Rare Category Detection for the Binary Case (*NNDB*)

```
Input: $S, p_2$
1: Let $K = np_2$. For each example, calculate the distance to its $K^{\text{th}}$ nearest neighbor. Set $r'$ to be the minimum value among all the examples.
2: $\forall x_i \in S$, let $NN(x_i, r') = \{x | x \in S, \|x - x_i\| \le r'\}$, and $n_i = |NN(x_i, r')|$.
3: for $t = 1 : n$ do
4:     $\forall x_i \in S$, if $x_i$ has not been selected, then $s_i = \max_{x_k \in NN(x_i, tr')} (n_i - n_k)$; otherwise, $s_i = -\infty$.
5:     Query $x = \arg\max_{x_i \in S} s_i$.
6:     If the label of $x$ is 2, break.
7: end for
```

#### Justification

Next we prove that if the minority class is concentrated in a small region and the pdf of the majority class is locally smooth, the proposed algorithm will repeatedly sample in the region where the rare examples occur with a high probability.

First of all, we make the following specific assumptions.

**Assumptions**

1. $f_2(x)$ is uniform within a hyper-ball $B$ of radius $r$ centered at $b$, i.e., $f_2(x) = \frac{1}{V(r)}$, if $x \in B$; and 0 otherwise, where $V(r) \propto r^d$ is the volume of $B$.
2. $f_1(x)$ is bounded and positive in $B$[1], i.e., $f_1(x) \geq \frac{\kappa_1 p_2}{(1-p_2)V(r)}$, $\forall x \in B$ and $f_1(x) \leq \frac{\kappa_2 p_2}{(1-p_2)V(r)}$, $\forall x \in \mathbb{R}^d$, where $\kappa_1, \kappa_2 > 0$ are two constants.

With the above assumptions, we have the following lemma and theorem. Note that variants of the following proof apply if we assume a different minority class distribution, such as a tight Gaussian.

**Lemma 1.** *$\forall \epsilon, \delta > 0$, if $n \geq \max\{\frac{1}{2\kappa_1^2 p_2^2}\log\frac{3}{\delta}, \frac{1}{2(1-2^{-d})^2 p_2^2}\log\frac{3}{\delta}, \frac{1}{\epsilon^4 V(\frac{r_2}{2})^4}\log\frac{3}{\delta}\}$, where $r_2 = \frac{r}{(1+\kappa_2)^{\frac{1}{d}}}$, and $V(\frac{r_2}{2})$ is the volume of a hyper-ball with radius $\frac{r_2}{2}$, then with probability at least $1-\delta$, $\frac{r_2}{2} \leq r' \leq r$ and $|\frac{n_i}{n} - E(\frac{n_i}{n})| \leq \epsilon V(r')$, $1 \leq i \leq n$, where $V(r')$ is the volume of a hyper-ball with radius $r'$.*

*Proof.* First, notice that the expected proportion of points falling inside $B$, $E(\frac{|NN(b,r)|}{n}) \geq (\kappa_1 + 1)p_2$, and that the maximum expected proportion of points falling inside any hyper-ball of radius $\frac{r_2}{2}$, $\max\limits_{x \in \mathbb{R}^d}[E(\frac{|NN(x,\frac{r_2}{2})|}{n})] \leq 2^{-d}p_2$. Then

$$\begin{aligned}
&\Pr[r' > r \text{ or } r' < \frac{r_2}{2} \text{ or } \exists x_i \in S \text{ s.t., } |\frac{n_i}{n} - E(\frac{n_i}{n})| > \epsilon V(r')] \\
&\leq \Pr[r' > r] + \Pr[r' < \frac{r_2}{2}] + \Pr[r' \geq \frac{r_2}{2} \text{ and } \exists x_i \in S \text{ s.t., } |\frac{n_i}{n} - E(\frac{n_i}{n})| > \epsilon V(r')] \\
&\leq \Pr[|NN(b,r)| < K] + \Pr[\max_{x \in \mathbb{R}^d}|NN(x,\frac{r_2}{2})| > K] + n\Pr[|\frac{n_i}{n} - E(\frac{n_i}{n})| > \epsilon V(r')|r' \geq \frac{r_2}{2}] \\
&= \Pr[|\frac{NN(b,r)}{n}| < p_2] + \Pr[\max_{x \in \mathbb{R}^d}|\frac{NN(x,\frac{r_2}{2})}{n}| > p_2] + n\Pr[|\frac{n_i}{n} - E(\frac{n_i}{n})| > \epsilon V(r')|r' \geq \frac{r_2}{2}] \\
&\leq e^{-2n\kappa_1^2 p_2^2} + e^{-2n(1-2^{-d})^2 p_2^2} + 2ne^{-2n\epsilon^2 V(r')^2}
\end{aligned}$$

where the last inequality is based on Hoeffding bound.

Let $e^{-2n\kappa_1^2 p_2^2} \leq \frac{\delta}{3}$, $e^{-2n(1-2^{-d})^2 p_2^2} \leq \frac{\delta}{3}$ and $2ne^{-2n\epsilon^2 V(r')} \leq 2ne^{-2n\epsilon^2 V(\frac{r_2}{2})^2} \leq \frac{\delta}{3}$, we obtain $n \geq \frac{1}{2\kappa_1^2 p_2^2}\log\frac{3}{\delta}$, $n \geq \frac{1}{2(1-2^{-d})^2 p_2^2}\log\frac{3}{\delta}$, and $n \geq \frac{1}{\epsilon^4 V(\frac{r_2}{2})^4}\log\frac{3}{\delta}$. □

Based on Lemma 1, we get the following theorem, which shows the effectiveness of the proposed algorithm.

**Theorem 1.** *If*

1. *Let $B^2$ be the hyper-ball centered at b with radius $2r$. The minimum distance between the points inside $B$ and the ones outside $B^2$ is not too large, i.e., $\min\{\|x_i - x_k\| | x_i, x_k \in S, \|x_i - b\| \leq r, \|x_k - b\| > 2r\} \leq \alpha$, where $\alpha$ is a positive parameter.*
2. *$f_1(x)$ is locally smooth, i.e., $\forall x, y \in \mathbb{R}^d, |f_1(x) - f_1(y)| \leq \frac{\beta\|x-y\|}{\alpha}$, where $\beta \leq \frac{p_2^2 OV(\frac{r_2}{2}, r)}{2^{d+1}V(r)^2}$ and $OV(\frac{r_2}{2}, r)$ is the volume of the overlapping region of two hyper-balls: one is of radius $r$, the other one is of radius $\frac{r_2}{2}$, and its center is on the sphere of the bigger one.*
3. *The number of examples is sufficiently large,<br>i.e., $n \geq \max\{\frac{1}{2\kappa_1^2 p_2^2}\log\frac{3}{\delta}, \frac{1}{2(1-2^{-d})^2 p_2^2}\log\frac{3}{\delta}, \frac{1}{(1-p_2)^4\beta^4 V(\frac{r_2}{2})^4}\log\frac{3}{\delta}\}$.*

*then with probability at least $1-\delta$, after $\lceil\frac{2\alpha}{r_2}\rceil$ iterations,* NNDB *will query at least one example whose probability of coming from the minority class is at least $\frac{1}{3}$, and it will continue querying such examples until the $\lfloor(\frac{2^d}{p_2(1-p_2)} - 2)\cdot\frac{\alpha}{r}\rfloor^{\text{th}}$ iteration.*

[1] Notice that here we are only dealing with the hard case where $f_1(x)$ is positive within $B$. In the separable case where the support regions of the two classes do not overlap, we can use other methods to detect the minority class, such as the one proposed in [Pelleg & Moore, 2004].

*Proof.* Based on Lemma 1, using condition 3, if the number of examples is sufficiently large, then with probability at least $1-\delta$, $\frac{r_2}{2} \leq r' \leq r$ and $|\frac{n_i}{n} - E(\frac{n_i}{n})| \leq (1-p_2)\beta V(r')$, $1 \leq i \leq n$. According to condition 2, $\forall x_i, x_k \in S$ s.t., $\|x_i - b\| > 2r$, $\|x_k - b\| > 2r$ and $\|x_i - x_k\| \leq \alpha$, $E(\frac{n_i}{n})$ and $E(\frac{n_k}{n})$ will not be affected by the minority class, and $|E(\frac{n_i}{n}) - E(\frac{n_k}{n})| \leq (1-p_2)\beta V(r') \leq (1-p_2)\beta V(r)$. Note that $\alpha$ is always bigger than $r$. Based on the above inequalities, we have

$$|\frac{n_i}{n} - \frac{n_k}{n}| \leq |\frac{n_i}{n} - E(\frac{n_i}{n})| + |\frac{n_k}{n} - E(\frac{n_k}{n})| + |E(\frac{n_i}{n}) - E(\frac{n_k}{n})| \leq 3(1-p_2)\beta V(r) \tag{3.1}$$

From Inequality 3.1, it is not hard to see that $\forall x_i, x_k \in S$, s.t., $\|x_i - b\| > 2r$ and $\|x_i - x_k\| \leq \alpha$, $\frac{n_i}{n} - \frac{n_k}{n} \leq 3(1-p_2)\beta V(r)$, i.e., when $tr' = \alpha$,

$$\frac{s_i}{n} \leq 3(1-p_2)\beta V(r) \tag{3.2}$$

This is because if $\|x_k - b\| \leq 2r$, the minority class may also contribute to $\frac{n_k}{n}$, and thus the score may be even smaller.

On the other hand, based on condition 1, there exist two points $x_k, x_l \in S$, s.t., $\|x_k - b\| \leq r$, $\|x_l - b\| > 2r$, and $\|x_k - x_l\| \leq \alpha$. Since the contribution of the minority class to $E(\frac{n_k}{n})$ is at least $\frac{p_2 \cdot OV(\frac{r_2}{2}, r)}{V(r)}$, so $E(\frac{n_k}{n}) - E(\frac{n_l}{n}) \geq \frac{p_2 \cdot OV(\frac{r_2}{2}, r)}{V(r)} - (1-p_2)\beta V(r') \geq \frac{p_2 \cdot OV(\frac{r_2}{2}, r)}{V(r)} - (1-p_2)\beta V(r)$. Since for any example $x_i \in S$, we have $|\frac{n_i}{n} - E(\frac{n_i}{n})| \leq (1-p_2)\beta V(r') \leq (1-p_2)\beta V(r)$, therefore

$$\frac{n_k}{n} - \frac{n_l}{n} \geq \frac{p_2 \cdot OV(\frac{r_2}{2}, r)}{V(r)} - 3(1-p_2)\beta V(r) \geq \frac{p_2 \cdot OV(\frac{r_2}{2}, r)}{V(r)} - \frac{3(1-p_2)p_2^2 \cdot OV(\frac{r_2}{2}, r)}{2^{d+1} V(r)}$$

Since $p_2$ is very small, $p_2 \gg \frac{3(1-p_2)p_2^2}{2^{d+1}}$; therefore, $\frac{n_k}{n} - \frac{n_l}{n} > 3(1-p_2)\beta V(r)$, i.e., when $tr' = \alpha$,

$$\frac{s_k}{n} > 3(1-p_2)\beta V(r) \tag{3.3}$$

In Step 4 of the proposed algorithm, we gradually enlarge the neighborhood to calculate the change of local density. When $tr' = \alpha$, based on inequalities (2) and (3), $\forall x_i \in S$, $\|x_i - b\| > 2r$, we have $s_k > s_i$. Therefore, in this round of iteration, we will pick an example from $B^2$. In order for $tr'$ to be equal to $\alpha$, the value of $t$ would be $\lceil \frac{\alpha}{r'} \rceil \leq \lceil \frac{2\alpha}{r_2} \rceil$.

If we further increase $t$ so that $tr' = w\alpha$, where $w > 1$, we have the following conclusion: $\forall x_i, x_k \in S$, s.t., $\|x_i - b\| > 2r$ and $\|x_i - x_k\| \leq w\alpha$, $\frac{n_i}{n} - \frac{n_k}{n} \leq (w+2)(1-p_2)\beta V(r)$, i.e., $\frac{s_i}{n} \leq (w+2)(1-p_2)\beta V(r)$. As long as $p_2 \geq \frac{(w+2)(1-p_2)p_2^2}{2^d}$, i.e., $w \leq \frac{2^d}{p_2(1-p_2)} - 2$, then $\forall x_i \in S$, $\|x_i - b\| > 2r$, $s_k > s_i$, and we will pick examples from $B^2$. Since $r' \leq r$, the algorithm will continue querying examples in $B^2$ until the $\lfloor (\frac{2^d}{p_2(1-p_2)} - 2) \cdot \frac{\alpha}{r} \rfloor^{\text{th}}$ iteration.

Finally, we show that the probability of picking a minority class example from $B^2$ is at least $\frac{1}{3}$. To this end, we need to calculate the maximum probability mass of the majority class within $B^2$. Consider the case where the maximum value of $f_1(x)$ occurs at $b$, and this pdf decreases by $\beta$ every time $x$ moves away from $b$ in the direction of the radius by $\alpha$, i.e., the shape of $f_1(x)$ is a cone in $(d+1)$ dimensional space. Since $f_1(x)$ must integrate to 1, i.e., $V(\frac{\alpha f_1(b)}{\beta}) \cdot \frac{f_1(b)}{d+1} = 1$, where $V(\frac{\alpha f_1(b)}{\beta})$ is the volume of a hyper-ball with radius $\frac{\alpha f_1(b)}{\beta}$, we have $f_1(b) = (\frac{d+1}{V(\alpha)})^{\frac{1}{d+1}} \beta^{\frac{d}{d+1}}$. Therefore, the probability mass of the majority class within

$B^2$ is:

$$\begin{aligned}
&V(2r)(f_1(b) - \frac{2r}{\alpha}\beta) + \frac{2r}{\alpha}\frac{\beta}{d+1}V(2r) < V(2r)f_1(b) \\
&= V(2r)(\frac{d+1}{V(\alpha)})^{\frac{1}{d+1}}\beta^{\frac{d}{d+1}} = 2^d\frac{V(r)}{V(\alpha)^{\frac{1}{d+1}}}(d+1)^{\frac{1}{d+1}}\beta^{\frac{d}{d+1}} \\
&< (d+1)^{\frac{1}{d+1}}(2^{d+1}V(r)\beta)^{\frac{d}{d+1}} \leq (d+1)^{\frac{1}{d+1}}(\frac{p_2^2 \cdot OV(\frac{r_2}{2}, r)}{V(r)})^{\frac{d}{d+1}} < 2p_2
\end{aligned}$$

where $V(2r)$ is the volume of a hyper-ball with radius $2r$. Therefore, if we select a point at random from $B^2$, the probability that this point is from the minority class is at least $\frac{p_2}{p_2+(1-p_2)\cdot 2p_2} \geq \frac{p_2}{p_2+2p_2} = \frac{1}{3}$. □

### 3.1.2 Rare Category Detection for Multiple Classes

In many real applications, there are often more than one minority classes[2]. Therefore, we need to develop an algorithm that is able to discover examples from all the minority classes.

#### Algorithm

In Subsection 3.1.1, we have discussed about rare category detection for the binary case. In this subsection, we focus on the case where $m > 2$. To be specific, let $p_1, \ldots, p_m$ be the priors of the $m$ classes, and $p_1 \gg p_c$, $c \neq 1$. Our goal is to use as few label requests as possible to find at least one example from each class.

The *NNDB* algorithm proposed in Subsection 3.1.1 can be easily generalized to multiple classes, which is presented in Alg. 2. It works as follows. Given the priors for the minority classes, we first estimate the number $K_c$ of examples from class $c$ in the set $S$. Then, for class $c$, at each example, we record its distance from the $K_c^{\text{th}}$ nearest neighbor. The minimum distance over all the examples is the class specific radius, and is assigned to $r_c'$. Next, we draw a hyper-ball centered at example $x_i$ with radius $r_c'$, and count the number of examples enclosed by this hyper-ball, which is denoted as $n_i^c$. $n_i^c$ is roughly in proportion to the local density. To find examples from class $c$, in each iteration of Step 10, we subtract the local density of neighboring points from $n_i^c$, and let the maximum value be the score of $x_i$. The example with the maximum score is selected for labeling by the oracle. If the example is from class $c$, stop the iteration; otherwise, enlarge the neighborhood where the scores of the examples are re-calculated and continue.

#### Justification

Similarly as before, next we prove that if the minority classes are concentrated in small regions and the pdf of the majority class is locally smooth, the proposed algorithm will repeatedly sample in the regions where the rare examples occur with a high probability.

First of all, we make the following specific assumptions.

**Assumptions**

1. The pdf $f_c(x)$ of minority class $c$ is uniform within a hyper-ball $B_c$ of radius $r_c$ [3] centered at $b_c$, $c = 2, \ldots, m$, i.e., $f_c(x) = \frac{1}{V(r_c)}$, if $x \in B_c$; and 0 otherwise, where $V(r_c) \propto r_c^d$ is the volume of $B_c$.

[2] As discussed in Section 1.6, more than one majority classes can be seen as a single majority class if they all satisfy the smoothness assumption introduced in Section 1.3

[3] This is the actual radius, as opposed to the class specific radius $r_c'$.

**Algorithm 2** Active Learning for Initial Class Exploration (*ALICE*)

```
Input: $S, p_2, \ldots, p_m$
Initialize all the minority classes as undiscovered.
for $c = 2 : m$ do
    Let $K_c = np_c$, where $n$ is the number of examples.
    For each example, calculate the distance between this example and its $K_c^{\text{th}}$ nearest neighbor. Set $r'_c$ to be the minimum value among all the examples.
end for
for $c = 2 : m$ do
    $\forall x_i \in S$, let $NN(x_i, r'_c) = \{x | x \in S, \|x - x_i\| \leq r'_c\}$, and $n_i^c = |NN(x_i, r'_c)|$.
end for
for $c = 2 : m$ do
    If class $c$ has been discovered, continue.
    for $t = 2 : n$ do
        For each $x_i$ that has been selected, $s_i^c = -\infty$; for all the other examples, $s_i^c = \max\limits_{x_k \in NN(x_i, tr'_c)} (n_i^c - n_k^c)$.
        Select and query the label of $x = \arg\max_{x_i \in S} s_i^c$.
        If the label of $x$ is equal to $c$, break; otherwise, mark the class that $x$ belongs to as discovered.
    end for
end for
```

2. $f_1(x)$ is bounded and positive in $B_c$, $c = 2, \ldots, m$, i.e., $f_1(x) \geq \frac{\kappa_{c1} p_c}{p_1 V(r_c)}$, $\forall x \in B_c$ and $f_1(x) \leq \frac{\kappa_{c2} p_c}{p_1 V(r_c)}$, $\forall x \in \mathbb{R}^d$, where $\kappa_{c1}, \kappa_{c2} > 0$ are two constants.[4]

Furthermore, for each minority class $c$, $c = 2, \ldots, m$, let $r_{c2} = \frac{r_c}{(1+\kappa_{c2})^{\frac{1}{d}}}$; and let $OV(\frac{r_{c2}}{2}, r_c)$ be the volume of the overlapping region of two hyper-balls: one is of radius $r_c$; the other one is of radius $\frac{r_{c2}}{2}$, and its center is on the sphere of the previous one.

To prove the performance of the proposed *ALICE* algorithm, we first have the following lemma.

**Lemma 2.** *For each minority class c,* $c = 2, \ldots, m$*,* $\forall \epsilon_c, \delta_c > 0$*, if* $n \geq \max\{\max_{c=2}^m \frac{1}{2\kappa_{c1}^2 p_c^2} \log \frac{3m-3}{\delta}, \max_{c=2}^m \frac{1}{2(1-2^{-d})^2 p_c^2} \log \frac{3m-3}{\delta}, \max_{c=2}^m \frac{1}{\epsilon^4 V(\frac{r_{c2}}{2})^4} \log \frac{3m-3}{\delta}\}$*, then with probability at least* $1-\delta$*,* $\frac{r_{c2}}{2} \leq r'_c \leq r_c$ *and* $|\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| \leq \epsilon V(r'_c)$*,* $1 \leq j \leq n$*.*

*Proof.* First, notice that for each minority class $c$, the expected proportion of points falling inside $B_c$, $E(\frac{|NN(b_c, r_c)|}{n}) \geq (\kappa_{c1} + 1)p_c$, and that the maximum expected proportion of points falling inside any

[4] Notice that here we are only dealing with the hard case where $f_1(x)$ is positive within $B_c$. In the separable case where the support regions of the majority and the minority classes do not overlap, we can use other methods to detect the minority classes, such as the one proposed in [Pelleg & Moore, 2004].

hyper-ball of radius $\frac{r_{c2}}{2}$, $\max_{x\in\mathbb{R}^d}[E(\frac{|NN(x,\frac{r_{c2}}{2})|}{n})] \leq 2^{-d}p_c$. Then

$$\begin{aligned}
&\Pr[\exists c,\ \text{s.t.},\ r'_c > r_c \text{ OR } \exists c,\ \text{s.t.},\ r'_c < \frac{r_{c2}}{2} \text{ OR } \exists c, \exists x_i \in S \text{ s.t.}, |\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| > \epsilon V(r'_c)]\\
&\leq \sum_{c=2}^{m} \Pr[r'_c > r_c] + \sum_{c=2}^{m} \Pr[r'_c < \frac{r_{c2}}{2}] + \sum_{c=2}^{m} \Pr[r'_c \geq \frac{r_{c2}}{2} \text{ AND } \exists x_i \text{ s.t.}, |\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| > \epsilon V(r'_c)]\\
&\leq \sum_{c=2}^{m} \Pr[|NN(b_c, r_c)| < K_c] + \sum_{c=2}^{m} \Pr[\max_{x\in\mathbb{R}^d} |NN(x, \frac{r_{c2}}{2})| > K_c] + \sum_{c=2}^{m} n \Pr[|\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| > \epsilon V(r'_c) | r'_c \geq \frac{r_{c2}}{2}]\\
&= \sum_{c=2}^{m} \Pr[|\frac{NN(b_c, r_c)}{n}| < p_c] + \sum_{c=2}^{m} \Pr[\max_{x\in\mathbb{R}^d} |\frac{NN(x, \frac{r_{c2}}{2})}{n}| > p_c] + n\sum_{c=2}^{m} \Pr[|\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| > \epsilon V(r'_c) | r'_c \geq \frac{r_{c2}}{2}]\\
&\leq \sum_{c=2}^{m} e^{-2n\kappa_{c1}^2 p_c^2} + \sum_{c=2}^{m} e^{-2n(1-2^{-d})^2 p_c^2} + 2n\sum_{c=2}^{m} e^{-2n\epsilon^2 V(r'_c)^2}
\end{aligned}$$

where the last inequality is based on Hoeffding bound.

Let $e^{-2n\kappa_{c1}^2 p_c^2} \leq \frac{\delta}{3m-3}$, $e^{-2n(1-2^{-d})^2 p_c^2} \leq \frac{\delta}{3m-3}$ and $2ne^{-2n\epsilon^2 V(r'_c)} \leq 2ne^{-2n\epsilon^2 V(\frac{r_{c2}}{2})^2} \leq \frac{\delta}{3m-3}$, we obtain $n \geq \frac{1}{2\kappa_{c1}^2 p_c^2} \log \frac{3m-3}{\delta}$, $n \geq \frac{1}{2(1-2^{-d})^2 p_c^2} \log \frac{3m-3}{\delta}$, and $n \geq \frac{1}{\epsilon^4 V(\frac{r_{c2}}{2})^4} \log \frac{3m-3}{\delta}$. □

Based on Lemma 2, we get the following theorem, which shows the effectiveness of the proposed algorithm.

**Theorem 2.** *If*

1. *For minority class $c$, $c = 2, \ldots, m$, let $B_c^2$ be the hyper-ball centered at $b_c$ with radius $2r_c$. The minimum distance between the points inside $B_c$ and the ones outside $B_c^2$ is not too large, i.e.,* $\max_{c=2}^{m} \min\{\|x_i - x_k\| | x_i, x_k \in S, \|x_i - b_c\| \leq r_c, \|x_i - b_c\| > 2r_c\} \leq \alpha$.
2. *The minority classes are far apart, i.e., if $x_i, x_k \in S$, $\|x_i - b_c\| \leq r_c$, $\|x_k - b_{c'}\| \leq r_{c'}$, $c, c' = 2, \ldots, m$, and $c \neq c'$, then $\|x_i - x_k\| > \alpha$.*
3. *$f_1(x)$ is locally smooth, i.e., $\forall x, y \in \mathbb{R}^d, |f_1(x) - f_1(y)| \leq \frac{\beta\|x-y\|}{\alpha}$, where $\beta \leq \min_{c=2}^{m} \frac{p_c^2 OV(\frac{r_{c2}}{2}, r_c)}{2^{d+1}V(r_c)^2}$.*
4. *The number of examples is sufficiently large, i.e., $n \geq \max\{\max_{c=2}^{m} \frac{1}{2\kappa_{c1}^2 p_c^2} \log \frac{3m-3}{\delta}$, $\max_{c=2}^{m} \frac{1}{2(1-2^{-d})^2 p_c^2} \log \frac{3m-3}{\delta}, \max_{c=2}^{m} \frac{1}{p_1^4 \beta^4 V(\frac{r_{c2}}{2})^4} \log \frac{3m-3}{\delta}\}$.*

*then with probability at least $1 - \delta$, in every iteration of Step 8, after $\lceil \frac{2\alpha}{r_{c2}} \rceil$ rounds of Step 10,* ALICE*will query at least one example whose probability of coming from a minority class is at least $\frac{1}{3}$.*

*Proof.* Based on this Lemma 2, using condition 4, let $\epsilon = p_1\beta$, if the number of examples is sufficiently large, then with probability at least $1 - \delta$, for each minority class $c$, $c = 2, \ldots, m$, $\frac{r_{c2}}{2} \leq r'_c \leq r$ and $|\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| \leq p_1\beta V(r'_c)$, $1 \leq i \leq n$.

To better prove the theorem, given a point $x_i \in S$, we say that $x_i$ is 'far from all the minority classes' iff for every minority class $c$, $\|x_i - b_c\| > 2r_c$, i.e., $x_i$ is not within $B_c^2$. According to condition 3, $\forall x_i, x_k \in S$ s.t., $x_i$ and $x_k$ are far from all the minority classes and $\|x_i - x_k\| \leq \alpha$, $E(\frac{n_i^c}{n})$ and $E(\frac{n_k^c}{n})$ will not be affected by the minority classes. Therefore, in iteration $i$ of Step 8 where we aim to find examples from minority class $c$, $|E(\frac{n_i^c}{n}) - E(\frac{n_k^c}{n})| \leq p_1\beta V(r'_c) \leq p_1\beta V(r_c)$. Furthermore, since $\alpha$ is always bigger than $r_c$, we have

$$|\frac{n_i^c}{n} - \frac{n_k^c}{n}| \leq |\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| + |\frac{n_k^c}{n} - E(\frac{n_k^c}{n})| + |E(\frac{n_i^c}{n}) - E(\frac{n_i^c}{n})| \leq 3p_1\beta V(r_c) \qquad (3.4)$$

From Inequality 3.4, it is not hard to see that $\forall x_i, x_k \in S$, s.t., $x_i$ is far from all the minority classes and $\|x_i - x_k\| \leq \alpha$, $\frac{n_i^c}{n} - \frac{n_k^c}{n} \leq 3p_1\beta V(r_c)$, i.e., when $tr'_c = \alpha$,

$$\frac{s_i^c}{n} \leq 3p_1\beta V(r_c) \tag{3.5}$$

This is because if $x_k$ is not far from any of the minority classes, the minority classes may also contribute to $\frac{n_k^c}{n}$, and thus the score of $x_i$ may be even smaller.

On the other hand, based on conditions 1 and 2, there exist two points $x_u, x_v \in S$, s.t., $\|x_u - b_c\| \leq r_c$, $x_v$ is far from all the minority classes, and $\|x_u - x_v\| \leq \alpha$. Since the contribution of minority class $c$ to $E(\frac{n_u^c}{n})$ is at least $\frac{p_c \cdot OV(\frac{r_{c2}}{2}, r_c)}{V(r_c)}$, so $E(\frac{x_u^c}{n}) - E(\frac{x_v^c}{n}) \geq \frac{p_c \cdot OV(\frac{r_{c2}}{2}, r_c)}{V(r_c)} - p_1\beta V(r'_c) \geq \frac{p_c \cdot OV(\frac{r_{c2}}{2}, r_c)}{V(r_c)} - p_1\beta V(r_c)$. Since for any example $x_i \in S$, we have $|\frac{n_i^c}{n} - E(\frac{n_i^c}{n})| \leq p_1\beta V(r'_c) \leq p_1\beta V(r_c)$, therefore

$$\begin{aligned}\frac{n_u}{n} - \frac{n_v}{n} &\geq \frac{p_c \cdot OV(\frac{r_{c2}}{2}, r_c)}{V(r_c)} - 3p_1\beta V(r_c)\\ &\geq \frac{p_c \cdot OV(\frac{r_{c2}}{2}, r_c)}{V(r_c)} - \frac{3p_1 p_c^2 \cdot OV(\frac{r_{c2}}{2}, r_c)}{2^{d+1}V(r_c)}\end{aligned}$$

Since $p_c$ is very small, $p_c \gg \frac{6p_1p_c^2}{2^{d+1}}$; therefore, $\frac{n_u^c}{n} - \frac{n_v^c}{n} > \frac{3p_1p_c^2 \cdot OV(\frac{r_{c2}}{2}, r_c)}{2^{d+1}V(r_c)} \geq 3p_1\beta V(r_c)$, i.e., when $tr'_c = \alpha$,

$$\frac{s_u^c}{n} > 3p_1\beta V(r_c) \tag{3.6}$$

In Step 10 of the proposed method, we gradually enlarge the neighborhood to calculate the change of local density to continue seeking an example of the minority class. When $tr'_c = \alpha$, based on Inequalities 3.5 and 3.6, $\forall x_i \in S$ s.t., $x_i$ is far from all the minority classes, we have $s_u^c > s_i^c$. Therefore, in this round of iteration, we will pick an example that is NOT far from one of the minority classes, i.e., there exists a minority class $c_t$ s.t., the selected example is within $B_{c_t}^2$. Note that $c_t$ is not necessarily equal to $c$, which is the minority class that we would like to discover in Step 8 of the method.

Finally, we show that the probability of picking an example that belongs to minority class $c_t$ from $B_{c_t}^2$ is at least $\frac{1}{3}$. To this end, we need to calculate the maximum probability mass of the majority class within $B_{c_t}^2$. Consider the case where the maximum value of $f_1(x)$ occurs at $b_{c_t}$, and this pdf decreases by $\beta$ every time $x$ moves away from $b_{c_t}$ in the direction of the radius by $\alpha$, i.e., the shape of $f_1(x)$ is a cone in $(d+1)$ dimensional space. Since $f_1(x)$ must integrate to 1, i.e., $V(\frac{\alpha f_1(b_{c_t})}{\beta}) \cdot \frac{f_1(b_{c_t})}{d+1}$, where $V(\frac{\alpha f_1(b_{c_t})}{\beta})$ is the volume of a hyper-ball with radius $\frac{\alpha f_1(b_{c_t})}{\beta}$, we have $f_1(b_{c_t}) = (\frac{d+1}{V(\alpha)})^{\frac{1}{d+1}}\beta^{\frac{d}{d+1}}$. Therefore, the probability mass of the majority class within $B_{c_t}^2$ is:

$$\begin{aligned}&V(2r_{c_t})(f_1(b_{c_t}) - \frac{2r_{c_t}}{\alpha}\beta) + \frac{2r_{c_t}}{\alpha}\frac{\beta}{d+1}V(2r_{c_t})\\ &< V(2r_{c_t})f_1(b_{c_t}) = V(2r_{c_t})(\frac{d+1}{V(\alpha)})^{\frac{1}{d+1}}\beta^{\frac{d}{d+1}}\\ &= 2^d\frac{V(r_{c_t})}{(V(\alpha))^{\frac{1}{d+1}}}(d+1)^{\frac{1}{d+1}}\beta^{\frac{d}{d+1}}\\ &< (d+1)^{\frac{1}{d+1}}(2^{d+1}V(r_{c_t})\beta)^{\frac{d}{d+1}}\\ &\leq (d+1)^{\frac{1}{d+1}}(\frac{p_{c_t}^2 \cdot OV(\frac{r_{c_t2}}{2}, r_{c_t})}{V(r_{c_t})})^{\frac{d}{d+1}} < 2p_{c_t}\end{aligned}$$

where $V(2r_{c_t})$ is the volume of a hyper-ball with radius $2r_{c_t}$. Therefore, if we select a point at random from $B_{c_t}^2$, the probability that this point is from minority class $c_t$ is at least $\frac{p_{c_t}}{p_{c_t}+p_1 \cdot 2p_{c_t}} \geq \frac{p_{c_t}}{p_{c_t}+2p_{c_t}} = \frac{1}{3}$. □

**Implementation Issues**

According to our theorem, in each iteration of Step 8, with high probability, we may pick examples belonging to the rare classes after selecting a small number of examples. However, the discovered rare class $c_t$ may not be the same as the rare class $c$ that we hope to discover in this iteration of Step 8. Furthermore, we may repeatedly select examples from class $c_t$ before finding one example from class $c$. To address these issues, we have modified the original *ALICE* algorithm to produce *MALICE*, which is shown in Alg. 3.

**Algorithm 3** Modified Active Learning for Initial Class Exploration (*MALICE*)

```
Input: $S, p_2, \ldots, p_m$
 1: Initialize all the rare classes as undiscovered.
 2: for $c = 2 : m$ do
 3:    Let $K_c = np_c$.
 4:    For each example, calculate the distance between this example and its $K_c^{\text{th}}$ nearest neighbor. Set $r'_c$
       to be the minimum value among all the examples.
 5: end for
 6: Let $r'_1 = \max_{c=2}^{m} r'_c$.
 7: for $c = 2 : m$ do
 8:    $\forall x_i \in S$, let $NN(x_i, r'_c) = \{x | x \in S, \|x - x_i\| \leq r'_c\}$, and $n_i^c = |NN(x_i, r'_c)|$.
 9: end for
10: for $c = 2 : m$ do
11:    If class $c$ has been discovered, continue.
12:    for $t = 2 : n$ do
13:       For each $x_i$ that has been selected, $\forall x_k \in S$, s.t., $\|x_i - x_k\| \leq r'_{y_i}$, $s_k^c = -\infty$; for all the other
          examples, $s_i^c = \max_{x_k \in NN(x_i, tr'_c)} (n_i^c - n_k^c)$.
14:       Select and query the label of $x = \arg\max_{x_i \in S} s_i^c$.
15:       If the label of $x$ is equal to $c$, break; otherwise, $t = t - 1$, mark the class that $x$ belongs to as
          discovered.
16:    end for
17: end for
```

There are two major differences between *MALICE* and *ALICE*. (1) In Step 12 of *MALICE*, once we have labeled an example, any unlabeled example within the class specific radius of this example will be precluded from selection. Since we have proved that with high probability, the class specific radius is less than the actual radius, this modification will help prevent examples of the same class from being selected repeatedly. (2) In Step 14 of *MALICE*, if the labeled example belongs to a rare class other than class $c$, we will not enlarge the neighborhood based on which the scores of the examples are re-calculated. This is to increase the chance that if $tr'_c$ is close to $\alpha$, we will select examples from $B_c^2$.

### 3.1.3 Experimental Results

In this subsection, we compare our algorithms (*NNDB* and *MALICE*) with the best method proposed in [Pelleg & Moore, 2004] (Interleave) and random sampling (RS) on both synthetic and real data sets. In Interleave, we use the number of classes as the number of components in the mixture model. For both Interleave and RS, we run the experiments 100 times and report the average results.

### Synthetic Data Sets

Fig. 3.1(a) shows a synthetic data set where the pdf of the majority class is Gaussian and the pdf of the minority class is uniform within a small hyper-ball. There are 1000 examples from the majority class and only 10 examples from the minority class. Using Interleave, we need to label 35 examples, using RS, we need to label 101 examples, and using *NNDB*, we only need to label 3 examples in order to sample one from the minority class, which are denoted as 'x's in Fig. 3.1(b). Notice that the first 2 examples that *NNDB* selects are not from the correct region. This is because the number of examples from the minority class is very small, and the local density may be affected by the randomness in the data.

In Fig. 3.2(a), the X-shaped data consisting of 3000 examples correspond to the majority class, and the four characters 'NIPS' correspond to four minority classes, which consist of 138, 79, 118, and 206 examples respectively. Using Interleave, we need to label 1190 examples, using RS, we need to label 83 examples, and using *MALICE*, we only need to label 5 examples in order to get one from each of the minority classes, which are denoted as 'x's in Fig. 3.2(b). Notice that in this example, Interleave is even worse than RS. This might be because some minority classes are located in the region where the density of the majority class is not negligible, and thus may be 'explained' by the majority-class mixture-model component.

### Real Data Sets

In this subsection, we compare different methods on two real data sets: Abalone [Asuncion & Newman, 2007] and Shuttle [Asuncion & Newman, 2007]. The first data set consists of 4177 examples, described by 7 dimensional features. The examples come from 20 classes: the proportion of the largest class is 16.50%, and the proportion of the smallest class is 0.34%. For the second data set, we sub-sample the original training set to produce a smaller data set with 4515 examples, described by 9 dimensional features. The examples come from 7 classes: the proportion of the largest class is 75.53%, and the proportion of the smallest class is 0.13%.

The comparison results are shown in Fig. 3.3 and Fig. 3.4 respectively. Further, Table 3.1 and Table 3.2 show the confidence intervals for the number of label requests needed to discover a certain number of classes in the data set for different methods. This is obtained via bootstrapping[5]. Notice that the number of classes in the two tables always starts from 2, since with all the methods, the first label request always results in a new class being discovered. From these results, we have the following observations. First, when the number of classes discovered is small, the average number of label requests by RS is smaller than *MALICE*; however, we are only interested in the classes whose proportions are much smaller than the rest, and to discover those classes, RS needs much more label requests compared with *MALICE*. Second, the performance of Interleave is not as good as *MALICE* in terms of the average number of label requests on the two data sets. This is because as the number of components becomes larger, the mixture model generated by Interleave is less reliable due to the lack of labeled examples, thus we need to select more examples. Furthermore, the majority and minority classes may not be near-separable, which is a disaster for Interleave. On the other hand, *MALICE* does not assume a generative model for the data, and only focuses on the change in local density, which is more effective on the two data sets. Finally, the confidence intervals by *MALICE* are always significantly smaller than those by both Interleave and RS, showing that the performance of *MALICE* is much more stable.

[5] Notice that different runs of bootstrapping are not independent of each other. However, since *MALICE* is a determinist method, bootstrapping may be the only way to obtain its confidence intervals given that we only have one sample from each domain

Table 3.1: Confidence intervals for the number of label requests needed by different methods to discover a certain number of classes in the Abalone data set.

| No. of Classes | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *MALICE* | 2±0.00 | 3±0.00 | 4±0.00 | 6±0.42 | 13±2.20 | 14±3.10 | 15±3.62 | 18±4.58 | 25±5.80 | 31±8.12 | 55±10.10 | 56±11.84 | 57±13.03 | 59±13.46 | 61±16.50 | 62±17.06 | 108±29.02 | 114±35.74 | 125±40.03 |
| Interleave | 2±0.00 | 4±0.50 | 7±2.96 | 37±9.11 | 38±11.37 | 39±14.90 | 40±16.76 | 42±21.29 | 53±25.95 | 76±29.50 | 78±34.75 | 88±39.11 | 89±49.69 | 95±61.92 | 108±80.80 | 138±94.93 | 206±119.71 | 248±152.71 | 270±196.23 |
| RS | 2±0.00 | 4±0.75 | 5±0.45 | 8±2.25 | 11±4.18 | 12±4.55 | 14±6.94 | 15±9.11 | 22±12.79 | 23±16.74 | 40±21.87 | 63±28.50 | 67±35.47 | 71±42.81 | 104±54.43 | 136±73.05 | 145±101.53 | 273±138.06 | 483±165.23 |

Table 3.2: Confidence intervals for the number of label requests needed by different methods to discover a certain number of classes in the Shuttle data set.

| No. of Classes | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| *MALICE* | 6±2.61 | 17±3.95 | 48±8.64 | 49±9.92 | 76±5.77 | 87±12.64 |
| Interleave | 2±0.00 | 52±9.90 | 107±16.15 | 109±30.59 | 115±45.30 | 140±58.83 |
| RS | 7±2.85 | 9±5.14 | 13±8.82 | 63±27.51 | 100±57.92 | 512±109.56 |

**Imprecise Priors**

The proposed algorithms need the priors of the minority classes as input. In this subsection, we test the robustness of *MALICE* against modest mis-estimations of the class priors. The performance of *NNDB* is similar to *MALICE* so we omit the results here. In the experiments, we use the same data sets as in Subsubsection 3.1.3, and add/subtract 5%, 10%, and 20% from the true priors of the minority classes. The results are shown in Fig. 3.5 and Fig. 3.6. Further, Table 3.3 and Table 3.4 show the confidence intervals for the number of label requests needed by *MALICE* with different priors to discover a certain number of classes in the data set. This is obtained via bootstrapping. From these results, we can see that *MALICE* is very robust to small perturbations in the priors. For example, with Abalone data set, if we subtract 10% from the true priors, only one more label request is needed in order to find all the classes. Furthermore, small perturbations in the priors do not affect the width of the confidence intervals by a large margin.

Table 3.3: Confidence intervals for the number of label requests needed by *MALICE* with different priors to discover a certain number of classes in the Abalone data set.

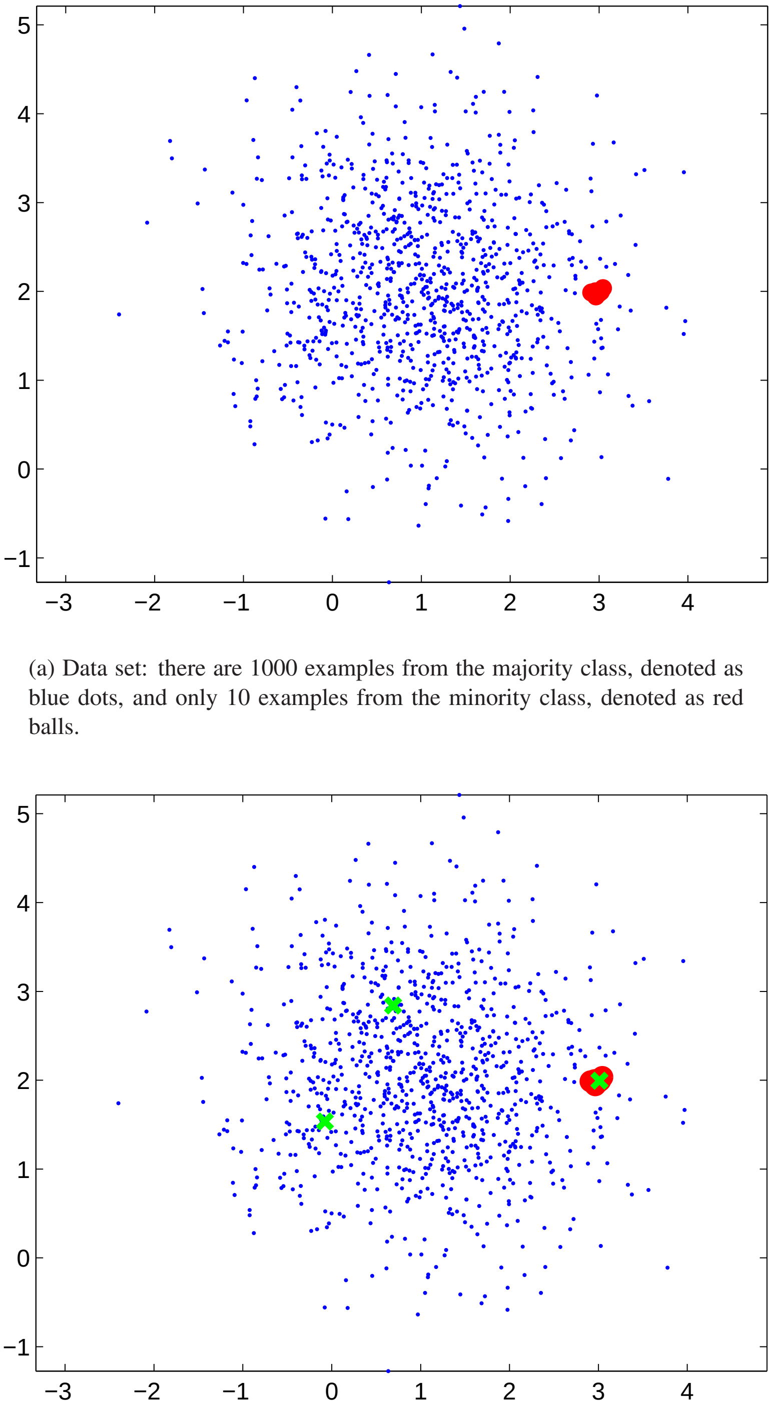
| No. of Classes | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| *MALICE* | 2±0.00 | 3±0.00 | 4±0.00 | 6±0.42 | 13±2.20 | 14±3.10 |
| −20% | 2±0.00 | 3±0.00 | 5±0.25 | 8±2.18 | 14±3.05 | 15±3.69 |
| −10% | 2±0.00 | 3±0.00 | 5±0.19 | 13±1.86 | 16±2.62 | 17±3.20 |
| −5% | 2±0.00 | 3±0.00 | 5±0.96 | 7±1.47 | 13±2.11 | 14±3.16 |
| +20% | 2±0.00 | 3±0.00 | 5±0.02 | 6±0.62 | 10±2.16 | 11±3.18 |
| +10% | 2±0.00 | 3±0.00 | 4±0.00 | 9±1.55 | 12±2.14 | 13±3.24 |
| +5% | 2±0.00 | 3±0.00 | 4±0.00 | 13±1.60 | 14±2.23 | 15±3.22 |

| No. of Classes | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|
| *MALICE* | 15±3.62 | 18±4.58 | 25±5.80 | 31±8.12 | 55±10.10 | 56±11.84 |
| −20% | 16±3.84 | 24±4.62 | 26±7.36 | 31±9.22 | 64±11.46 | 65±13.96 |
| −10% | 22±4.16 | 23±5.30 | 29±6.32 | 35±8.06 | 57±10.71 | 58±13.18 |
| −5% | 15±4.00 | 19±4.72 | 26±6.63 | 31±8.65 | 56±10.70 | 57±11.33 |
| +20% | 15±4.05 | 20±4.65 | 32±6.62 | 38±7.56 | 39±8.07 | 40±8.87 |
| +10% | 14±3.78 | 16±4.78 | 17±5.87 | 20±7.96 | 49±10.38 | 50±12.47 |
| +5% | 16±4.01 | 17±5.29 | 21±6.38 | 26 ±7.58 | 51±9.03 | 52±10.59 |

| No. of Classes | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|
| *MALICE* | 57±13.03 | 59±13.46 | 61±16.50 | 62±17.06 | 108±29.02 | 114±35.74 | 125±40.03 |
| −20% | 66±16.37 | 70±19.54 | 72±22.21 | 73±21.80 | 74±34.68 | 135±52.07 | 230±61.22 |
| −10% | 59±14.23 | 60±14.56 | 62±19.30 | 65±17.77 | 114±30.04 | 117±43.54 | 126±46.66 |
| −5% | 58±13.11 | 60±17.08 | 62±18.90 | 63±17.53 | 107±29.29 | 109±40.29 | 116±51.17 |
| +20% | 41±11.40 | 60±13.80 | 73±16.32 | 77±23.88 | 80±28.08 | 82±35.04 | 141±41.34 |
| +10% | 51±12.39 | 53±15.64 | 88±16.96 | 93±25.83 | 94±34.57 | 101±36.38 | 171±48.17 |
| +5% | 53±13.74 | 55±16.27 | 56±16.77 | 57±21.88 | 95±31.84 | 97±39.92 | 105±48.60 |

Table 3.4: Confidence intervals for the number of label requests needed by *MALICE* with different priors to discover a certain number of classes in the Shuttle data set.

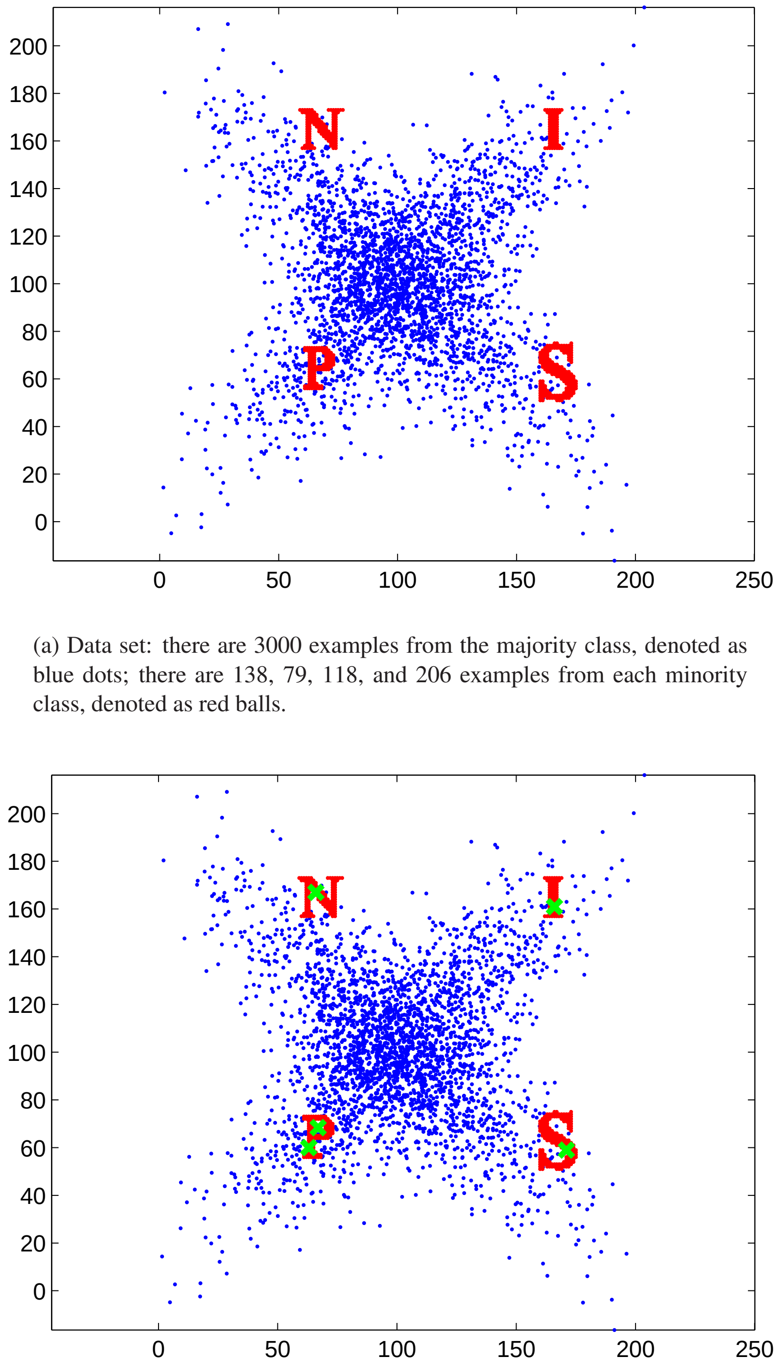
| No. of Classes | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| *MALICE* | 6±2.61 | 17±3.95 | 48±8.64 | 49±9.92 | 76±5.77 | 87±12.64 |
| −20% | 7±3.08 | 10±4.36 | 47±9.26 | 48±11.37 | 74±6.60 | 85±14.40 |
| −10% | 7±2.76 | 18±4.07 | 52±8.28 | 53±10.48 | 80±6.70 | 91±14.62 |
| −5% | 7±2.71 | 18±4.10 | 50±8.10 | 51±9.98 | 78±6.19 | 89±13.87 |
| +20% | 5±2.45 | 9±3.36 | 28±7.30 | 46±7.52 | 65±9.90 | 76±9.85 |
| +10% | 7±2.18 | 8±3.43 | 46±7.80 | 47±8.61 | 71±7.51 | 82±12.38 |
| +5% | 7±2.31 | 8±3.77 | 47±8.12 | 48±9.26 | 72±7.72 | 83±12.07 |



(a) Data set: there are 1000 examples from the majority class, denoted as blue dots, and only 10 examples from the minority class, denoted as red balls.

(b) Examples selected by *NNDB*, denoted as green ‘x’s.

Figure 3.1: Synthetic data set 1.



(a) Data set: there are 3000 examples from the majority class, denoted as blue dots; there are 138, 79, 118, and 206 examples from each minority class, denoted as red balls.

(b) Examples selected by *MALICE*, denoted as green ‘x’s.

Figure 3.2: Synthetic data set 2.

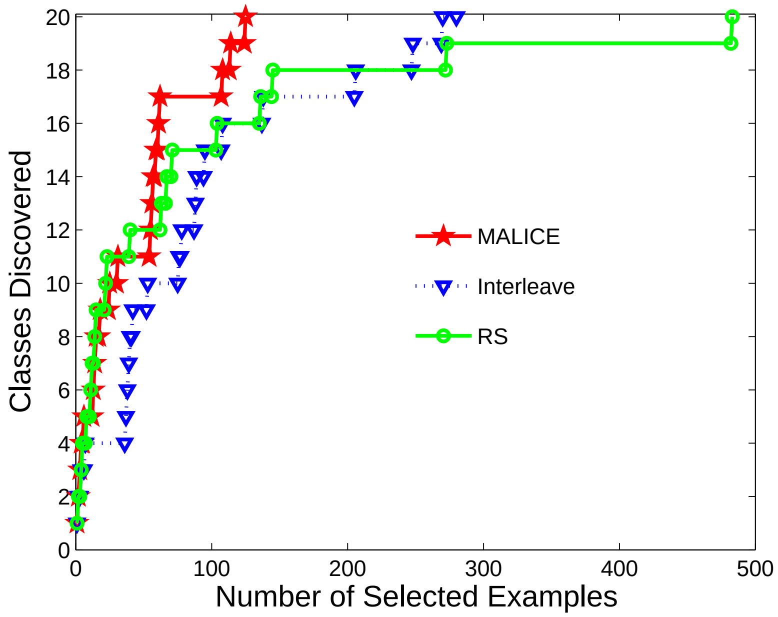


Figure 3.3: Learning curve for Abalone data set.

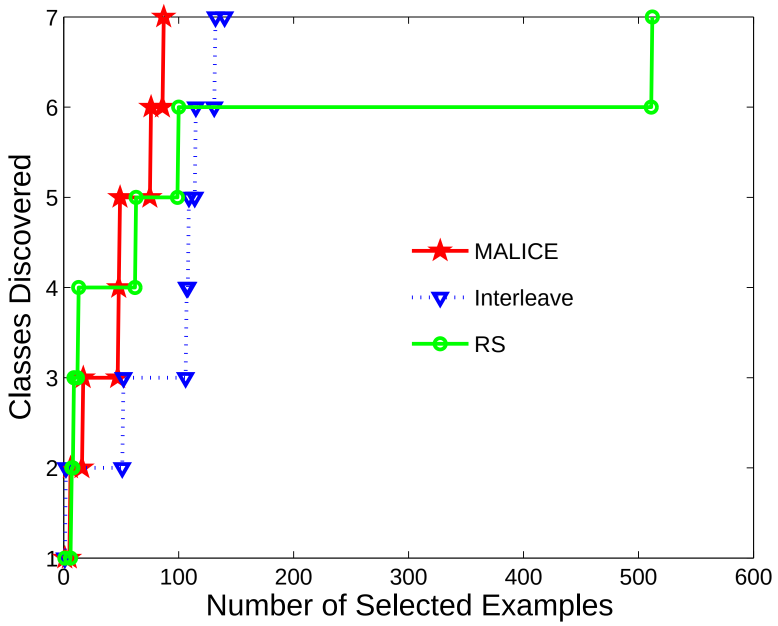


Figure 3.4: Learning curve for Shuttle data set.

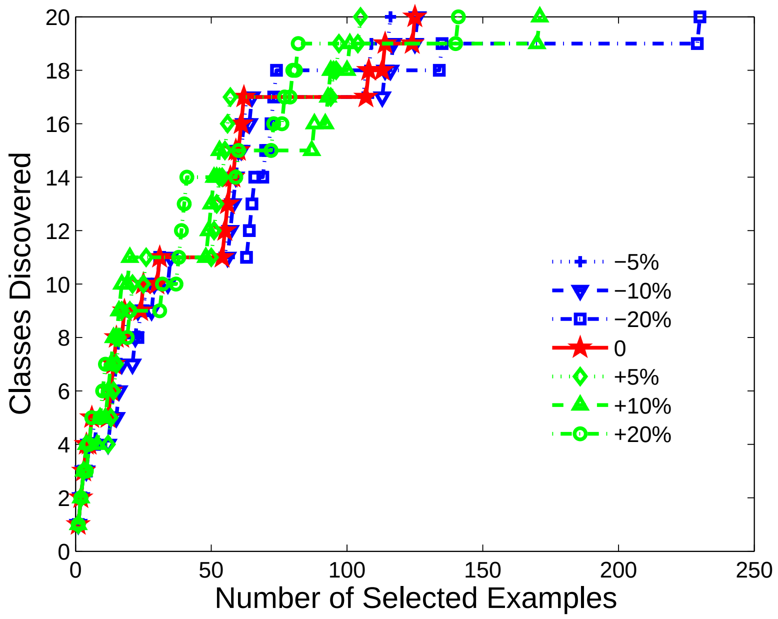


Figure 3.5: Robustness study on Abalone data set: -5%, -10%, and -20% denote the performance of *MALICE* after we subtract 5%, 10%, and 20% from the true priors; +5%, +10%, and +20% denote the performance of *MALICE* after we add 5%, 10%, and 20% to the true priors.

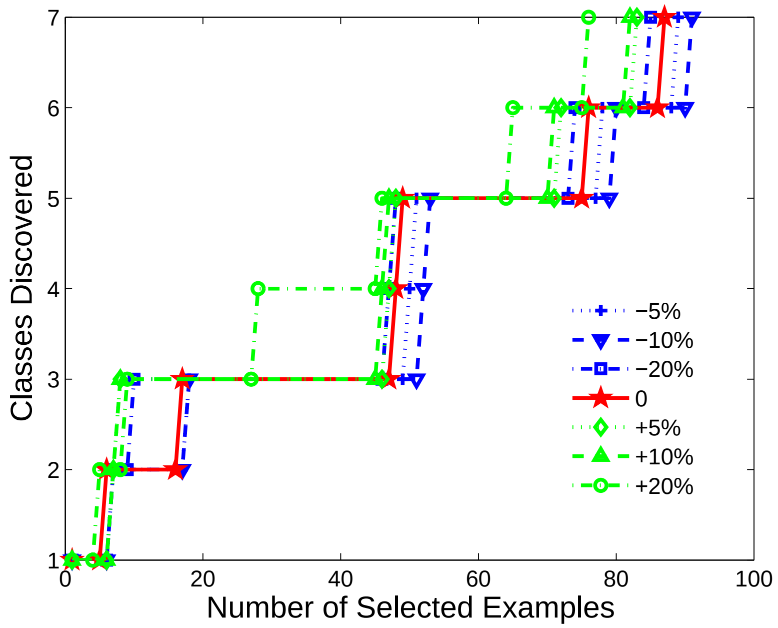


Figure 3.6: Robustness study on Shuttle data set: -5%, -10%, and -20% denote the performance of *MALICE* after we subtract 5%, 10%, and 20% from the true priors; +5%, +10%, and +20% denote the performance of *MALICE* after we add 5%, 10%, and 20% to the true priors.

## 3.2 Prior-free Rare Category Detection for Data with Features

In the last section, we have discussed rare category detection algorithms that need prior information about the data set as input, including the number of classes and the proportions of different classes. However, in some real applications, it is often the case that we do not know the number of classes in the data set, not to mention the proportions of different classes. For example, in outbreaks of epidemics, the number of classes (indicating the number of different diseases) and the class proportions could vary drastically over time as new diseases occur, strain or die out. To address this problem, in this section, we focus on the more challenging case where we do not have any prior information about the data set. The proposed method, SEmiparametric Density Estimation based Rare category detection (*SEDER*), implicitly performs semiparametric density estimation using specially designed exponential families, and selects the examples with the largest norm of the gradients for labeling by the oracle. In this way, it focuses on the areas with the maximum change in the local density. Different from existing methods, *SEDER* does not require any prior information about the data set. Therefore, it is more suitable for real applications.

The rest of the section is organized as follows. In Subsection 3.2.1, we introduce the specially designed exponentially families used in *SEDER* and derive the scoring function. The complete algorithm of *SEDER* is presented in Subsection 3.2.2. Finally, in Subsection 3.2.3, we compare *SEDER* with state-of-the-art techniques on both synthetic and real data sets.

### 3.2.1 Semiparametric Density Estimation for Rare Category Detection

In rare category detection, based on our assumptions introduced in Section 1.3, abrupt changes in local density indicate the presence of rare classes. By sampling in these areas, we have a high probability of finding examples from the rare classes. Following this line of reasoning, our proposed method *SEDER* implicitly estimates the density using specially designed exponential families, which essentially define a semiparametric model. At each data point, we set the score to be the norm of the gradient of the estimated density, which measures the maximum change rate of the local density, and pick the examples with the largest scores to be labeled by the oracle. Although the intuition of *SEDER* and *ALICE* (*MALICE*) is quite similar: to pick the examples with the maximum change in the local density, *ALICE* (*MALICE*) is a nearest-neighbor-based method, it depends on the proportions of different classes to set the size of the neighborhood, and the scores of the examples roughly indicate the change in the local density; whereas *SEDER* is based on semiparametric density estimation, it is prior-free, i.e., it does not require any prior information about the data set, and the scores measure exactly the maximum change rate in the local density.

#### Additional Notation

Besides the general notation introduced in Section 1.6, we further define the following additional notation in Table 3.5.

#### Specially Designed Exponential Families

Traditional density estimation methods belong to two categories [Efron & Tibshirani, 1996]: by fitting a parametric model via maximum likelihood, or by nonparametric methods such as kernel density estimation. For the purpose of rare category detection, parametric models are not appropriate since we can not assume a specific form of the underlying distribution for a given data set. On the other hand, the estimated density based on nonparametric methods tends to be under-smoothed, and the examples from rare classes will be buried among numerous spikes in the estimated density.

Table 3.5: Notation

| Symbol | Definition |
|---|---|
| $g_\beta(x)$ | The density defined by specially designed exponential families |
| $g_0(x)$ | The carrier density |
| $\beta_0$ | The normalizing parameter in $g_\beta(x)$ |
| $t(x)$ | The $p \times 1$ vector of sufficient statistics |
| $t^j(x)$ | The $j^{\text{th}}$ component of $t(x)$ |
| $\beta_1$ | The $p \times 1$ parameter vector |
| $\beta_1^j$ | The $j^{\text{th}}$ component of $\beta_1$ |
| $\sigma^j$ | The bandwidth for the $j^{\text{th}}$ feature |
| $\beta$ | $(\beta_1, \beta_0)$ |
| $\hat{\beta}$ | The maximum likelihood estimate of $\beta$ |
| $l(\beta)$ | The log-likelihood of the data |
| $g_\beta^j(x^j)$ | The marginal distribution of the $j^{\text{th}}$ feature based on $g_\beta(x)$ |
| $g^j(x^j)$ | The true marginal distribution of the $j^{\text{th}}$ feature |
| $b^j$ | Positive parameter which is a function of $\beta_1^j$ |
| $\hat{b}^j$ | The maximum likelihood estimate of $b^j$ |
| $A$ | $\frac{1}{n}\sum_{k=1}^n \frac{\sum_{i=1}^n \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})(x_i^j)^2}{\sum_{i=1}^n \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})}$ |
| $B$ | $(\sigma^j)^2$ |
| $C$ | $\frac{1}{n}\sum_{k=1}^n (x_k^j)^2$ |
| $D_i(x)$ | $\frac{1}{n}\prod_{j=1}^d \frac{1}{\sqrt{2\pi b^j}\sigma^j} \exp(-\frac{(x^j - b^j x_i^j)^2}{2(\sigma^j)^2 b^j})$ |
| $s_i$ | The score of $x_i$ |

As proposed in [Efron & Tibshirani, 1996], these two kinds of methods can be combined by putting an exponential family through a kernel density estimator, the so-called specially designed exponential families. It is a favorable compromise between parametric and nonparametric density estimation: the nonparametric smoother allows local adaptation to the data, while the exponential term matches some of the data's global properties, and makes the density much smoother [Efron & Tibshirani, 1996]. To be specific, the estimated density $g_\beta(x) = g_0(x)\exp(\beta_0 + \beta_1^T t(x))$ [Efron & Tibshirani, 1996]. Here, $x \in \mathbb{R}^d$, $g_0(x)$ is a carrier density, $t(x)$ is a $p \times 1$ vector of sufficient statistics, $\beta_1$ is a $p \times 1$ parameter vector, and $\beta_0$ is a normalizing parameter that makes $g_\beta(x)$ integrate to 1. In our application, we use the kernel density estimator with the Gaussian kernel as the carrier density, i.e., $g_0(x) = \frac{1}{n}\sum_{i=1}^n \prod_{j=1}^d \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2})$, where $x^j$ is the $j^{\text{th}}$ feature of $x$, $x_i^j$ is the $j^{\text{th}}$ feature of the $i^{\text{th}}$ data point, and $\sigma^j$ is the bandwidth for the $j^{\text{th}}$ feature. In *SEDER*, $\sigma^j$ is determined by cross validation [Scott, 1992] on the $j^{\text{th}}$ feature. Here, the parameters $\beta = (\beta_1, \beta_0)$ can be estimated according to the following theorem.

**Theorem 3.** *The maximum likelihood estimate $\hat{\beta}$ of $\beta$ satisfies the following conditions [Efron & Tibshirani, 1996]:* $\forall j \in \{1, \ldots, p\}$

$$\int_{x^1} \cdots \int_{x^d} t^j(x) g_{\hat{\beta}}(x) dx^d \cdots dx^1 = \frac{1}{n}\sum_{i=1}^n t^j(x_i)$$

*where $t^j(x)$ is the $j^{th}$ component of the vector $t(x)$.*

*Proof.* Firstly, notice that $\beta_0$ is a normalizing parameter that makes $g_\beta(x)$ integrate to 1, i.e.,

$$\beta_0 = -\log \int_{x^1} \cdots \int_{x^d} g_0(x) \exp(\beta_1^T t(x)) dx^d \cdots dx^1$$

Therefore, $\forall j \in \{1, \ldots, p\}$

$$\begin{aligned}\frac{\partial \beta_0}{\partial \beta_1^j} &= -\frac{\int_{x^1} \cdots \int_{x^d} t^j(x) g_0(x) \exp(\beta_1^T t(x)) dx^d \cdots dx^1}{\int_{x^1} \cdots \int_{x^d} g_0(x) \exp(\beta_1^T t(x)) dx^d \cdots dx^1} \\ &= -\int_{x^1} \cdots \int_{x^d} t^j(x) g_0(x) \exp(\beta_0 + \beta_1^T t(x)) dx^d \cdots dx^1 \\ &= -\int_{x^1} \cdots \int_{x^d} t^j(x) g_\beta(x) dx^d \cdots dx^1\end{aligned}$$

where $\beta_1^j$ is the $j^{\text{th}}$ component of the vector $\beta_1$.

Secondly, the log-likelihood of the data is $l(\beta) = \sum_{i=1}^n \log(g_\beta(x_i)) = \sum_{i=1}^n \log(g_0(x_i)) + n\beta_0 + \sum_{i=1}^n \beta_1^T t(x_i)$. Taking the partial derivative of $l(\beta)$ with respect to $\beta_1^j$, we have:

$$\frac{\partial l(\beta)}{\partial \beta_1^j} = n\frac{\partial \beta_0}{\partial \beta_1^j} + \sum_{i=1}^n t^j(x_i) = -n \int_{x^1} \cdots \int_{x^d} t^j(x) g_\beta(x) dx^d \cdots dx^1 + \sum_{i=1}^n t^j(x_i)$$

Setting the partial derivative to 0, we have that the maximum likelihood estimate $\hat{\beta}$ of $\beta$ satisfies $\int_{x^1} \cdots \int_{x^d} t^j(x) g_{\hat{\beta}}(x) dx^d \cdots dx^1 = \frac{1}{n}\sum_{i=1}^n t^j(x_i), \forall j \in \{1, \ldots, p\}$. □

In *SEDER*, we set the vector of sufficient statistics to be $t(x) = [(x^1)^2, \ldots, (x^d)^2]^T$ [6]. If we estimate the parameters according to Theorem 3, different parameters will be coupled due to the normalizing parameter $\beta_0$. Let $\beta_1^j$ be the $j^{\text{th}}$ component of the vector $\beta_1$. In order to de-couple the estimation of different $\beta_1^j$s, we make the following changes. Firstly, we decompose $\beta_0$ into $\beta_0^j$s such that $\sum_{j=1}^d \beta_0^j = \beta_0$, then $g_\beta(x)$ can be seen as a kernel density estimator with a 'special' kernel, i.e., $g_\beta(x) = \frac{1}{n}\sum_{i=1}^n \prod_{j=1}^d [\frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2}) \exp(\beta_0^j + \beta_1^j (x^j)^2)]$. Next, we relax the constraint on $\beta_0^j$s, and let them depend on $x_i^j$ in such a way that

$$\int_{x^j} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2}) \exp(\beta_{0i}^j + \beta_1^j (x^j)^2) dx^j = 1 \tag{3.7}$$

where $\beta_{0i}^j$ implies the dependence of $\beta_0^j$ on $x_i^j$. In this way, the marginal distribution of the $j^{\text{th}}$ feature is

$$\begin{aligned}g_\beta^j(x^j) &= \int_{x^1} \cdots \int_{x^{j-1}} \int_{x^{j+1}} \cdots \int_{x^d} g_\beta(x) dx^d \cdots dx^{j+1} dx^{j-1} \cdots dx^1 \\ &= \frac{1}{n}\sum_{i=1}^n \{ \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2}) \exp(\beta_{0i}^j + \beta_1^j (x^j)^2) \cdot \\ &\prod_{k \neq j} \int_{x^k} \frac{1}{\sqrt{2\pi}\sigma^k} \exp(-\frac{(x^k - x_i^k)^2}{2(\sigma^k)^2}) \exp(\beta_{0i}^k + \beta_1^k (x^k)^2) dx^k \} \\ &= \frac{1}{n}\sum_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2}) \exp(\beta_{0i}^j + \beta_1^j (x^j)^2)\end{aligned}$$

[6] Note that the following analysis also applies to other forms of the sufficient statistics, such as $t(x) = [x^1, \ldots, x^d]^T$. In all our experiments, the second order sufficient statistics perform the best. So we use this form in *SEDER*.

To estimate the parameters in our current model, we have the following theorem.

**Theorem 4.** *The maximum likelihood estimates $\hat{\beta}_1^j$ and $\hat{\beta}_{0i}^j$ of $\beta_1^j$ and $\beta_{0i}^j$ satisfy the following conditions:* $\forall j \in \{1, \ldots, d\}$

$$\sum_{k=1}^{n} (x_k^j)^2 = \sum_{k=1}^{n} \frac{\sum_{i=1}^{n} \exp(\hat{\beta}_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2}) E_i^j((x^j)^2)}{\sum_{i=1}^{n} \exp(\hat{\beta}_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})} \qquad (3.8)$$

*where* $E_i^j((x^j)^2) = \int_{x^j} (x^j)^2 \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2}) \exp(\hat{\beta}_{0i}^j + \hat{\beta}_1^j (x^j)^2) dx^j$.

*Proof.* First of all, according to Equation 3.7, we have $\beta_{0i}^j = -\log \int_{x^j} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma)^2}) \exp(\beta_1^j (x^j)^2) dx^j$. Therefore,

$$\begin{aligned}\frac{\partial \beta_{0i}^j}{\beta_1^j} &= -\frac{\int_{x^j} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma)^2}) \exp(\beta_1^j (x^j)^2)(x^j)^2 dx^j}{\int_{x^j} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma)^2}) \exp(\beta_1^j (x^j)^2) dx^j} \\ &= -\int_{x^j} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2}) \exp(\beta_{0i}^j + \beta_1^j (x^j)^2)(x^j)^2 dx^j = -E_i^j((x^j)^2)\end{aligned}$$

Then the log-likelihood of the data on the $j^{\text{th}}$ component is

$$\begin{aligned} l(\beta_1^j) &= \sum_{k=1}^{n} \log(g_\beta^j(x_k^j)) = \sum_{k=1}^{n} \log(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2}) \cdot \exp(\beta_{0i}^j + \beta_1^j (x_k^j)^2)) \\ &= \sum_{k=1}^{n} \log(\frac{1}{n\sqrt{2\pi}\sigma^j} \exp(\beta_1^j (x_k^j)^2) \cdot \sum_{i=1}^{n} \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2}) \exp(\beta_{0i}^j))\end{aligned}$$

Taking the partial derivative of $l(\beta_1^j)$ with respect to $\beta_1^j$, we have:

$$\begin{aligned}\frac{\partial l(\beta_1^j)}{\partial \beta_1^j} &= \sum_{k=1}^{n} (x_k^j)^2 + \sum_{k=1}^{n} \frac{\sum_{i=1}^{n} \exp(\beta_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2}) \frac{\partial \beta_{0i}^j}{\partial \beta_1^j}}{\sum_{i=1}^{n} \exp(\beta_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})} \\ &= \sum_{k=1}^{n} (x_k^j)^2 - \sum_{k=1}^{n} \frac{\sum_{i=1}^{n} \exp(\beta_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2}) E_i^j((x^j)^2)}{\sum_{i=1}^{n} \exp(\beta_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})}\end{aligned}$$

Setting the partial derivative to 0, we have that the maximum likelihood estimate $\hat{\beta}_1^j$ and $\hat{\beta}_{0i}^j$ of $\beta_1^j$ and $\beta_{0i}^j$ satisfy $\sum_{k=1}^{n} (x_k^j)^2 = \sum_{k=1}^{n} \frac{\sum_{i=1}^{n} \exp(\hat{\beta}_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2}) E_i^j((x^j)^2)}{\sum_{i=1}^{n} \exp(\hat{\beta}_{0i}^j - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})}$. □

Notice that according to Theorem 4, $\beta_1^j$s can be estimated separately, which greatly simplifies our problem. At the first glance, Equation 3.8 is hard to solve. Next, we let $\beta_1^j = (1 - \frac{1}{b^j}) \frac{1}{2(\sigma^j)^2}$, where $b^j \neq 1$ is a positive parameter, the introduction of which will simplify this equation. According to Equation 3.7, $\beta_{0i}^j$ can be expressed in terms of $b^j$, i.e.,

$$\begin{aligned}\beta_{0i}^j &= -\log \int_{x^j} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j - x_i^j)^2}{2(\sigma^j)^2} + \beta_1^j (x^j)^2) dx^j \\ &= -\log \int_{x^j} \frac{1}{\sqrt{2\pi}\sigma^j} \exp(-\frac{(x^j)^2 + b^j (x_i^j)^2 - 2b^j x^j x_i^j}{2(\sigma^j)^2 b^j}) dx^j = \frac{(1-b^j)(x_i^j)^2}{2(\sigma^j)^2} - \frac{1}{2}\log b^j\end{aligned}$$

Therefore, the estimated density becomes

$$\tilde{g}_b(x) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d} \frac{1}{\sqrt{2\pi b^j}\sigma^j} \exp(-\frac{(x^j - b^j x_i^j)^2}{2(\sigma^j)^2 b^j}) \tag{3.9}$$

Replacing $\hat{\beta}_1^j$ and $\hat{\beta}_{0i}^j$ with functions of $\hat{b}^j$ (the maximum likelihood estimate of $b^j$) in the definition of $E_i^j((x^j)^2)$, we have $E_i^j((x^j)^2) = \hat{b}^j(\sigma^j)^2 + (\hat{b}^j)^2 (x_i^j)^2$, and Equation 3.8 becomes

$$\sum_{k=1}^{n} (x_k^j)^2 = n\hat{b}^j (\sigma^j)^2 + (\hat{b}^j)^2 \sum_{k=1}^{n} \frac{\sum_{i=1}^{n} \exp(\frac{(1-\hat{b}^j)(x_i^j)^2}{2(\sigma^j)^2} - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})(x_i^j)^2}{\sum_{i=1}^{n} \exp(\frac{(1-\hat{b}^j)(x_i^j)^2}{2(\sigma^j)^2} - \frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})}$$

In general, the value of $\hat{\beta}_1^j$ is very close to 0, and $g_{\hat{\beta}}(x)$ is a smoothed version of $g_0(x)$. Therefore, $\hat{b}^j$ should be close to 1, and we can re-write the above equation as follows.

$$\frac{1}{n}\sum_{k=1}^{n} (x_k^j)^2 \approx \hat{b}^j (\sigma^j)^2 + (\hat{b}^j)^2 \frac{1}{n} \sum_{k=1}^{n} \frac{\sum_i \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})(x_i^j)^2}{\sum_i \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})}$$

This is a second-degree polynomial equation of $\hat{b}^j$, and the roots can be easily obtained by Vieta's theorem[7], i.e., $\forall j \in \{1, \ldots, d\}$

$$\hat{b}^j = \frac{-B + \sqrt{B^2 + 4AC}}{2A} \tag{3.10}$$

where $A = \frac{1}{n}\sum_{k=1}^{n} \frac{\sum_{i=1}^{n} \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})(x_i^j)^2}{\sum_{i=1}^{n} \exp(-\frac{(x_k^j - x_i^j)^2}{2(\sigma^j)^2})}$, $B = (\sigma^j)^2$, and $C = \frac{1}{n}\sum_{k=1}^{n} (x_k^j)^2$.

**Theorem 5.** *Let $g^j(x^j)$ be the true density for the $j^{th}$ feature. If $\frac{1}{n}\sum_{i=1}^{n} \frac{x_i^j}{g^j(x_i^j)} \cdot \frac{dg^j(x_i^j)}{dx_i^j} \geq -1 + O(1)$, then $\hat{b}^j \leq 1$ and $\hat{\beta}_1^j \leq 0$.*

*Proof.* For the sake of simplicity, let $z = x^j$, $h = \sigma^j$, and $f(z) = g^j(x^j)$. Then $A = \frac{1}{n}\sum_{k=1}^{n} \frac{\sum_{i=1}^{n} \exp(-\frac{(z_k - z_i)^2}{2h^2})(z_i)^2}{\sum_{i=1}^{n} \exp(-\frac{(z_k - z_i)^2}{2h^2})}$, $B = h^2$, and $C = \frac{1}{n}\sum_{k=1}^{n} (z_k)^2$. Consider the following regression problem where the true regression function $r(z) = z^2$, the noise has mean 0, and we use kernel regression to estimate this function. Then $A - C$ is the bias of kernel regression on the training data, i.e., $A - C = \frac{1}{n}\sum_{i=1}^{n} h^2(\frac{1}{2}r''(z_i) + \frac{r'(z_i)f'(z_i)}{f(z_i)}) \int z^2 k(z)dz + O(h^2)$ [Wasserman, 2005], where $k(z)$ is the Gaussian kernel used in kernel regression, i.e., $k(z) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{z^2}{2})$. Therefore, $A - C = h^2 + \frac{h^2}{n}\sum_{i=1}^{n} \frac{2z_i f'(z_i)}{f(z_i)} + O(h^2)$, and

[7] Note that the other root $\frac{-B - \sqrt{B^2+4AC}}{2A}$ is disregarded since it is negative.

$A+B-C \geq 0$ if and only if $\frac{1}{n}\sum_{i=1}^{n}\frac{z_i f'(z_i)}{f(z_i)} \geq -1+O(1)$. Given that $A+B-C \geq 0$, we can show that $\hat{b}^j = \frac{-B+\sqrt{B^2+4AC}}{2A} \leq \frac{-B+\sqrt{B^2+4A(A+B)}}{2A} = 1$, and $\hat{\beta}_1^j = (1-\frac{1}{\hat{b}^j})\frac{1}{2(\sigma^j)^2} \leq 0$. □

In Section 1.3, we have made the following assumptions: 1) the distribution of the majority classes is sufficiently smooth; and 2) the minority classes form compact clusters in the feature space. In this case, the first order derivative of the density would be close to 0 for most examples, and have large absolute values for a few examples near the rare classes. Therefore, the condition in Theorem 5 is always satisfied, and the exponential term appended to the carrier density decreases away from the origin.

### Scoring Function

Once we have estimated all the parameters using Equation 3.10, we can measure the change in the local density at each data point based on the estimated density in Equation 3.9. Note that at each data point, if we pick a different direction, the change in local density would be different. In *SEDER*, we measure the change along the gradient, which gives the maximum change at each data point.

**Theorem 6.** *Using the estimated density in Equation 3.9, $\forall x \in \mathbb{R}^d$, the maximum change rate of the density at $x$ is $\sqrt{\sum_{l=1}^{d}\frac{(\sum_{i=1}^{n}D_i(x)(x^l-b^l x_i^l))^2}{((\sigma^l)^2 b^l)^2}}$, where $D_i(x)=\frac{1}{n}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi b^j}\sigma^j}\exp(-\frac{(x^j-b^j x_i^j)^2}{2(\sigma^j)^2 b^j})$ is the contribution of $x_i$ to the estimated density at $x$.*

*Proof.* $\forall x \in \mathbb{R}^d$, let the gradient vector be $w \in \mathbb{R}^d$. We have $\forall l \in \{1,\ldots,d\}$

$$w_l = \frac{\partial \tilde{g}_b(x)}{\partial x^l} = \frac{1}{n}\sum_{i=1}^{n}(-\frac{x^l-b^l x_i^l}{(\sigma^l)^2 b^l})\prod_{j=1}^{d}\frac{\exp(-\frac{(x^j-b^j x_i^j)^2}{2(\sigma^j)^2 b^j})}{\sqrt{2\pi b^j}\sigma^j} = -\sum_{i=1}^{n}\frac{D_i(x)(x^l-b^l x_i^l)}{(\sigma^l)^2 b^l}$$

where $w_l$ is the $l^{\text{th}}$ component of $w$.

Therefore, the maximum change rate of the density at $x$ is

$$\|w\|_2 = \sqrt{\sum_{l=1}^{d}(-\sum_{i=1}^{n}\frac{D_i(x)(x^l-b^l x_i^l)}{(\sigma^l)^2 b^l})^2} = \sqrt{\sum_{l=1}^{d}\frac{(\sum_{i=1}^{n}D_i(x)(x^l-b^l x_i^l))^2}{((\sigma^l)^2 b^l)^2}}$$

□

If the distribution of the majority classes is sufficiently smooth, and the minority classes form compact clusters in the feature space, the minority classes are always located in the regions where the density changes the most. Therefore, in *SEDER*, to discover the rare classes, we set the score of each example to be the maximum change rate of the density at this example, i.e., $\forall k \in \{1,\ldots,n\}$

$$s_k = \sqrt{\sum_{l=1}^{d}\frac{(\sum_{i=1}^{n}D_i(x_k)(x_k^l-b^l x_i^l))^2}{((\sigma^l)^2 b^l)^2}} \tag{3.11}$$

where $s_k$ is the score of $x_k$. We pick the examples with the largest scores for labeling until we find at least one example from each class.

### 3.2.2 Algorithm

The intuition of *SEDER* is to select the examples with the maximum change in the local density for labeling by the oracle. As introduced in Subsubsection 3.2.1, the scores of the examples measure the maximum change rate in the local density, and they do not take into account the fact that nearby examples tend to have the same class label. Therefore, if we ask the oracle to label all the examples with large scores, we may repeatedly select examples from the most distinctive rare class, rather than discovering all the rare classes. To address this problem in *SEDER*, we make use of the following heuristic: if $x_i \in S$ has been labeled, $\forall x_k \in S, x_k \neq x_i$, if $\forall j \in \{1, \ldots, d\}$, $|x_i^j - x_k^j| \leq 3\sigma^j$, we would preclude $x_k$ from being selected. In other words, if an unlabeled example is very close to a previously labeled one, it is quite likely that the labels of the two examples are the same, and labeling that example will not have a high probability of detecting a new rare class. The size of the neighborhood is set to $3\sigma^j$ such that the estimated density for the examples outside this neighborhood using Gaussian kernel is hardly affected by the labeled example. It should be pointed out that the feedback strategy is orthogonal to the remaining parts of the proposed algorithm. In our experiments, we find that despite its simplicity, the current strategy leads to satisfactory performance.

**Algorithm 4** SEmiparametric Density Estimation based Rare category detection (*SEDER*)

```
Input: Unlabeled data set $S$
Output: The set $I$ of selected examples and the set $L$ of their labels
 1: Initialize $I = \phi$ and $L = \phi$.
 2: for $j = 1 : d$ do
 3:     Calculate the bandwidth $\sigma^j$ using cross validation [Scott, 1992].
 4:     Calculate the maximum likelihood estimate $\hat{b}^j$ of the parameter $b^j$ according to Equation 3.10.
 5: end for
 6: for $i = 1 : n$ do
 7:     Calculate the score $s_i$ of the $i^{\text{th}}$ example according to Equation 3.11 using the estimated parameters.
 8: end for
 9: while the labeling budget is not exhausted do
10:     Set $S' = \{x | x \in S, \forall i \in I, \exists j \in \{1, \ldots, d\}, \text{ s.t., } |x^j - x_i^j| > 3\sigma^j\}$
11:     Query $x = \text{argmax}_{x_i \in S'} s_i$ for its label $y_x$
12:     $I = I \cup \{x\}$, $L = L \cup \{y_x\}$;
13: end while
```

The proposed method, *SEDER*, is summarized in Alg. 4. It works as follows. Firstly, we initialize the set $I$ of selected examples and the set $L$ of their labels to empty sets. Then step 2 to step 5 calculate the parameters in our model. Step 6 to step 8 calculate the score for each example in $S$. Finally, step 9 to step 13 gradually include the example with the maximum score into $I$ and its label into $L$ until we run out of the labeling budget. In each round, the selected example should be far away from all the labeled examples.

Note that: 1) unlike the methods proposed in Section 3.1, *SEDER* does not need to be given the number of classes in $S$ or any other information, hence it is more suitable for real applications; 2) in *SEDER*, we do not need to explicitly calculate the density at each example; 3) *SEDER* does not depend on the assumption that the majority and minority classes be separable or near-separable.

### 3.2.3 Experimental Results

In this subsection, we compare *SEDER* with *MALICE* proposed in Section 3.1, Interleave (the best method proposed in [Pelleg & Moore, 2004]), random sampling (RS) and *SEDER* with $b^j = 1$ for $j = 1, \ldots, d$ (abbreviated as Kernel, which is equivalent to using kernel density estimator to estimate the density and to get the scores) on both synthetic and real data sets. For this purpose, we run these methods until all the classes have been discovered, and compare the number of label requests by each method in order to find a certain number of classes. Note that *SEDER* *MALICE* and Kernel are deterministic, whereas the results for Interleave and random sampling are averaged over 100 runs.

Here we would like to emphasize that only *SEDER* RS and Kernel do not need any prior information about the data set, whereas *MALICE* and Interleave need extra information about the data set as inputs, such as the number of classes and the proportions of different classes. When such prior information is not available, which is quite common in real applications, *MALICE* and Interleave are not applicable.

### Synthetic Data Sets

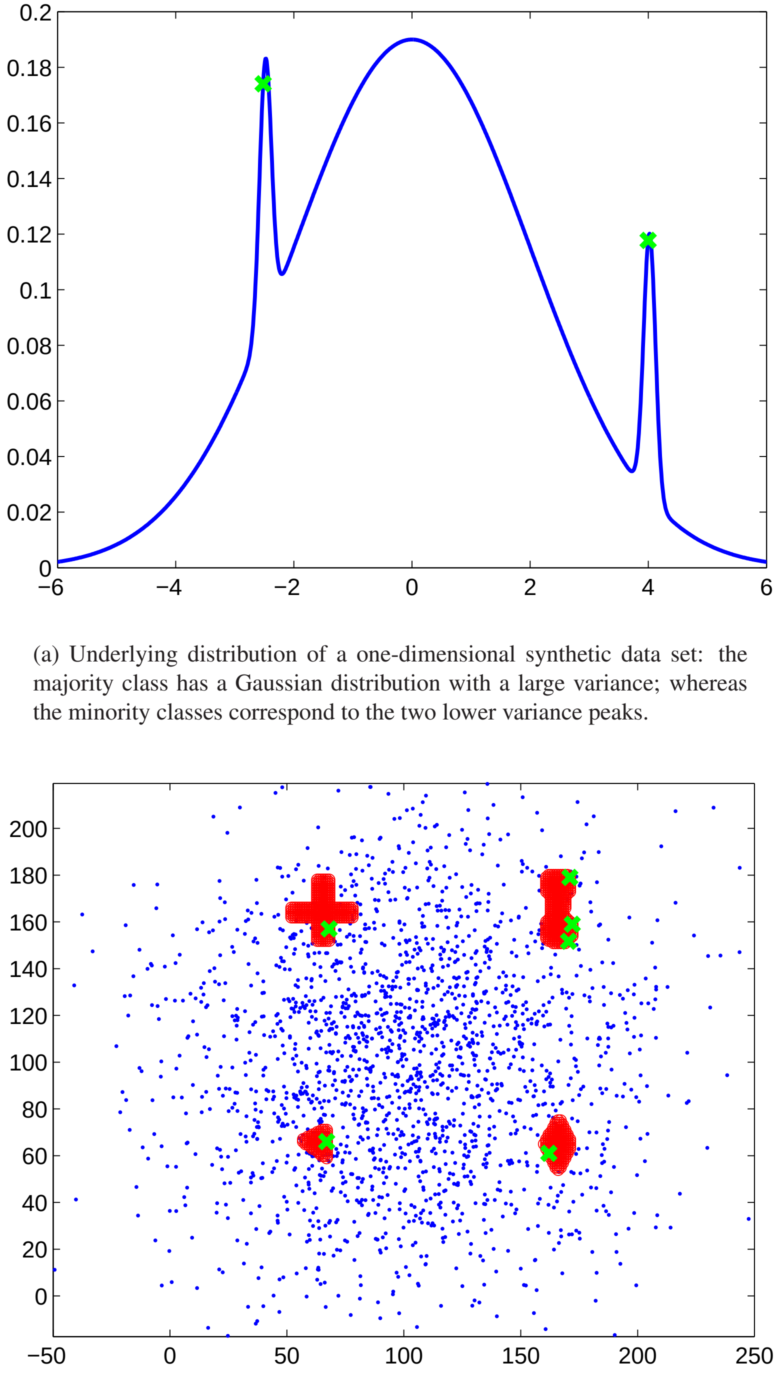
Fig. 3.7(a) shows the underlying distribution of a 1-dimensional synthetic data set. The majority class with 2000 examples has a Gaussian distribution with a large variance; whereas the minority classes with 50 examples each correspond to the two lower-variance peaks. As can be seen from this figure, the first two examples selected by *SEDER* (red stars) are both from the regions where the density changes the most.

Fig. 3.7(b) shows a 2-dimensional synthetic data set. The majority class has 2000 examples (blue dots) with a Gaussian distribution. The four minority classes (red balls) all have different shapes, and each has 267, 280, 84 and 150 examples respectively. This data set is similar to the one used in [He & Carbonell, 2008]. To discover all the classes, *SEDER* only needs to label 6 examples, which are represented by green 'x's in the figure; whereas random sampling needs to label more than 50 examples on average.

### Real Data Sets

In this subsubsection, we present the experimental results on some real data sets. The properties of the data sets are summarized in Table 3.6. Notice that all these data sets are skewed: the proportion of the smallest class is less than 5%. For the last three data sets (Page Blocks, Abalone and Shuttle), it is even less than 1%. We refer to these three data sets as 'highly' skewed; whereas the remaining two data sets (Ecoli and Glass) are referred to as 'moderately' skewed.

First, let us focus on the 'moderately' skewed data sets, which are shown in Fig. 3.8 and Fig. 3.9. With Ecoli data set, to discover all the classes, *MALICE* needs 36 label requests, Interleave needs 41 label requests on average, RS needs 43 label requests on average, Kernel needs 78 label requests, and *SEDER* only needs 20 label requests; with Glass data set, to discover all the classes, *MALICE* needs 18 label requests, Interleave needs 24 label requests on average, RS needs 31 label requests on average, Kernel needs 102 label requests, and *SEDER* needs 22 label requests. Therefore, on these 'moderately' skewed data sets, the performance of *SEDER* is better than or comparable with *MALICE*, which requires more prior information than *SEDER* including the number of classes in the data set and the proportions of different classes.



(a) Underlying distribution of a one-dimensional synthetic data set: the majority class has a Gaussian distribution with a large variance; whereas the minority classes correspond to the two lower variance peaks.

(b) Two-dimensional synthetic data set: there are 2000 examples from the majority class, denoted as blue dots; there are 267, 280, 84, and 150 examples from each minority class, denoted as red balls.

Figure 3.7: Synthetic data sets: examples selected by *SEDER* are denoted as green ‘x’s.

Next, let us look at the ‘highly’ skewed data sets. For example, in Shuttle data set, the largest class has

580 times more examples than the smallest class. With Page Blocks data set (Fig. 3.10), to discover all the classes, *SEDER* needs 36 label requests, *MALICE* needs 23 label requests, Interleave needs 77 label requests on average, RS needs 199 label requests on average, and Kernel needs more than 1000 label requests; with Abalone data set (Fig. 3.11), to discover all the classes, *SEDER* needs 316 label requests, *MALICE* needs 179 label requests, Interleave needs 333 label requests on average, RS needs 483 label requests on average[8], and Kernel needs more than 1000 label requests; with Shuttle data set (Fig. 3.12), to discover all the classes, *SEDER* needs 249 label requests, *MALICE* needs 87 label requests, Interleave needs 140 label requests on average, RS needs 512 label requests on average, and Kernel needs more than 1000 label requests.

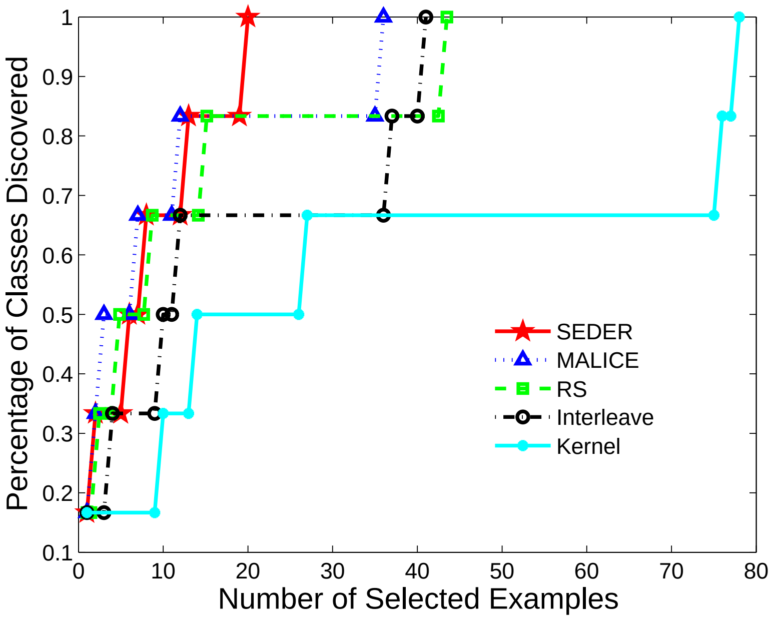


Figure 3.8: Ecoli data set.

Based on the above results, we have the following observations. First, *SEDER*, RS and Kernel require no prior information about the data set, and yet *SEDER* is significantly better than RS and Kernel in all the experiments. Second, the performance of Interleave is sometimes worse than *SEDER* (except Fig. 3.12), even though it is given the additional information about the number of classes in the data set. Finally, although *MALICE* is better than *SEDER* for these 'highly' skewed data sets, in real applications, it is very difficult to estimate the number of classes in the data set, not to mention the proportions of the different classes. If the information provided to *MALICE* is not accurate enough, the performance of *MALICE* may be negatively affected. Moreover, when such information is not available, *MALICE* is not applicable at all.

[8] Note that with Abalone data set, the results of *MALICE* and Interleave are slightly different from [He & Carbonell, 2007]. This is due to the effect of normalization on the data.

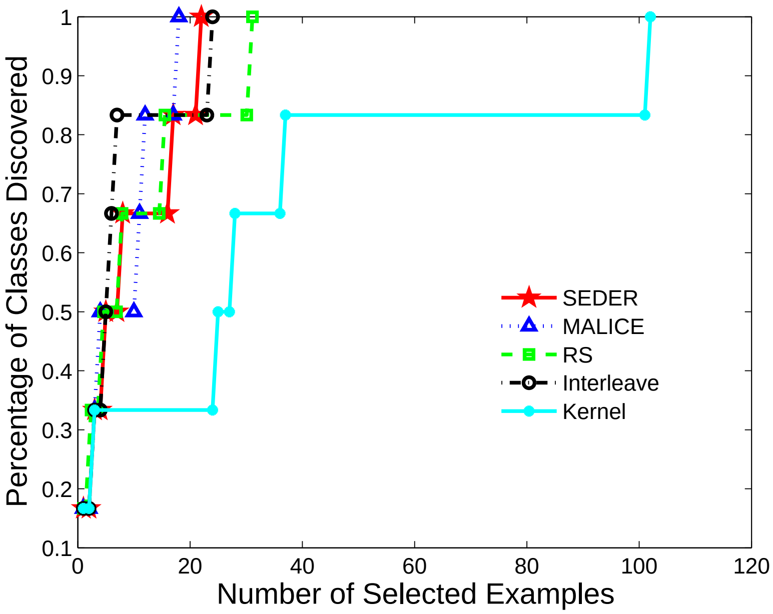


Figure 3.9: Glass data set.

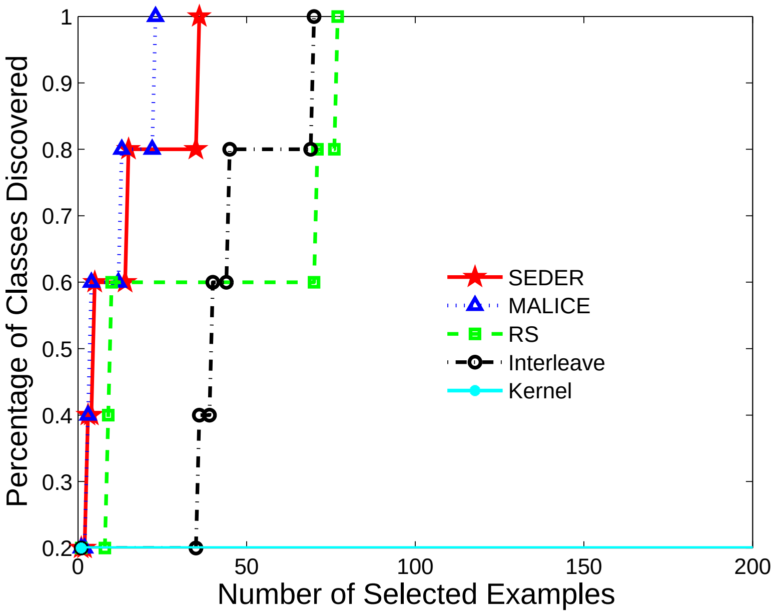


Figure 3.10: Page Blocks data set.

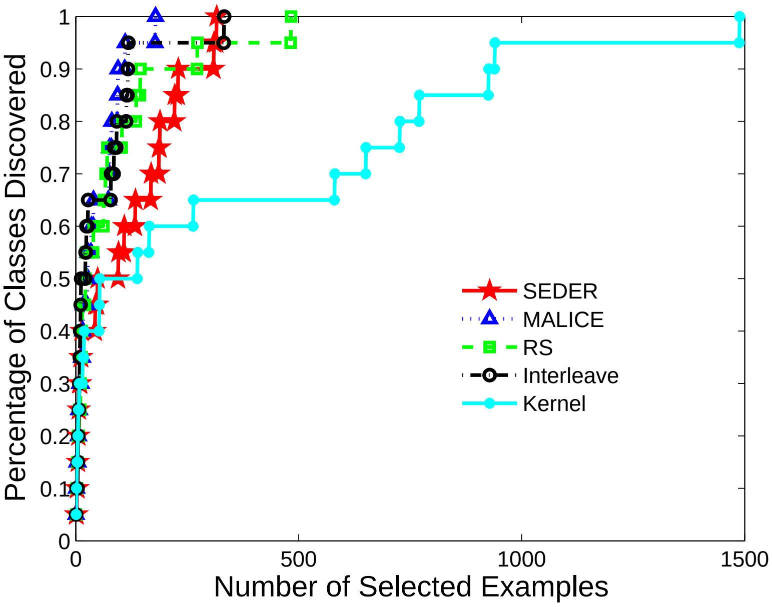


Figure 3.11: Abalone data set.

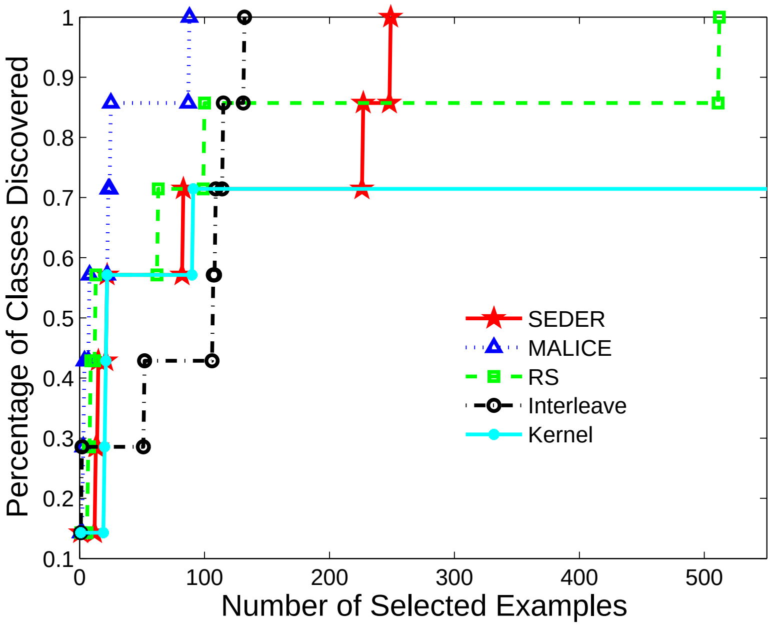


Figure 3.12: Shuttle data set.

Table 3.6: Properties of the data sets used

| Data Set | $n$ | $d$ | $m$ | Largest Class | Smallest Class |
|---|---|---|---|---|---|
| Ecoli [Asuncion & Newman, 2007] | 336 | 7 | 6 | 42.56% | 2.68% |
| Glass [Asuncion & Newman, 2007] | 214 | 9 | 6 | 35.51% | 4.21% |
| Page Blocks [Asuncion & Newman, 2007] | 5473 | 10 | 5 | 89.77% | 0.51% |
| Abalone [Asuncion & Newman, 2007] | 4177 | 7 | 20 | 16.50% | 0.34% |
| Shuttle [Asuncion & Newman, 2007] | 4515 | 9 | 7 | 75.53% | 0.13% |

## 3.3 Rare Category Detection for Graph Data

In the previous two sections, we have discussed rare category detection algorithms for data with feature representations. In many real applications, sometimes the data is given to us as a graph, such as transaction networks and social networks, and we hope to discover minority classes on the graph data, such as collusion-type of fraud transactions and groups of terrorists who constantly interact with each other. To address this problem, in this section, we propose a new rare category detection method for graph data, or relational data: Graph-based Rare Category Detection (*GRADE*). The basic idea is to utilize the global similarity matrix and get more compact clusters for the examples from the minority classes, which is motivated by the manifold ranking algorithm [Zhou *et al.*, 2003b] and the consistency method [Zhou *et al.*, 2003a]. This results in sharp changes in local density near the boundary of the minority classes and thus makes it easier to discover those classes. Furthermore, we improve the *GRADE* algorithm to get the *GRADE-LI* algorithm, which requires less prior information compared with the *GRADE* algorithm, and thus is more suitable for real applications. Notice that our algorithms can deal with both data with feature representations and graph data, whereas existing rare category detection methods can only work with data with feature representations.

The rest of this section is organized as follows. In Subsection 3.3.1, we describe the *GRADE* algorithm for rare category detection and analyze its effectiveness. The improved algorithm *GRADE-LI* is presented in Subsection 3.3.2. In Subsection 3.3.3, we show some experimental results of *GRADE* and *GRADE-LI* compared with existing methods. Finally, we discuss about some implementation issues in Subsection 3.3.4.

### 3.3.1 *GRADE* Algorithm

In this subsection, we first introduce the *GRADE* algorithm for data with feature representation. Then we discuss its application to graph data.

#### Algorithm

The *GRADE* algorithm for examples with observed features is described in Alg. 5.

**Algorithm 5** Graph-based Rare Category Detection (*GRADE*)

```
Input: Unlabeled data set $S$, $p_1, \ldots, p_m$, $\alpha$
Output: The set $I$ of selected examples and the set $L$ of their labels
1:  Let $K = \max_{c=2}^{m} n \times p_c$.
2:  For each example, calculate the distance between this example and its $K^{\text{th}}$ nearest neighbor. Set $\sigma$ to be
    the minimum value of all such distances.
3:  Construct the pair-wise similarity matrix $W'$, $n \times n$, where $n$ is the number of examples, and $\forall i, k =
    1, \ldots, n$,
        $$W'_{ik} = \exp(-\frac{\|x_i - x_k\|^2}{2\sigma^2})\mathrm{I}(i \neq k) \qquad (3.12)$$
    where $\mathrm{I}(\cdot)$ is the indicator function.
4:  Construct the diagonal matrix $D$, $n \times n$, where $D_{ii} = \sum_{k=1}^{n} W'_{ik}$, $i = 1, \ldots, n$.
5:  Calculate the normalized matrix $W = D^{-1/2} W' D^{-1/2}$.
6:  Calculate the global similarity matrix $A = (I_{n \times n} - \alpha W)^{-1}$, where $I_{n \times n}$ is an $n \times n$ identity matrix.
7:  for $c = 2 : m$ do
8:      Let $K_c = np_c$.
9:      For each row of $A$, find the $(K_c)^{\text{th}}$ largest element. Set $a_c$ to be the largest value of all such elements.
10:     $\forall x_i \in S$, let $NN(x_i, a_c) = \{x | x \in S, A(x, x_i) \geq a_c\}$, and $n_i^c = |NN(x_i, a_c)|$, where $A(x, x_i)$ is
        the corresponding element in $A$.
11: end for
12: for $c = 2 : m$ do
13:     If class $c$ has been discovered, continue.
14:     for $t = 2 : n$ do
15:         For each $x_i$ that has been labeled $y_i$, $\forall x_k \in S$, if $A(x_i, x_k) \geq a^{y_i}$, $s_k = -\infty$; for all the other
            examples, $s_i = \max_{x_k \in NN(x_i, \frac{a_c}{t})} (n_i^c - n_k^c)$.
16:         Select and query the label of $x = \arg\max_{x_i \in S} s_i$.
17:         If the label of $x$ is equal to $c$, break; otherwise, mark the class that $x$ belongs to as discovered.
18:     end for
19: end for
```

Here $\alpha$ is a positive parameter which is very close to 1. It works as follows. First of all, we calculate the maximum number of examples $K$ in each minority class. Then using this number, we pick the parameter $\sigma$, which is the smallest distance to the $K^{\text{th}}$ nearest neighbor. Next, we construct the pair-wise similarity matrix $W'$, whose elements are calculated using the Gaussian kernel. In Step 4, we construct the diagonal matrix $D$, whose elements are the row sums of $W'$. Next, we calculate the normalized matrix $W$ and the global similarity matrix $A$. The specific form of the global similarity matrix has been used in the manifold ranking algorithm [Zhou *et al.*, 2003b] for ranking data with respect to a query point and the consistency method [Zhou *et al.*, 2003a] for semi-supervised learning. The following steps are based on the similarity measure in $A$. For each class $c$, we calculate the number of examples $K_c$ from this class, and find the largest global similarity to the $(K_c)^{\text{th}}$ nearest neighbor, which is the class specific similarity $a_c$. Then, for each example $x_i$, we find all of its neighbors with global similarity bigger than or equal to $a_c$, which is denoted $NN(x_i, a_c)$, and let $n_i^c$ be the number of examples in this set. In Step 12 to Step 19, we calculate the score for each example and ask the oracle to label the example with the largest score. To be specific, for each class $c$, if we have not found any example from this class, we set the score of $x_i$ to be the maximum difference of $n_i^c$ and that of the neighboring points with similarity bigger than or equal to $\frac{a_c}{t}$, where $t$ is the iteration index. By querying the label of the example with the largest score, we are focusing on the regions where the underlying density changes the most. If this example is not from class $c$, we increase $t$ by 1 and repeat; otherwise, we proceed to the next class. Notice that for a labeled example, any unlabeled example with global similarity bigger than or equal to the class specific similarity will not be selected in the future.

### Justification

**Global similarity matrix** $\forall c = 1, \ldots, m$, let $S_c$ be the subset of $S$ that consists of all the examples from class $c$, which are denoted $S_c = \{x_1(c), \ldots, x_{K_c}(c)\} \subset S$. Let $W'^c$ be the associated pair-wise similarity matrix, $K_c \times K_c$, whose elements are defined as in Equation 3.12. Notice that by setting $\sigma$ to be the smallest distance to the $K^{\text{th}}$ nearest neighbor, we guarantee that for any example from a minority class, its pair-wise similarity with at least another example from the same minority class is reasonably large. Next, define the diagonal matrix $D^c$, $K_c \times K_c$, where $D^c_{ii} = \sum_{k=1}^{K_c} W'^c_{ik}$. Finally define the normalized matrix $W^c = (D^c)^{-1/2} W'^c (D^c)^{-1/2}$. Notice that $W^c$ is positive semi-definite, so its eigen-values are non-negative, which are denoted $\lambda^c_1 \geq \lambda^c_2 \geq \ldots \geq \lambda^c_{K_c} \geq 0$, and the corresponding eigen-vectors are denoted $u^c_1, \ldots, u^c_{K_c}$, s.t., $\|u^c_i\| = 1$, $i = 1, \ldots, K_c$. Furthermore, the largest eigen-value $\lambda^c_1$ is 1, with eigen-vector $u^c_1 \propto (D^c)^{1/2} \mathbf{1}_{K_c \times 1}$, where $\mathbf{1}_{K_c \times 1}$ is a vector of 1s. With respect to $u^c_1$, we have the following lemma.

**Lemma 3.** *If $\sigma$ changes with the number of examples $n$ such that $\lim_{n\to\infty} \sigma = 0$ and $\lim_{n\to\infty} n(\sigma)^d = \infty$, then as $n$ goes to infinity, $\forall c = 1, \ldots, m$, $u^c_1 \cdot (u^c_1)^T$ converges in probability to $C_c U^c$, where $C_c$ is a constant, $U^c$ is a $K_c \times K_c$ matrix, its elements at the $i^{th}$ row and $k^{th}$ column $U^c_{ik} = \sqrt{f_c(x_i(c)) \times f_c(x_k(c))}$, and $f_c(x_i(c))$ is the probability density function of class $c$ at $x_i(c)$.*

*Proof.* As we have mentioned before, the $i^{\text{th}}$ element of $u^c_1$: $u^c_1(i) \propto \sqrt{D^c_{ii}}$. On the other hand, as $n$ goes to infinity, $K_c = np_c$ goes to infinity, and $\frac{D^c_{ii}}{(K_c-1)(\sqrt{2\pi}\sigma)^d} = \frac{\sum_{k=1,k\neq i}^{K_c} \exp(-\frac{\|x_i(c)-x_k(c)\|^2}{2\sigma^2})}{(K_c-1)(\sqrt{2\pi}\sigma)^d}$ converges in probability to $E(\delta(x_i(c), x)) = \int \delta(x_i(c), x) f_c(x) dx = f_c(x_i(c))$, where $\delta(x_i(c), x)$ is a delta function at $x = x_i(c)$. This is based on both the law of large numbers and the fact that as $\sigma$ goes to 0, $\frac{\exp(-\frac{\|x_i(c)-x\|^2}{2\sigma^2})}{(\sqrt{2\pi}\sigma)^d}$ converges to $\delta(x_i(c), x)$. Therefore, as $n$ goes to infinity, $u^c_1(i)$ is in proportion to $\sqrt{f_c(x_i(c))}$, and $u^c_1 \cdot (u^c_1)^T$ converges in probability to $C_c U^c$, where $U^c_{ik} = \sqrt{f_c(x_i(c)) f_c(x_k(c))}$. □

If we knew the class labels of all the examples in $S$, we can group the examples from the same class, and put the examples from the majority class at the end. To start with, suppose that if $x_i$ and $x_k$ are from different classes, $W_{ik} = 0$. Then the normalized matrix $W$ is block-diagonal, i.e., $W = \text{diag}(W^2, \ldots, W^{m-1}, W^1)$. Therefore $A = (I_{n\times n} - \alpha W)^{-1}$ is also block-diagonal, and it satisfies the following lemma.

**Lemma 4.** *If $x_i$ and $x_k$ both belong to class $c$, $A_{ik} = \sum_{l=1}^{K_c} \frac{1}{1-\alpha\lambda^c_l} u^c_l(i) u^c_l(k)$; otherwise, $A_{ik} = 0$.*

*Proof.* It is easy to see that the eigen-values of $A$ are $\frac{1}{1-\alpha\lambda^c_i}$, $c = 1, \ldots, m$, $i = 1, \ldots, K_c$, and the eigen-vectors have the same value as $u^c_i$ if the corresponding example is from class $c$, and 0 otherwise. Reconstructing $A$ based on eigen-decomposition, we get the above lemma. □

If $\alpha$ is very close to 1, $A$ can be approximated as follows. If $x_i$ and $x_k$ both belong to class $c$, $A_{ik} \approx \frac{1}{1-\alpha} u^c_1(i) u^c_1(k)$. According to Lemma 3, as $n$ goes to infinity, $A_{ik}$ converges in probability to $\frac{C_c}{1-\alpha}\sqrt{f_c(x_i) f_c(x_k)}$.

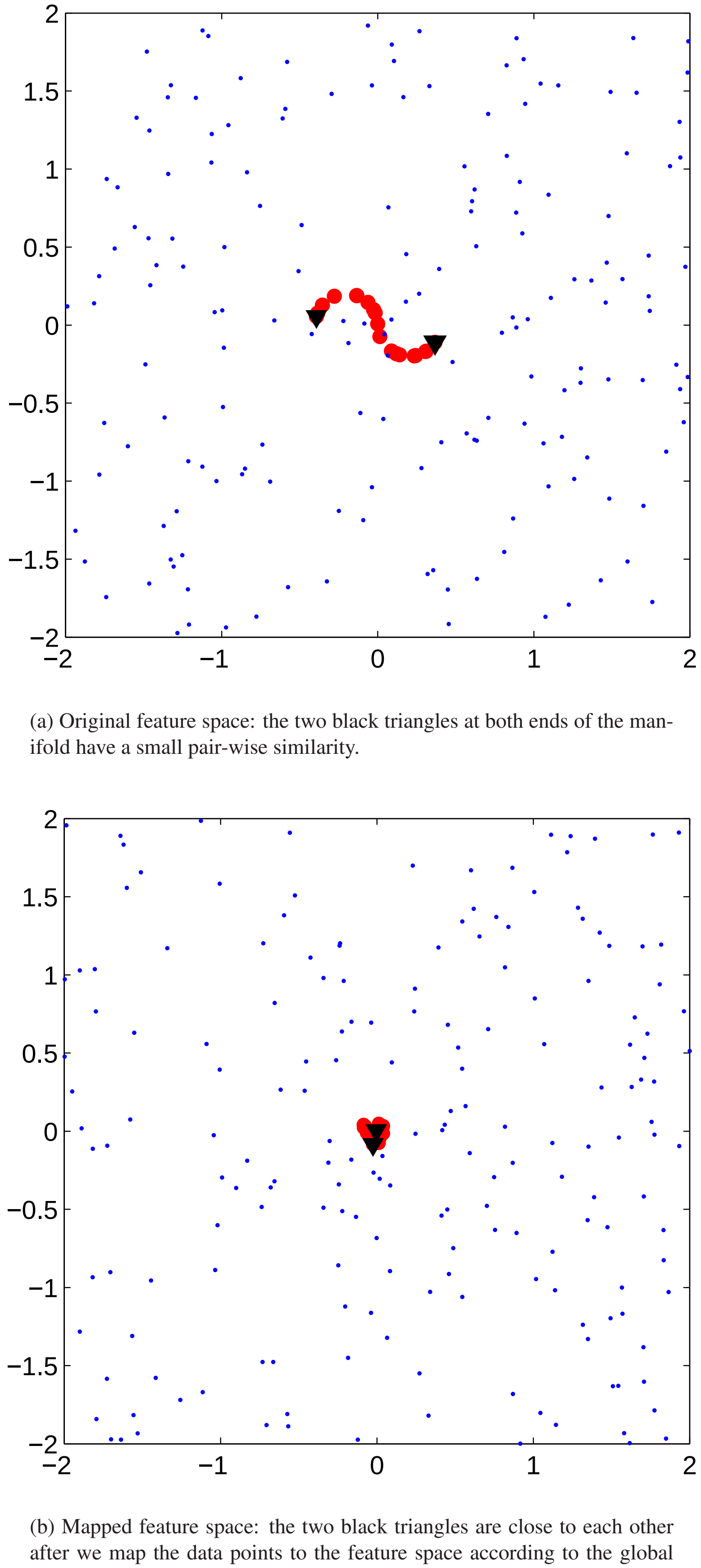
Notice that $\|u^c_i\| = 1$, $c = 1, \ldots, m$, $i = 1, \ldots, K_c$. In general, the absolute value of the elements of $u^c_1$ for the minority classes are much larger than that for the majority class since the majority class has far more examples than the minority classes. Therefore, if two examples are both from a minority class, their global similarity tends to be much larger than that if they are both from the majority class.

Compared with the pair-wise similarity matrix $W'$, the global similarity matrix $A$ is better suited for rare category detection. This is because if the minority class has a manifold structure, two examples on this manifold may be far away from each other in terms of Euclidean distance, so their pair-wise similarity is very small; whereas their global similarity is large since it is roughly in proportion to the density of the minority class at both points. Fig.3.13a shows an example where the majority class (blue dots) has a uniform distribution, and the minority class (red balls) forms a 1-dimensional manifold. The black triangles at both ends of the manifold have a small pair-wise similarity. However, in terms of the global similarity, they are quite similar, which matches our intuition. Furthermore, if we take the global similarity matrix as the pair-wise similarity matrix, and map all the points to the original feature space while preserving the pair-wise similarity, the examples from the minority classes tend to form more compact clusters compared with the original feature representation (Fig.3.13b), whereas the probability density function of the majority class is still quite smooth, which makes the following querying process more effective. This is particularly beneficial if the manifold structures of the minority classes are elongated, as shown in Fig.3.13a.

**Querying process** Step 7 to Step 19 select examples to be labeled by the oracle. According to our discussion in the last subsubsection, if we reconstruct the features according to the global similarity matrix, the minority classes will form compact clusters and the probability density function of the majority class will be locally smooth. Generally speaking, the querying process selects the examples from the regions where the local density changes the most, which have a high probability of coming from the minority classes. To be specific, as discussed before, if two examples are both from a minority class, their global similarity tends to be much larger than that if they are both from the majority class. So the class specific similarity $a_c$ is likely to be determined by the examples from the minority classes. Furthermore, as $n$ goes to infinity, $A_{ik}$ is roughly in proportion to the density of class $c$ at $x_i$ and $x_k$ if they both belong to minority class $c$. Therefore, if $f_c(x_i)$ is large, the global similarity between $x_i$ and the other examples from class $c$ tends to be large, and $n_i^c$ is large accordingly. In other words, if we take the global similarity as the pair-wise similarity based on the new feature representation, $n_i^c$ is the number of neighbors within a fixed distance. Therefore, $n_i^c$ is roughly in proportion to the local density at $x_i$.

For each class $c$, we calculate the score of each example $x_i$, which is the maximum difference in the local density between $x_i$ and the other examples with global similarity bigger than or equal to $\frac{a_c}{t}$. By querying the data point with the largest score, we focus on the regions where the local density changes the most, so we have a high probability of finding examples from the minority classes. Furthermore, by increasing the value of $t$, we gradually enlarge the size of the neighborhood. In this way, we are not only able to select points on the boundary of the minority classes, but also points in the interior, thus increase our chance of finding the minority class examples. Finally, we make use of a simple feedback strategy: if an unlabeled example is quite similar to a labeled one, we preclude it from being selected in the future. In this way, we avoid wasting the labeling effort on the minority classes that have been discovered already. Notice that the feedback strategy is orthogonal to the other components of our algorithm. Currently, we are exploring more effective feedback strategies.

It should be mentioned that we do not make any assumption about the separability between the majority class and the minority classes, which is different from [Fine & Mansour, 2006] and [Pelleg & Moore, 2004]. In fact, our algorithm works well when the support regions of the majority class and the minority classes overlap.



(a) Original feature space: the two black triangles at both ends of the manifold have a small pair-wise similarity.

(b) Mapped feature space: the two black triangles are close to each other after we map the data points to the feature space according to the global similarity.

Figure 3.13: Synthetic data set: blue dots denote the majority class, and red balls denote the minority class.

The *NNDB* algorithm proposed in Section 3.1 can be seen as a special case of our algorithm for the binary case. If we use the pair-wise similarity matrix $W'$ instead of the global similarity matrix $A$, and the update of the neighborhood size is slightly modified in Step 15, our algorithm queries the same examples as *NNDB*. In the *NNDB* algorithm, it has been proven that under certain conditions, with a high probability, after a few iteration steps, *NNDB* queries at least one example whose probability of coming from the minority class is at least $\frac{1}{3}$. If the new feature representation based on the global similarity matrix satisfies these conditions, our algorithm shares the same theoretical properties as *NNDB*. In real applications, our algorithm is better than *NNDB* or *MALICE* (the counterpart of *NNDB* for multiple classes) since we make use of the global similarity instead of the pair-wise similarity, which makes the minority class examples more tightly clustered with the new feature representation.

**Application to Graph Data**

The proposed *GRADE* algorithm essentially makes use of the graph structure to help with rare category detection. Previously we have discussed its application on data with feature representations, which is based on the construction of the pair-wise similarity matrix using features. In this paper, we mainly focus on this type of data. On the other hand, for graph data, it can be readily applied on the input graph. To be specific, given a graph $G = (V, W')$, where $V = \{v_1, \ldots, v_n\}$ consists of all the vertices, and $W'$ is the connectivity matrix, i.e., $W'_{ik}$ is the edge weight if $v_i$ is connected with $v_k$, and $W'_{ik} = 0$ otherwise. $W'$ can be either binary or real-valued (non-negative). Notice that the elements of $W'$ denote the pair-wise similarity, which is similar to the pair-wise similarity matrix constructed in Step 3 of Alg. 5 for data with feature representation. To detect the rare categories using Alg. 5, we input the graph $G$, $p_1, \ldots, p_m$ and $\alpha$. Then, we skip Step 1 to Step 3. All the other steps are the same as before.

It is worth mentioning that in graph mining, researchers have developed algorithms for detecting dense subgraphs or communities [Flake *et al.*, 2000, Kumar *et al.*, 2003, Gibson *et al.*, 2005]. If we want to use these approaches for rare category detection, it is labor-intensive to have the oracle label the whole subgraph. Conversely, the problem of picking representative vertices of the subgraphs for the oracle to label for detection purposes has not been addressed by existing work.

## 3.3.2 *GRADE-LI* Algorithm

In the *GRADE* algorithm, we need to input the proportions of all the classes. Notice that in practise, it is often difficult to estimate the number of classes in the data set, not to mention the priors of different classes. However, it may be relatively easier to obtain an upper bound on the proportions of the minority classes of interest to us. In this subsection, we relax this requirement to produce the *GRADE-LI* algorithm, which only needs an upper bound $p$ on the proportions of all the minority classes. Compared with *GRADE*, *GRADE-LI* is more suited for real applications.

The *GRADE-LI* algorithm is summarized in Alg. 6. It works as follows. Step 1 to Step 3 construct the pair-wise similarity matrix. The only difference from the *GRADE* algorithm is that here we use the upper bound $p$ to set the value of $K$. Step 4 to Step 6 calculate the global similarity matrix, which is the same as in the *GRADE* algorithm. Step 7 calculates the largest global similarity to the $K^{\text{th}}$ nearest neighbor and assigns it to $a$. Then in Step 8, for each example $x_i$, we find the number $n_i$ of its neighbors with global similarity bigger than or equal to $a$. The while loop in Step 9 is essentially the same as in the *GRADE* algorithm except that we are using a single similarity $a$ instead of a set of class specific similarities.

If we only have one minority class, and $p = p_2$, *GRADE* and *GRADE-LI* produce the same result. If we have multiple classes, *GRADE-LI* requires less information than *GRADE*. The larger the number of minority classes, the greater the reduction in the amount of information needed as input. If the proportions of different minority classes do not vary a lot, $a$ well represents the set of class specific similarities, and the performance of *GRADE-LI* is similar to *GRADE*. Furthermore, as we will show in the next subsection, *GRADE-LI* is quite robust to small perturbations in the upper bound $p$.

**Algorithm 6** Graph-based Rare Category Detection with Less Information (*GRADE-LI*)

```
Input: Unlabeled data set $S$, $p$, $\alpha$
Output: The set $I$ of selected examples and the set $L$ of their labels
Let $K = n \times p$.
For each example, calculate the distance between this example and its $K^{\text{th}}$ nearest neighbor. Set $\sigma$ to be the minimum value of all such distances.
Construct the pair-wise similarity matrix $W'$, $n \times n$, where $n$ is the number of examples, and $\forall i, k = 1, \ldots, n$,
    $$W'_{ik} = \exp(-\frac{\|x_i - x_k\|^2}{2\sigma^2})\mathrm{I}(i \neq k)$$
Construct the diagonal matrix $D$, $n \times n$, where $D_{ii} = \sum_{k=1}^{n} W'_{ik}$, $i = 1, \ldots, n$.
Calculate the normalized matrix $W = D^{-1/2} W' D^{-1/2}$.
Calculate the global similarity matrix $A = (I_{n \times n} - \alpha W)^{-1}$.
For each row of $A$, find the $K^{\text{th}}$ largest element. Set $a$ to be the largest value of all such elements.
$\forall x_i \in S$, let $NN(x_i, a) = \{x | x \in S, A(x, x_i) \geq a\}$, and $n_i = |NN(x_i, a)|$.
while not all the classes have been discovered do
  for $t = 2 : n$ do
    For each $x_i$ that has been labeled $y_i$, $\forall x_k \in S$, if $A(x_i, x_k) \geq a$, $s_k = -\infty$; for all the other examples, $s_i = \max_{x_k \in NN(x_i, \frac{a}{t})} (n_i - n_k)$.
    Select and query the label of $x = \arg\max_{x_i \in S} s_i$.
    Mark the class that $x$ belongs to as discovered.
  end for
end while
```

### 3.3.3 Experimental Results

In this subsection, we present the experimental results on both synthetic and real data sets to show the effectiveness of *GRADE* and *GRADE-LI*.

#### Synthetic Data Sets

Fig. 3.14 shows the result of applying *GRADE* on the synthetic data set in Fig. 3.13a. There are 1000 examples from the majority class, and only 20 examples from the minority class. Using random sampling, we need to label 51 examples to discover the minority class on average, whereas using the *GRADE* algorithm, we only need to label 1 example, denoted as the green ‘x’. Note that in this data set, we only have one minority class. If we run *GRADE-LI* with the prior of the minority class as input, we get exactly the same result as *GRADE*.

Fig. 3.15 shows another synthetic data set. It has one majority class (blue dots) and 4 minority classes (red balls), which form the 4 characters. The majority class has 1000 examples, whereas each minority class has 100 examples. In order to find all the minority classes, using random sampling, we need to label 29 examples on average, whereas using the *GRADE* algorithm, we only need to label 4 examples, one from each minority class, which are denoted as green ‘x’s. Similar as *GRADE*, if we apply *GRADE-LI* on this data set with the exact upper bound, i.e., $p = \max_{c=2}^{m} p_c$, we also need 4 label requests to find all the minority classes.

Based on the synthetic data set in Fig. 3.15, we gradually reduce the number of examples that form the character ‘I’, and compare the total number of label requests needed by *GRADE* and *GRADE-LI* in Fig. 3.16. As the number of examples from ‘I’ decreases, the proportions of different minority classes become more skewed, and the difference between *GRADE* and *GRADE-LI* becomes significant. This matches our intuition since *GRADE-LI* only has access to an upper bound on the proportions of all the minority classes. Therefore, if the proportions of different minority classes are quite different, it may not be very good at discovering the smallest class.

### Real Data Sets

In this subsubsection, we perform experiments on 4 real data sets, which are summarized in Table 3.7. Note that we have pre-processed the data so that each feature component has mean 0 and standard deviation 1. In the following experiments, we have compared *GRADE* and *GRADE-LI* with the following methods: *MALICE*, Interleave (the best method proposed in [Pelleg & Moore, 2004]) and random sampling (RS). Notice that the results for Interleave and RS are averaged over 100 runs.

Table 3.7: Properties of the data sets used.

| Data Set | $n$ | $d$ | $m$ | Largest Class | Smallest Class |
|---|---|---|---|---|---|
| Ecoli [Asuncion & Newman, 2007] | 336 | 7 | 6 | 42.56% | 2.68% |
| Glass [Asuncion & Newman, 2007] | 214 | 9 | 6 | 35.51% | 4.21% |
| Abalone [Asuncion & Newman, 2007] | 4177 | 7 | 20 | 16.50% | 0.34% |
| Shuttle [Asuncion & Newman, 2007] | 4515 | 9 | 7 | 75.53% | 0.13% |

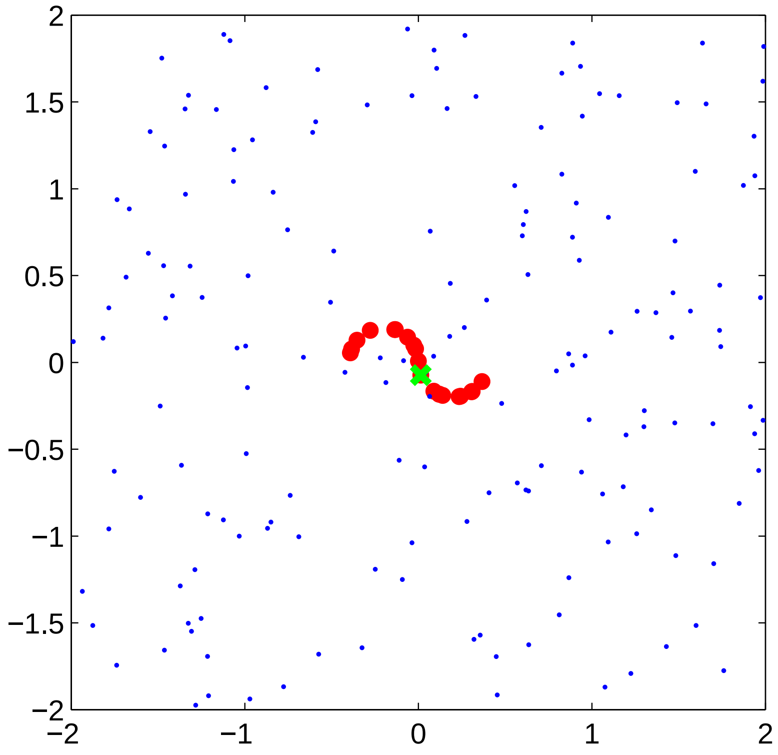


Figure 3.14: Synthetic data set: the example selected by *GRADE* is denoted as green ‘x’.

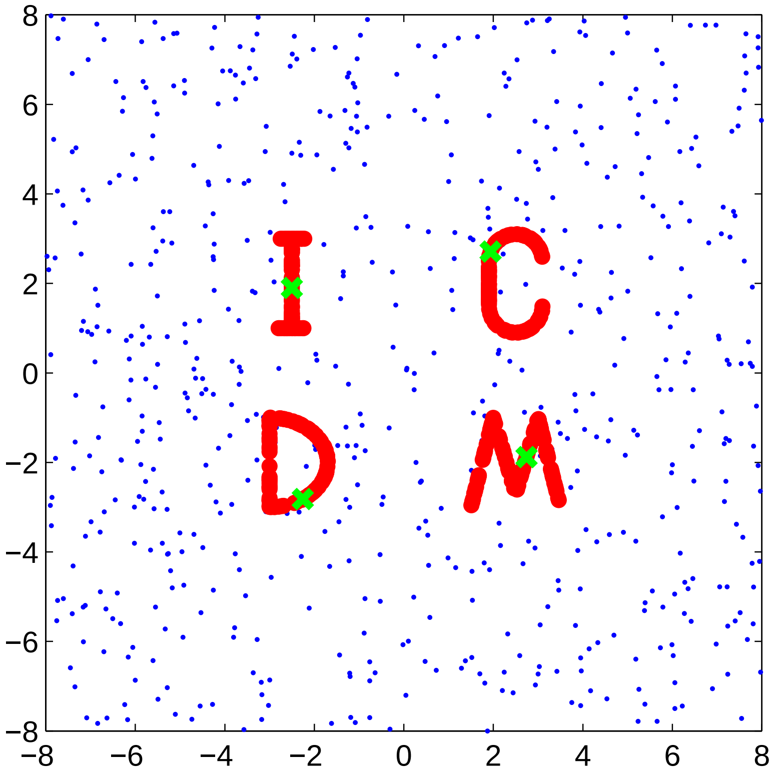


Figure 3.15: Synthetic data set: the examples selected by *GRADE* are denoted as green ‘x’s.

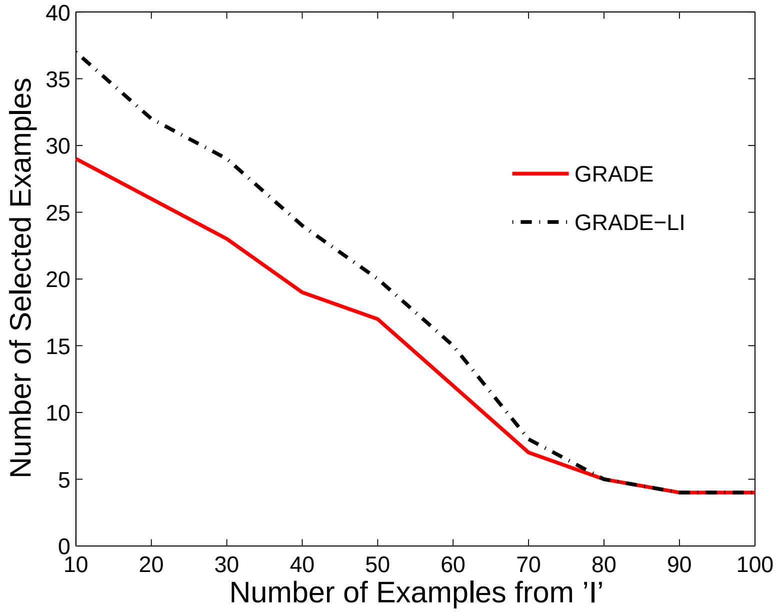


Figure 3.16: Comparison between *GRADE* and *GRADE-LI* on the synthetic data set in Fig. 3.15.

Fig. 3.17 to Fig. 3.20 show the comparison results on the 4 data sets. Note that for *GRADE-LI*, we use the exact upper bound as input. With the Ecoli data set, to discover all the classes, Interleave needs 41 label requests on average, *MALICE* needs 36 label requests, RS needs 43 label requests on average, *GRADE* needs 6 label requests, and *GRADE-LI* needs 32 label requests; with the Glass data set, to discover all the classes, Interleave needs 24 label requests on average, *MALICE* needs 18 label requests, RS needs 31 label requests on average, and both *GRADE* and *GRADE-LI* need 14 label requests; with the Abalone data set, to discover all the classes, Interleave needs 333 label requests on average, *MALICE* needs 179 label requests, RS needs 483 label requests on average, *GRADE* needs 149 label requests, and *GRADE-LI* needs 318 label requests; with the Shuttle data set, Interleave needs 140 label requests on average, *MALICE* needs 87 label requests, RS needs 512 label requests on average, *GRADE* needs 33 label requests, and *GRADE-LI* needs 36 label requests.

From these results, we have the following observations. First, with all the data sets, *GRADE* is much better than *MALICE*, which is the prior best method for rare category detection. Notice that both of the two algorithms need the number of classes as well as the proportions of all the classes as input. Second, the performance of *GRADE-LI* is better than *MALICE* on all the data sets except the Abalone data set. The reason might be the following. With the Abalone data set, the proportion of the majority class is 16.50%, the proportion of the largest minority class is 15.18%, and the proportion of the smallest minority class is 0.34%. As we have shown with the synthetic data set in the last subsubsection, if the proportions of different minority classes do not vary a lot, which is the case for the other 3 data sets, the performance of *GRADE-LI* is similar to *GRADE*. On the other hand, if the proportions of different minority classes vary a lot, which is the case for the Abalone data set, the performance of *GRADE-LI* is worse than *GRADE*. It should be pointed out that compared with *MALICE*, *GRADE-LI* needs much less information: only an upper bound on the proportions of the minority classes is needed. The reduction in the prior knowledge about the data set is significant especially when the number of classes in the data set is large, as with the Abalone data set.

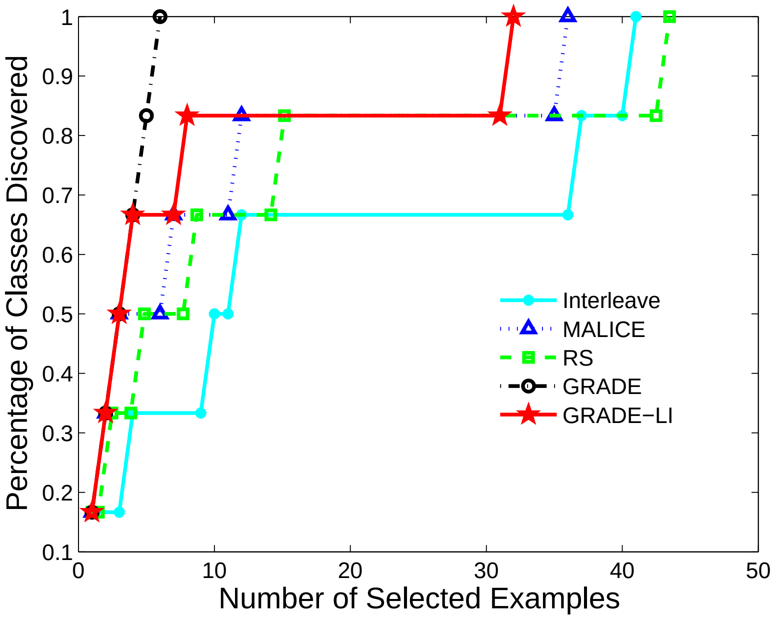


Figure 3.17: Comparison on Ecoli data set.

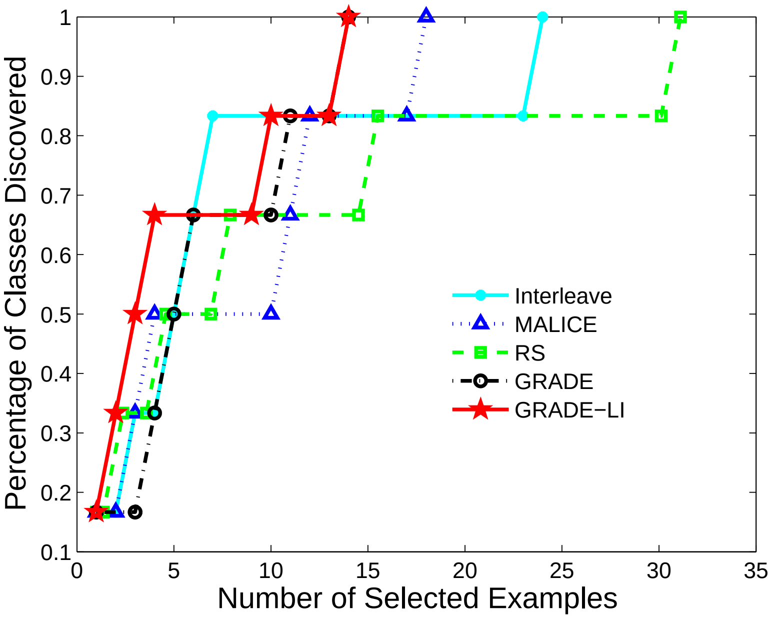


Figure 3.18: Comparison on Glass data set.

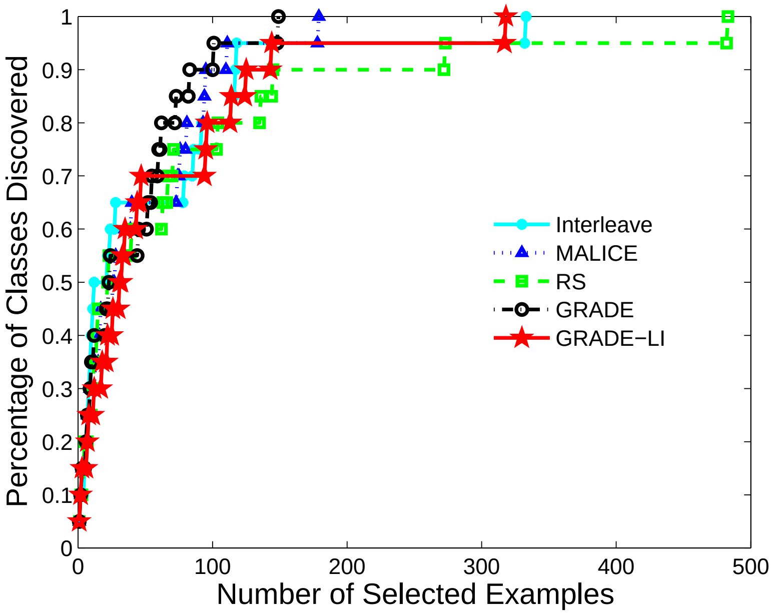


Figure 3.19: Comparison on Abalone data set.

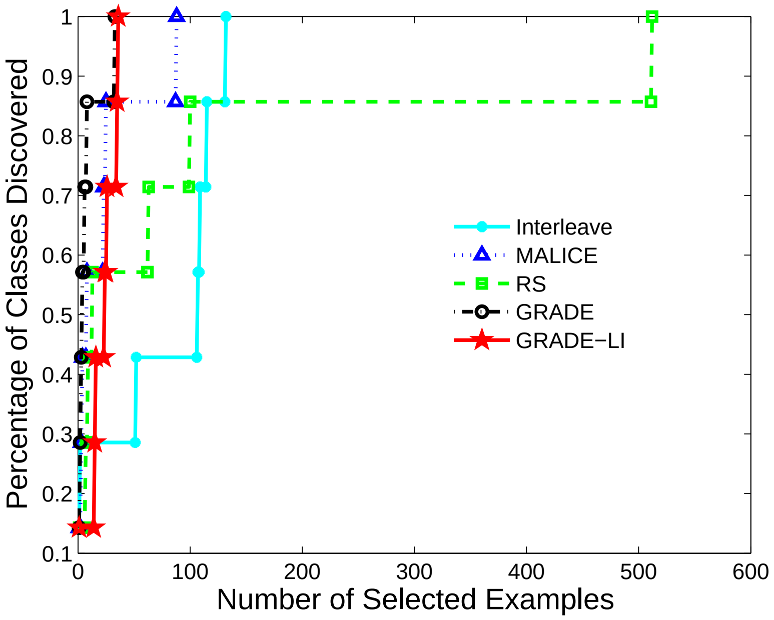


Figure 3.20: Comparison on Shuttle data set.

The *GRADE-LI* algorithm needs an upper bound on the proportions of all the minority classes as input. Next we study the robustness of *GRADE-LI* with respect to this upper bound using the 4 real data sets. To this end, we add and subtract 15% from the exact upper bounds, and provide *GRADE-LI* with the perturbed upper bounds. In Fig. 3.21, we compare the following 5 methods in terms of the total number of label requests: *MALICE*, *GRADE*, *GRADE-LI*, *GRADE-LI* with $p = 0.85 \times \max_{c=2}^{m} p_c$, and *GRADE-LI* with $p = 1.15 \times \max_{c=2}^{m} p_c$.

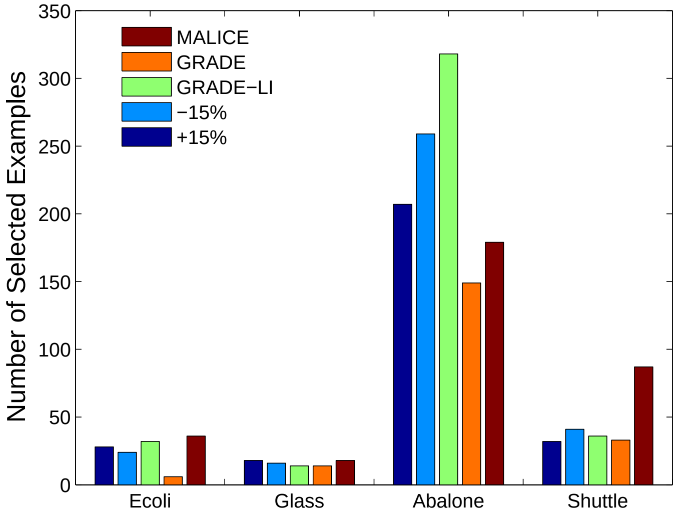


Figure 3.21: Robustness study: -15% denotes the performance of *GRADE-LI* after we subtract 15% from the exact upper bounds; +15% denotes the performance *GRADE-LI* after we add 15% to the exact upper bounds.

From Fig. 3.21, we can see that *GRADE-LI* is quite robust against small perturbations in the upper bounds. For example, with the Glass data set, to find all the classes, using *GRADE-LI*, if $p = \max_{c=2}^{m} p_c$, we need 14 label requests; if $p = 0.85 \times \max_{c=2}^{m} p_c$, we need 16 label requests; if $p = 1.15 \times \max_{c=2}^{m} p_c$, we need 18 label requests.

### 3.3.4 Discussion

In this subsection, we discuss about some implementation issues of *GRADE* and *GRADE-LI*.

#### Pair-wise similarity

Note that in Step 3 of Alg. 5 and Alg. 6, the Gaussian kernel is used to get the pair-wise similarity (Equation 3.12). In general, we can use the function $\varphi(\cdot, \cdot)$ to define $W'$, which satisfies the following conditions: $\varphi(x_i, x_k) = \phi(\frac{x_i - x_k}{\sigma})$, $\forall x \in \mathbb{R}^d$, $\phi(x) \geq 0$, $\int \phi(x)dx = 1$, $\sup_x \phi(x) < \infty$, and $\lim_{\|x\| \to \infty} \phi(x) \prod_{j=1}^{d} x^j = 0$, where $x^j$ is the $j^{\text{th}}$ feature component of $x$. In this case, it can be proven that Lemma 3 still holds. In practise, besides the above similarity functions, we can use the one that is best suited for our application, such as the cosine similarity for text data.

#### Calculating the global similarity matrix

The time complexity of both *GRADE* and *GRADE-LI* is dominated by the matrix inversion in calculating the global similarity matrix in Step 6 of Alg. 5 and Alg. 6, which is $O(n^3)$ with naive implementation, and $O(n^{2.376})$ with Coppersmith and Winograd's implementation [Coppersmith & Winograd, 1987]. To speed up this process, we can make use of one of the following two strategies.

- The global similarity matrix $A$ is defined as $(I_{n \times n} - \alpha W)^{-1}$. Based on Taylor's expansion, we have $A = I_{n \times n} + \sum_{i=1}^{\infty} \alpha^i W^i$. Therefore, we can use the following iterative process to get $A$, i.e., $A_0 = I_{n \times n}$, $A_l = \alpha W \times A_{l-1} + I_{n \times n}$. It is easy to see that $A = A_\infty$, and we can approximate $A$ with $A_L$, i.e., $A \approx I_{n \times n} + \sum_{i=1}^{L} \alpha^i W^i$. This strategy is best suited for the cases where $W$ is a sparse matrix with $z$ non-zero elements. In this case, with naive implementation, the complexity of calculating $A$ is $O(Lnz)$, which can be further improved with more sophisticated methods such as [Yuster & Zwick, 2005].
- Another way to accelerate the calculation of the global similarity matrix is by means of eigen-decomposition of $W$. Let $W = \sum_{i=1}^{n} \lambda_i u_i u_i^T$, where $\lambda_i$ is the $i^{\text{th}}$ largest eigen-value, and $u_i$ is the corresponding eigen-vector s.t., $\|u_i\| = 1$. Therefore, $A = \sum_{i=1}^{n} \frac{1}{1-\alpha\lambda_i} u_i u_i^T$. We can approximate $W$ using the $r$ largest eigen-values, i.e., $W \approx \sum_{i=1}^{r} \lambda_i u_i u_i^T$. In this way, $A \approx \sum_{i=1}^{r} \frac{1}{1-\alpha\lambda_i} u_i u_i^T$. Since we are only interested in the $r$ largest eigen-values of $W$, by making use of efficient algorithms such as [Lehoucq & Sorensen, 1996], the processing time will be greatly reduced.

#### Stopping criterion

In *GRADE-LI* the querying process is stopped once we have at least one labeled example from each class. Note that the rest of *GRADE-LI* does not need the information of the number of classes to work. When this information is not available, we need to design a stopping criterion for the querying process. One possible criterion is to set a threshold on the number of consecutive uninformative label requests, which is the number of consecutive labeled examples whose classes have already been discovered, and stop the querying process once the threshold is reached. A very small threshold may result in the overlooking of certain minority classes; whereas a very large threshold may result in unaffordable labeling efforts. Therefore, we need to set a proper threshold according to specific applications. It should be pointed out that when the stopping criterion is not based on the number of classes in the data set, *GRADE-LI* may need to label more examples, but the number of label requests to discover all the class is the same as reported in Subsection 3.3.3.

### Picking the value of $\alpha$

As we have discussed in Subsection 3.3.1, if the pair-wise similarity between examples from different classes is 0, and the value of $\alpha$ is close to 1, the global similarity matrix $A$ can be approximated using the eigen-vectors that correspond to eigen-value 1. In real applications, the pair-wise similarity between examples from different classes is rarely 0. In this case, if the value of $\alpha$ is close to 1, the resulting matrix $A$ would be unreliable since the eigen-vectors that correspond to eigen-value 1 are not directly related to the probability density function of each class. Therefore, for the synthetic data sets, we set $\alpha = 0.8$; whereas for the real data sets, we set $\alpha = 0.6$.

## 3.4 Summary of Rare Category Detection

In this chapter, we have discussed our work on rare category detection. Different from existing work, we target the challenging cases where the support regions of the majority and minority classes overlap with each other in the feature space. The overlapping phenomenon is observed in many real applications, such as financial fraud detection and spam image detection. Yet it has not been well studied before.

For data with feature representations, we propose *NNDB*, *ALICE*, and *MALICE* algorithms, which need full prior information of the data set as input, including the number of classes and the proportions of different classes. We also propose *SEDER* algorithm, which is prior-free. The basic idea of these algorithms is to select the examples from the regions where the density changes the most, since these examples have a high probability of coming from the minority classes. For graph data (relational data), we propose *GRADE* algorithm, which needs full prior information of the data set as input, and *GRADE-LI* algorithm, which only needs an upper bound on the proportions of the minority classes. The basic idea of these algorithms is to perform implicit feature transformation based on the global similarity, which results in more compact clusters of the minority classes. Furthermore, we provide theoretical guarantees for these algorithms under the smoothness assumption of the majority classes and the compactness assumption of the minority classes. Based on the experimental results on both synthetic data sets and real data sets, we have the following conclusions.

1. Given full prior information about the data set, including the number of classes and the proportions of different classes, the proposed *GRADE* algorithm performs the best on our data sets, since it enjoys the benefits of both the global similarity and the *MALICE* algorithm.
2. If we only know an upper bound on the proportions of the minority classes, but do not know the number of classes or the exact proportions of different classes, we can apply the *GRADE-LI* algorithm to detect the rare categories.
3. The *GRADE-LI* algorithm is robust to small perturbations in the upper bound.
4. The *GRADE* and *GRADE-LI* algorithms essentially make use of the graph structure to help with rare category detection. They can be applied to both data with feature representations and graph data. For data with feature representations, we first need to construct a pair-wise similarity matrix based on the features.
5. In our experiments, the performance of *GRADE* is better than *MALICE*, and both of them are much better than existing methods, such as random sampling and Interleave (the best method proposed

in [Pelleg & Moore, 2004]). However, the processing time of *GRADE* is more than *MALICE* since it needs to compute the global similarity for each pair of data points.

6. The *MALICE* algorithm is robust to small perturbations in the class priors.
7. If we do not have any prior information about the data set, we can apply the *SEDER* algorithm to detect the rare categories. In our experiments, if a data set is 'moderately' skewed, its performance is comparable to *MALICE*; if a data set is 'highly' skewed, its performance is not as good as *MALICE*. However, based on our experiments, the performance of *SEDER* is always better than random sampling.

### 3.4.1 Limitations

The limitations of the current work on rare category detection can be summarized as follows.

1. Violations of the assumptions. It is not clearly how the performance of our algorithms will deteriorate if one of more of the assumptions are not satisfied. For example, the majority class may consist of several sub-classes of different sizes.
2. Computational issues. Some of the proposed algorithms do not scale well to very large data sets. For example, in *MALICE*, a major component is nearest neighbor search. This can be very time-consuming even with a well designed data structure such as kd-trees [Moore, 1991]. Parallelizing these algorithms may be one way to address this problem.
3. Application to real world data sets. The current algorithms are evaluated using benchmark data sets only. It is of great interest to test their performance on real world data sets, e.g., financial fraud detection.
4. Other density estimators. In *SEDER*, the specially designed exponential families are used to estimate the density as well as to calculate the scores for each data point. However, there is no systematic comparison between this density estimator and the large number of other methods on density estimation from both theoretical and practical point of view.

# Chapter 4

# Rare Category Characterization

In Chapter 3, we have introduced various algorithms for rare category detection, which result in a set of labeled examples. Based on this labeled set, a natural follow-up step is rare category characterization, i.e., to characterize the minority classes in order to identify all the rare examples in the data set. For example, in Medicare fraud detection, once we have discovered a bogus claim for durable equipments (e.g., wheelchairs, breathing machines), we may want to find the fraud patterns related to such equipments in order to prevent similar fraudulent claims in the future. To this end, in this chapter, we focus on rare category characterization, the second task in the supervised settings.

Rare category characterization is to characterize the minority classes for the purpose of understanding and correctly classifying those classes. Here, our key observation is that the minority classes often exhibit *compactness*. That is, each minority class often forms a compact cluster. For example, the fraudulent people often make multiple similar transactions to maximize their profits [Chau *et al.*, 2006]. For rare diseases, the patients with the same type of rare disease often share similar genes or chromosomal abnormalities [EURODIS, 2005].

In this chapter, we propose *RACH* by exploring such compactness for rare category characterization. The core of *RACH* is to represent the minority classes with a hyper-ball. We present the optimization framework as well as an effective algorithm to solve it. Furthermore, we show how *RACH* can be naturally kernelized. We also analyze the complexity of *RACH*. Finally, we justify the effectiveness of the proposed *RACH* by both theoretical analysis and empirical evaluations.

The main contributions of this chapter can be summarized as follows.

**Problem Formulation.** We formulate the problem of rare category characterization as an optimization problem, which takes into account both labeled an unlabeled examples, and imposes different constraints for different types of data;

**Algorithm Design.** We design an effective algorithm to find the solution of the optimization problem. It repeatedly converts the original problem into a convex optimization problem, and solves it in its dual form by a projected subgradient method, which is well justified theoretically.

The rest of this chapter is organized as follows. In Section 4.1, we propose the optimization framework to provide a compact representation for the minority class with justification, followed by the conversion of this framework to the convex optimization problem as well as its dual form. Then we introduce the *RACH* algorithm to solve the dual problem with performance guarantees in Section 4.2, and the kernelized *RACH* algorithm in Section 4.3. Finally, following the experimental results presented in Section 4.4, we give a brief summary of rare category characterization in Section 4.5.

## 4.1 Optimization Framework

In this section, we present our optimization framework, after we introduce the additional notation and the pre-processing step.

### 4.1.1 Additional Notation

For the sake of simplicity, we assume that there is only one majority class and one minority class in the data set, i.e., $m = 2$. (Multiple majority and minority classes can be converted into several binary problems.) Throughout this chapter, we will use calligraphic capital letters to denote sets. Let $x_1, \ldots, x_{n_1} \in \mathbb{R}^d$ denote the labeled examples from the majority class, which correspond to $y_i = 1, i = 1, \ldots, n_1$; let $x_{n_1+1}, \ldots, x_{n_1+n_2} \in \mathbb{R}^d$ denote the labeled examples from the minority class, which correspond to $y_j = 2, j = n_1 + 1, \ldots, n_1 + n_2$; let $x'_{n_1+n_2+1}, \ldots, x'_{n'} \in \mathbb{R}^d$ denote all the unlabeled examples. Here, $n_1$, $n_2$, and $n'$ denote the number of labeled examples from the majority class, the number of labeled examples from the minority class, and the total number of examples, both labeled and unlabeled. $d$ is the dimensionality of the input space. Our goal is to identify a list of unlabeled examples which are believed to come from the minority class with high precision and recall.

### 4.1.2 Assumptions

In many imbalanced problems, it is often the case that the rare examples from the same minority class are very close to each other, whereas the examples from the same majority class may be scattered in the feature space. This assumption is also used in [Wu *et al.*, 2007][Pelleg & Moore, 2004], etc, when dealing with imbalanced data sets, either explicitly or implicitly. For example, in financial fraud detection, fraud transactions of the same type are very similar to each other, whereas the normal transactions are more scattered in terms of their features compared with these fraud transactions. Furthermore, we also assume that the rare examples can be enclosed by a minimum-radius hyper-ball in the input space without including too many majority class examples. This seemingly rigorous assumption will become more flexible when we use the high-dimensional feature space instead of the input space via the kernel trick in Section 4.3. With this assumption, we allow the support regions of the majority and minority classes to overlap with each other. In other words, our algorithms works in both the separable and non-separable cases, which makes it more suitable for real applications.

### 4.1.3 Pre-processing: Filtering

**Algorithm 7** Filtering Process for Rare Category Characterization

```
Input: $x_1, \ldots, x_{n_1+n_2}, x'_{n_1+n_2+1}, \ldots, x'_{n'}$
Output: $x_{n_1+n_2+1}, \ldots, x_n$
if $n_2 > 1$ then
    Estimate the center $b$ of the hyper-ball by one-class SVM [Schölkopf et al., 2001], using all the labeled minority examples
else
    Set the center $b = x_{n_1+1}$
end if
for $i = n_1 + n_2 + 1, \ldots, n'$ do
    Calculate the distance $d_i = \|x'_i - b\|$
end for
Calculate $p = \frac{n_2}{n_1+n_2}$; set $D_{\text{thre}}$ as the $(n' - n_1 - n_2) \times p^{\text{th}}$ smallest value among all $d_i$ $(i = n_1 + n_2 + 1, \ldots, n')$;
$n = n_1 + n_2$
for $i = n_1 + n_2 + 1, \ldots, n'$ do
    if $d_i \leq 3 \cdot D_{\text{thre}}$ then
        $n = n + 1$, $x_n = x'_i$
    end if
end for
```

In the unlabeled data, there might be some examples which are far away from the hyper-ball. These examples can be safely classified as majority class examples without affecting the performance of our classifier. Therefore, we first filter the unlabeled data to exclude such examples from the following optimization framework, and only focus on the examples that are close to the hyper-ball. The filtering process is described in Alg. 7. It takes both the labeled and the unlabeled examples as input, and outputs a set of unlabeled examples which are close to the hyper-ball. Here, $n - n_1 - n_2$ is the number of unlabeled examples after the filtering process. The algorithm works as follows. It first estimates the center $b$ of the hyper-ball using one-class SVM [Schölkopf *et al.*, 2001] or a single labeled example; then it estimates the proportion $p$ of the rare examples in the unlabeled data using the labeled data; finally, it calculates the distance threshold $D_{\text{thre}}$ based on $p$, which is used to filter out the unlabeled examples far away from the hyper-ball. Notice that $3 \times D_{\text{thre}}$ is actually used to filter the unlabeled data. This is to ensure that we do not miss any rare example. We should point out that the filtering process is orthogonal to the other parts of the proposed algorithm. In the remainder of this chapter, unlabeled data (unlabeled examples) refer to the examples output by the filtering process.

### 4.1.4 Problem Formulations

Now, we are ready to give the problem formulations for rare category characterization. We first give its original formulation and illustrate its intuitions. Then, we present its convex approximation together with its dual form.

*Original Formulation.* To find the center and radius of the minimum-radius hyper-ball, we construct the following optimization framework, which is inspired by one-class SVM [Schölkopf *et al.*, 2001].

**Problem 4.1**

$$\begin{aligned}
\min_{R^2,b,\alpha,\beta} \; & R^2 + C_1 \sum_{i=1}^{n_1} \alpha_i + C_2 \sum_{k=1}^{n-n_1-n_2} \beta_k \\
\text{s.t., } & \|x_i - b\|^2 \geq R^2 - \alpha_i,\; i = 1, \dots, n_1 \\
& \alpha_i \geq 0,\; i = 1, \dots, n_1 \\
& \|x_j - b\|^2 \leq R^2,\; j = n_1 + 1, \dots, n_1 + n_2 \\
& \|x_k - b\|^2 \leq R^2 + \beta_{k-n_1-n_2},\; k = n_1 + n_2 + 1, \dots, n \\
& \beta_{k-n_1-n_2} \geq 0,\; k = n_1 + n_2 + 1, \dots, n
\end{aligned}$$

where $R$ is the radius of the hyper-ball; $b$ is the center of the hyper-ball; $C_1$ and $C_2$ are two positive constants that balance among the three terms in the objective function; $\alpha$ and $\beta$ correspond to the non-negative slack variables for the labeled examples from the majority class and the unlabeled examples; $\alpha_i$ and $\beta_k$ are the $i^{\text{th}}$ and $k^{\text{th}}$ component of $\alpha$ and $\beta$ respectively.

In Problem 4.1, we minimize the squared radius of the hyper-ball and a weighted combination of the slack variables. Furthermore, we have three types of constraints with respect to the training data. The first type is for the labeled examples from the majority class, i.e., they should be outside the hyper-ball. Notice that these are not strict constraints, and the labeled examples from the majority class falling inside the hyper-ball correspond to positive slack variables $\alpha_i$. In this way, we allow the support regions of the majority and minority classes to overlap with each other. The second type is for the labeled examples from the minority class, i.e., they should be inside the hyper-ball. In contrast, these are strict constraints, and the hyper-ball should be large enough to enclose all the labeled rare examples. The last type is for the unlabeled examples, i.e., we want the hyper-ball to enclose as many unlabeled examples as possible. Different from the second type of constraints, these constraints are not strict, and the examples falling outside the hyper-ball correspond to positive slack variables $\beta_k$. The intuition of this type of constraints is that after the filtering process, the unlabeled examples are all in the neighborhood of the minority class. The support region of the minority class should have a higher density compared with the rest of the neighborhood. Therefore, we want the hyper-ball to enclose as many unlabeled examples as possible.

*Convex Approximation of Problem 4.1.* Note that Problem 4.1 is difficult to solve due to the first type of constraints on the labeled examples from the majority class, which make this framework non-convex in the center $b$. To address this problem, we approximate these constraints based on first-order Taylor expansion around the current center $\tilde{b}$, and have the following optimization problem, which is convex.

**Problem 4.2 (Convex Approximation of Problem 4.1)**

$$\begin{aligned}
\min_{R^2,b,\alpha,\beta} \; & R^2 + C_1 \sum_{i=1}^{n_1} \alpha_i + C_2 \sum_{i=1}^{n-n_1-n_2} \beta_i \\
\text{s.t., } & R^2 - \alpha_i - \|x_i\|^2 + \|\tilde{b}\|^2 + 2(x_i - \tilde{b})^T b \leq 0,\; i = 1, \dots, n_1 \\
& \alpha_i \geq 0,\; i = 1, \dots, n_1 \\
& \|x_j - b\|^2 \leq R^2,\; j = n_1 + 1, \dots, n_1 + n_2 \\
& \|x_k - b\|^2 \leq R^2 + \beta_{k-n_1-n_2},\; k = n_1 + n_2 + 1, \dots, n \\
& \beta_k \geq 0,\; k = 1, \dots, n - n_1 - n_2
\end{aligned}$$

Based on Problem 4.2, we find the solution to Problem 4.1 in an iterative way. To be specific, in each iteration step, we form Problem 4.2 based on the current estimate $\tilde{b}$ of the center, find the optimal $R^2$, $b$, $\alpha$ and $\beta$, and then update Problem 4.2 based on the new center $b$. We stop the iteration when the solutions in two consecutive steps are very close to each other or when the maximum number of iteration steps is reached.

*Dual Problem for Problem 4.2.* It is obvious that Problem 4.2 satisfies Slater's condition [Boyd & Vandenberghe, 2004]. Therefore, we solve this problem via the following dual problem.

**Problem 4.3 (Dual Problem for Problem 4.2)**

$$\max_{\lambda} \sum_{j=n_1+1}^{n} \lambda_j \|x_j\|^2 - \sum_{i=1}^{n_1} \lambda_i \|x_i\|^2 + \sum_{i=1}^{n_1} \lambda_i \|\tilde{b}\|^2 - \frac{\| \sum_{j=n_1+1}^{n} \lambda_j x_j - \sum_{i=1}^{n_1} \lambda_i (x_i - \tilde{b}) \|^2}{\sum_{j=n_1+1}^{n} \lambda_j}$$

$$\text{s.t., } 1 + \sum_{i=1}^{n_1} \lambda_i = \sum_{j=n_1+1}^{n} \lambda_j$$

$$0 \le \lambda_i \le C_1,\ i = 1, \ldots, n_1$$

$$0 \le \lambda_j,\ j = n_1 + 1, \ldots, n_1 + n_2$$

$$0 \le \lambda_k \le C_2,\ k = n_1 + n_2 + 1, \ldots, n$$

where $\lambda$ is the vector of Lagrange multipliers, $\lambda_i$, $i = 1, \ldots, n_1$ are associated with the constraints on the labeled examples from the majority class, $\lambda_j$, $j = n_1 + 1, \ldots, n_1 + n_2$ are associated with the constraints on the labeled examples from the minority class, and $\lambda_k$, $k = n_1 + n_2 + 1, \ldots, n$ are associated with the constraints on the unlabeled examples. Furthermore, based on the KKT conditions of Problem 4.2, the center $b$ of the hyper-ball can be calculated as follows.

$$b = \frac{\sum_{j=n_1+1}^{n} \lambda_j x_j - \sum_{i=1}^{n_1} \lambda_i (x_i - \tilde{b})}{\sum_{j=n_1+1}^{n} \lambda_j} \tag{4.1}$$

## 4.2 Optimization Algorithm: *RACH*

Here, we present the proposed optimization algorithm to solve Problem 4.1. The basic idea is as follow: after an initialization step; we will recursively formulate Problem 4.2 using the current estimate $\tilde{b}$ for the center of the hyper-ball; and then solve Problem 4.2 in its dual form (Problem 4.3) by a projected subgradient method.

### 4.2.1 Initialization Step

First, we need to initialize the center $b$ of the hyper-ball and the Lagrange multipliers $\lambda$ in Problem 4.3, which is summarized in Alg. 8. It takes as input both the labeled and the unlabeled examples (after the filtering process), and outputs the initial estimates of the center $b$ and the Lagrange multipliers $\lambda$. In Step 1, it initializes the center $b$ and the radius $R$ of the hyper-ball using one-class SVM [Schölkopf *et al.*, 2001] if we have more than one labeled examples from the minority class; otherwise, it uses the only labeled rare example as the center $b$, and the smallest distance between this example and the nearest labeled example from the majority class as $R$. In Step 2, it initializes the Lagrange multipliers based on the KKT conditions of Problem 4.1. For a labeled example from the majority class, if its distance to the center $b$ is bigger than $R$, $\lambda_i = 0$; if the distance is less than $R$, $\lambda_i = C_1$; and if the distance is equal to $R$, we use $\frac{C_1}{2}$ as the value for $\lambda_i$. For a labeled example from the minority class, if its distance to the center $b$ is less than $R$, $\lambda_j = 0$; otherwise, we use $\frac{C_1+C_2}{2}$ as the value for $\lambda_j$. For an unlabeled example, if its distance to the center $b$ is less than $R$, $\lambda_k = 0$; if the distance is bigger than $R$, $\lambda_k = C_2$; and if the distance is equal to $R$, we use $\frac{C_2}{2}$ as the value for $\lambda_k$.

**Algorithm 8** Initialization for *RACH*

```
Input: $x_1, \ldots, x_n$
Output: initial estimates of $b$ and $\lambda$
if $n_2 > 1$ then
    initialize the center $b$ and the radius $R$ of the hyper-ball using one-class SVM [Schölkopf et al., 2001]
else
    set $b = x_{n_1+1}$, and set $R$ as the smallest distance between $x_{n_1+1}$ and the nearest labeled example from the majority class
end if
Initialize $\lambda$ as follows.
        For $1 \leq i \leq n_1$, if $\|x_i - b\| > R$, $\lambda_i = 0$; if $\|x_i - b\| < R$, $\lambda_i = C_1$; if $\|x_i - b\| = R$, $\lambda_i = \frac{C_1}{2}$
        For $n_1 + 1 \leq j \leq n_1 + n_2$, if $\|x_j - b\| < R$, $\lambda_j = 0$; if $\|x_j - b\| = R$, $\lambda_j = \frac{C_1+C_2}{2}$
        For $n_1 + n_2 + 1 \leq k \leq n$, if $\|x_k - b\| < R$, $\lambda_k = 0$; if $\|x_k - b\| > R$, $\lambda_k = C_2$; if $\|x_k - b\| = R$, $\lambda_k = \frac{C_2}{2}$
```

### 4.2.2 Projected Subgradient Method for Problem 4.3

Projected subgradient methods minimize a convex function $f(\lambda)$ subject to the constraint that $\lambda \in \mathcal{X}$, where $\mathcal{X}$ is a convex set, by generating the sequence $\{\lambda^{(t)}\}$ via

$$\lambda^{(t+1)} = \prod_{\mathcal{X}}(\lambda^{(t)} - \tau_t \nabla^{(t)})$$

where $\nabla^{(t)}$ is the (sub)gradient of $f$ at $\lambda^{(t)}$, $\tau_t$ is the step size, and $\prod_{\mathcal{X}}(x) = \arg\min_y\{\|x - y\| : y \in \mathcal{X}\}$ is the Euclidean projection of $x$ onto $\mathcal{X}$. To solve Problem 4.3, the gradient descent step is straight-forward.[1] Next, we will focus on the projection step, where $\mathcal{X} = \{\lambda : 1 + \sum_{i=1}^{n_1} \lambda_i = \sum_{j=n_1+1}^{n} \lambda_j, 0 \leq \lambda_i \leq C_1,\ i = 1, \ldots, n_1; 0 \leq \lambda_j,\ j = n_1 + 1, \ldots, n_1 + n_2; 0 \leq \lambda_k \leq C_2,\ k = n_1 + n_2 + 1, \ldots, n\}$.

In the projection step, we consider the following optimization problem.

**Problem 4.4 (Projection Step of Problem 4.3)**

$$\min_{\lambda} \frac{1}{2}\|\lambda - v\|_2^2 \quad \text{s.t.}, \sum_{i=1}^{n} a_i \lambda_i = z, 0 \leq \lambda_i \leq \varepsilon_i$$

where $a_i$ ($i = 1, \ldots, n$) denote a set of constants which are either 1 or -1; $z$ is a constant; $v$ can be seen as the updated vector for $\lambda$ based on gradient descent in each iteration step of the projected subgradient method, or $\lambda^{(t)} - \tau_t \nabla^{(t)}$; and $\varepsilon_i$ is the upper bound for $\lambda_i$. Without loss of generality, we assume that $\varepsilon_i > 0$, $i = 1, \ldots, n$. For this optimization problem, define $S_+ = \{i : 1 \leq i \leq n, a_i = 1\}$, and $S_- = \{i : 1 \leq i \leq n, a_i = -1\}$.

Before we give our optimization algorithm for Problem 4.4, we first give the following lemma, which is the key for solving Problem 4.4.[2]

**Lemma 5.** *Let $\lambda$ be the optimal solution to Problem 4.4. Let $s$ and $t$ be two indices such that $s, t \in S_+$ or $s, t \in S_-$, and $v_s > v_t$. If $\lambda_s = 0$, then $\lambda_t$ must be zero as well. On the other hand, let $s'$ and $t'$ be two indices such that $s', t' \in S_+$ or $s', t' \in S_-$, and $v_{s'} - \varepsilon_{s'} < v_{t'} - \varepsilon_{t'}$. If $\lambda_{s'} = \varepsilon_{s'}$, then $\lambda_{t'}$ must be $\varepsilon_{t'}$ as well.*

[1] Note that in our case, we are maximizing a concave function in Problem 4.3, and gradient ascent is actually used in *RACH*.

[2] Note that in [Duchi *et al.*, 2008], the authors addressed a much simpler problem where $a_i = 1$, and $\varepsilon_i = \infty$, $i = 1, \ldots, n$.

*Proof.* The Lagrange function of Problem 4.4:

$$L(\lambda, \theta, \zeta, \eta) = \frac{1}{2}\|\lambda - v\|^2 + \theta(\sum_{i=1}^{n} a_i\lambda_i - z) - \sum_{i=1}^{n} \zeta_i\lambda_i - \sum_{i=1}^{n} \eta_i(\varepsilon_i - \lambda_i)$$

where $\theta$ is a Lagrange multiplier associated with the equality constraint; $\zeta$ and $\eta$ are two vectors of Lagrange multipliers associated with the inequality constraints whose elements are $\zeta_i$ and $\eta_i$ respectively. Taking the partial derivative of $L(\lambda, \theta, \zeta, \eta)$ with respect to $\lambda$ and set it to 0, we get

$$\lambda_i = v_i - a_i\theta + \zeta_i - \eta_i \tag{4.2}$$

For the first half of Lemma 1, suppose that $s, t \in S_+$. If $\lambda_s = 0$ and $\lambda_t > 0$, we have $\zeta_s \geq 0$, $\eta_s = 0$, $\zeta_t = 0$ and $\eta_t \geq 0$. Therefore, $v_s - \theta + \zeta_s = 0$ and $v_t - \theta - \eta_t > 0$, which can not be satisfied simultaneously since $v_s > v_t$. Therefore, if $\lambda_s = 0$, $\lambda_t$ must be zero as well. Similar proof can be applied when $s, t \in S_-$. For the second half of Lemma 1, suppose that $s', t' \in S_+$. If $\lambda_{s'} = \varepsilon_{s'}$ and $\lambda_{t'} < \varepsilon_{t'}$, we have $\zeta_{s'} = 0$, $\eta_{s'} \geq 0$, $\zeta_{t'} \geq 0$ and $\eta_{t'} = 0$. Therefore, $\lambda_{s'} - \varepsilon_{s'} = v_{s'} - \varepsilon_{s'} - \theta - \eta_{s'} = 0$ and $\lambda_{t'} - \varepsilon_{t'} = v_{t'} - \varepsilon_{t'} - \theta + \zeta_{t'} < 0$, which can not be satisfied simultaneously since $v_{s'} - \varepsilon_{s'} < v_{t'} - \varepsilon_{t'}$. Similar proof can be applied when $s, t \in S_-$. □

Besides the vector $v$, define the vector $v'$ such that its $i^{\text{th}}$ element $v'_i = v_i - \varepsilon_i$. Based on Lemma 5, for $S_+$ ($S_-$), we can keep two lists: the first list sorts the elements of $v$ whose indices are in $S_+$ ($S_-$) in an ascending order, and only a top portion of the list corresponds to 0 in $\lambda$; the second list sorts the elements of $v'$ whose indices are in $S_+$ ($S_-$) in a descending order, and only a top portion of the list corresponds to the elements of $\lambda$ that reach their upper bounds. For the remaining indices in $S_+$ ($S_-$), their corresponding elements in $\lambda$ are between 0 and the upper bound, and the Lagrange multipliers $\zeta_i = \eta_i = 0$. Therefore, according to Equation 4.2, $\lambda_i = v_i - \theta$ ($\lambda_i = v_i + \theta$). Finally, with respect to the value of $\theta$, we have the following lemma.

**Lemma 6.** *Let $\lambda$ be the optimal solution to Problem 4.4. Let $S^1_+$, $S^2_+$ and $S^3_+$ denote subsets of $S_+$ which correspond to the elements in $\lambda$ that are equal to 0, equal to the upper bound, and between 0 and the upper bound respectively. $S^1_+ \bigcup S^2_+ \bigcup S^3_+ = S_+$. Let $S^1_-$, $S^2_-$ and $S^3_-$ denote subsets of $S_-$ which correspond to the elements in $\lambda$ that are equal to 0, equal to the upper bound, and between 0 and the upper bound respectively. $S^1_- \bigcup S^2_- \bigcup S^3_- = S_-$. $\theta$ can be calculated as follows.*

$$\theta = \frac{\sum_{k \in S^3_+} v_k + \sum_{j \in S^2_+} \varepsilon_j - \sum_{k \in S^3_-} v_k - \sum_{j \in S^2_-} \varepsilon_j - z}{|S^3_+| + |S^3_-|} \tag{4.3}$$

*Proof.* According to the definition of $S^1_+$, $S^2_+$, $S^3_+$ and $S^1_-$, $S^2_-$, $S^3_-$, $\forall i \in S^1_+ \bigcup S^1_-$, $\lambda_i = 0$, $\zeta_i \geq 0$, $\eta_i = 0$; $\forall j \in S^2_+ \bigcup S^2_-$, $\lambda_j = \varepsilon_j$, $\zeta_j = 0$, $\eta_j \geq 0$; $\forall k \in S^3_+ \bigcup S^3_-$, $0 < \lambda_k < \varepsilon_k$, $\zeta_k = \eta_k = 0$. Furthermore, $\forall k \in S^3_+$, $\lambda_k = v_k - \theta$; $\forall k \in S^3_-$, $\lambda_k = v_k + \theta$. Therefore,

$$z = \sum_{i=1}^{n} a_i\lambda_i = \sum_{i \in S^2_+ \bigcup S^3_+} \lambda_i - \sum_{i \in S^2_- \bigcup S^3_-} \lambda_i = \sum_{j \in S^2_+} \varepsilon_j + \sum_{k \in S^3_+} (v_k - \theta) - \sum_{j \in S^2_-} \varepsilon_j - \sum_{k \in S^3_-} (v_k + \theta)$$

Solving this equation with respect to $\theta$, we get Equation 4.3. □

Based on Lemma 5 and Lemma 6, to solve Problem 4.4, we gradually increase the number of elements in $S_+^1$, $S_+^2$, $S_-^1$ and $S_-^2$, calculate $\theta$ accordingly, and determine the value of $\lambda$ which has the smallest value of $\frac{1}{2}\|\lambda - v\|_2^2$. Alg. 9 gives the details for solving Problem 4.3 in *RACH*.

**Algorithm 9** Projected Subgradient Method for Problem 4.3

```
Input: $x_1, \ldots, x_n$; step size $\tau$; $C_1$, $C_2$; $N_2$; $\tilde{b}$; initial estimate of $\lambda$
Output: $\lambda$
 1: Define $S_+ = \{n_1 + 1, \ldots, n\}$ and $S_- = \{1, \ldots, n_1\}$
 2: for $step = 1$ to $N_2$ do
 3:     Calculate $\nabla$ as follows:
            $l = 1, \ldots, n_1 :$
            $\nabla_l = -\|x_l\|^2 + \|\tilde{b}\|^2 + \frac{2(\sum_{j=n_1+1}^{n} \lambda_j (x_j)^T - \sum_{i=1}^{n_1} \lambda_i (x_i - \tilde{b})^T)(x_l - \tilde{b})}{\sum_{j=n_1+1}^{n} \lambda_j}$
            $l = n_1 + 1, \ldots, n :$
            $\nabla_l = \|x_l\|^2 - \frac{(\sum_{j=n_1+1}^{n} \lambda_j (x_j)^T - \sum_{i=1}^{n_1} \lambda_i (x_i - \tilde{b})^T)}{(\sum_{j=n_1+1}^{n} \lambda_j)^2} \cdot (\sum_{j=n_1+1}^{n} \lambda_j (2x_l - x_j) + \sum_{i=1}^{n_1} \lambda_i (x_i - \tilde{b}))$
 4:     Update $\lambda$ via gradient ascent to obtain: $v = \lambda + \tau\nabla$
 5:     Calculate $v'$ as follows:
            $v'_i = v_i - C_1, \ i = 1, \ldots, n_1$
            $v'_j = -\infty, \ j = n_1 + 1, \ldots, n_1 + n_2$
            $v'_k = v_k - C_2, \ k = n_1 + n_2 + 1, \ldots, n$
 6:     Set $D = \infty$
 7:     for $I_1 = 1, I_2 = 1, I_3 = 1, I_4 = 1$ to $I_1 = |S_+| + 1, I_2 = |S_-| + 1, I_3 = |S_+| + 1, I_4 = |S_-| + 1$ do
 8:         Let $S_+^1 \subset S_+$ denote the subset of indices in $S_+$ such that the corresponding elements in $v$ are no
            larger than the $I_1^{\text{th}}$ largest element; let $S_+^2 \subset S_+$ denote the subset of indices in $S_+$ such that the
            corresponding elements in $v'$ are no smaller than the $I_3^{\text{th}}$ smallest element
 9:         If $S_+^1 \bigcap S_+^2 \neq \emptyset$ or $S_+^2 \bigcap \{n_1 + 1, \ldots, n_1 + n_2\} \neq \emptyset$, continue; otherwise, $S_+^3 = S_+ \backslash (S_+^1 \bigcap S_+^2)$
10:         Let $S_-^1 \subset S_-$ denote the subset of indices in $S_-$ such that the corresponding elements in $v$ are no
            larger than the $I_2^{\text{th}}$ largest element; let $S_-^2 \subset S_-$ denote the subset of indices in $S_-$ such that the
            corresponding elements in $v'$ are no smaller than the $I_4^{\text{th}}$ smallest element
11:         If $S_-^1 \bigcap S_-^2 \neq \emptyset$, continue; otherwise, $S_-^3 = S_- \backslash (S_-^1 \bigcap S_-^2)$
12:         Calculate $\theta = \frac{\sum_{k \in S_+^3} v_k - \sum_{k \in S_-^3} v_k + |S_+^2| C_2 - |S_-^2| C_1 - 1}{|S_+^3| + |S_-^3|}$
13:         Calculate $w$ as follows: $w_i = 0, \ i \in S_+^1 \bigcup S_-^1$; $w_i = C_2, \ i \in S_+^2$; $w_i = C_1, \ i \in S_-^2$; $w_i =$
            $v_i - \theta, \ i \in S_+^3$; $w_i = v_i + \theta, \ i \in S_-^3$
14:         If $\|v - w\| < D$, set $\lambda = w$ and $D = \|v - w\|$.
15:     end for
16: end for
```

In Step 3 of Alg. 9, we calculate the gradient of the objective function in Problem 4.3 at the current value of $\lambda$ and $\tilde{b}$; then in Step 4, $\lambda$ is updated via gradient ascent to obtain $v$. The remaining steps (Step 5- 16) are for the projection step (i.e., for solving Problem 4.4): in Step 5, we calculate the vector $v'$ using both $v$ and the upper bounds; to calculate the projection of $v$ on $\mathcal{X}$, in Step 7 to Step 15, we try different sizes for $S_+^1$, $S_+^2$, $S_-^1$ and $S_-^2$, calculate $\theta$ and $\lambda$ accordingly, and determine the projection of $v$ based on the distance between $v$ and $w$, where $w$ is calculated based on the current sets $S_+^1$, $S_+^2$, $S_+^3$, $S_-^1$, $S_-^2$ and $S_-^3$.

### 4.2.3 *RACH* for Problem 4.1

*Algorithm.* Now, we are ready to present the *RACH* algorithm (Alg. 10) to solve Problem 4.1. It is given the training data, the step size $\tau$, $C_1$, $C_2$, and the numbers of iteration steps $N_1$, $N_2$. (Note that $N_2$ is used in Alg. 9 in Step 3 of *RACH*.) The output is the unlabeled examples whose predicted class labels are class $\#2$. *RACH* works as follows. First of all, it initializes the center $b$ and the Lagrange multipliers $\lambda$ using Alg. 8; then it repeatedly forms Problem 4.3 based on the current estimate of the center $b$, and applies Alg. 9 to solve it, which is based on the projected subgradient method; after solving Problem 4.3, the center $b$ is updated using Equation 4.1; finally, we classify the unlabeled examples based on their corresponding Lagrange multipliers $\lambda_k$. The last step can be justified as follows. In Problem 4.1, for the unlabeled instances $x_k$, $k = n_1 + n_2 + 1, \ldots, n$, if $\|x_k - b\| < R$, $\lambda_k = 0$, then $y_k = 2$; if $\|x_k - b\| = R$, $0 < \lambda_k < C_2$, then $y_k = 2$; if $\|x_k - b\| > R$, $\lambda_k = C_2$, then $y_k = 1$.

**Algorithm 10** *RACH*: Rare Category Characterization

```
Input: $x_1, \ldots, x_n$; step size $\tau$; $C_1$, $C_2$; $N_1$, $N_2$
Output: unlabeled examples whose predicted class labels are 2
 1: Initialize the hyper-ball center $b$ and the Lagrange multipliers $\lambda$ by Alg. 8
 2: for $step = 1$ to $N_1$ do
 3:     Update the Lagrange multipliers $\lambda$ by Alg. 9 based on the current center $b$
 4:     Update the center $b$ based on Equation 4.1
 5: end for
 6: for $k = n_1 + n_2 + 1$ to $n$ do
 7:     if $\lambda_k < C_2$ then
 8:         set $y_k = 2$
 9:     else
10:         set $y_k = 1$
11:     end if
12: end for
```

*Concise Representation of the Minority Class.* From Alg. 10, we can also compute the radius $R$ of the hyberball, which is the maximum distance from the center $b$ to $x_{n_1+1}, \ldots, x_n$ whose Lagrange multipliers are less than the corresponding upper bounds. The resulting hyberball (fully described by the center $b$ and the radius $R$) provides a concise representation for the minority class. This representation can help us better understand the minority class. We can also use it to predict an unlabeled example as follows: if it is within the hyberball (i.e., its distance to the center $b$ is less than $R$), we classify it as a rare example; otherwise, it belongs to the majority class. Furthermore, domain experts who have no knowledge in machine learning and data mining may be able to better understand the learning results based on this representation.

*Computational Complexity of* RACH. It can be shown that the time complexity of *RACH* is $O(N_1 N_2 (n - n_1)^2 (n_1)^2)$. In practice, we can reduce the running time in the following three ways. First, we find that in our experiments *RACH* converges very quickly, often within a few tens of iterations. Second, in the applications that we are interested in, there are very few labeled examples from both the majority and the minority classes. A typical value for $n_1$ is a few tens, and a typical value for $n_2$ is less than 10. Finally, recall that in Section 4.1, we have applied Alg. 7 to filter out the unlabeled examples which are far away from the minority class. After this operation, only a small portion of the unlabeled data (typically less than $10\%$) is passed on to Alg. 10.

## 4.3 Kernelized *RACH* Algorithm

In this section, we briefly introduce how to generalize *RACH* to the high-dimensional feature space induced by kernels. The major benefit of kernelizing the *RACH* algorithm is that, instead of enclosing the rare examples by a minimum-radius hyper-ball, we can now use more complex shapes, which make our algorithm more flexible and may lead to more accurate classification results.

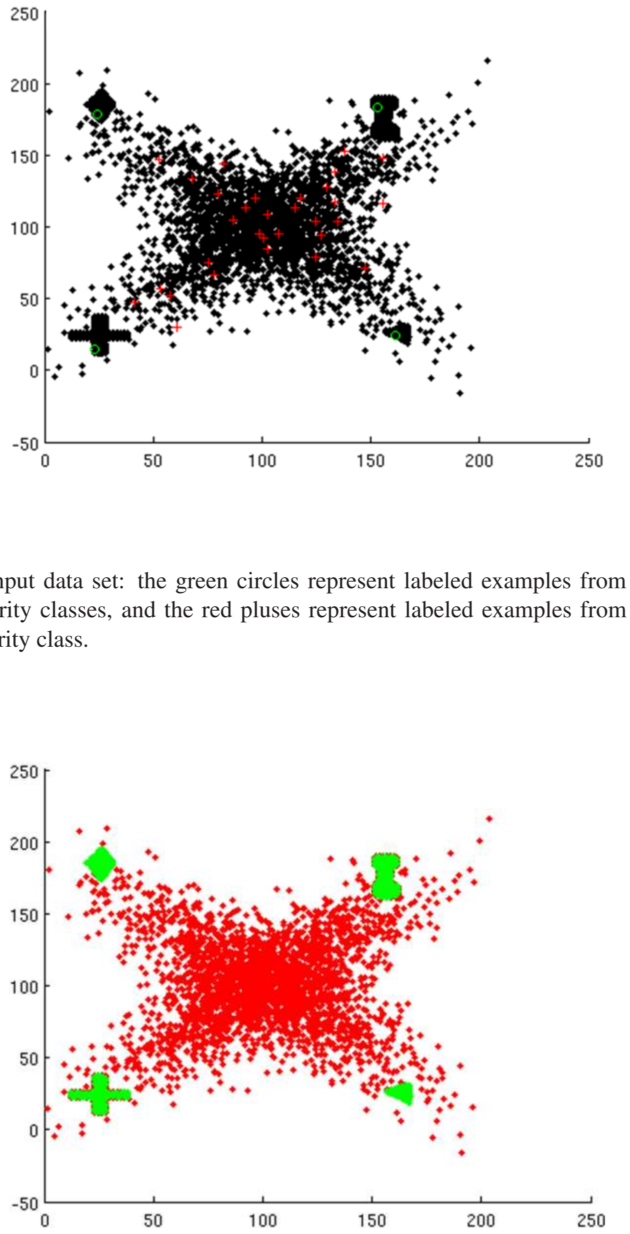
Compared with the original Alg. 10 which is designed for the input space, in the kernelized *RACH* algorithm, we only need to make the following changes. First, instead of directly calculating the center $b$, we keep a set of coefficients $\gamma_i$, $i = 1, \ldots, n$ such that $b = \sum_{i=1}^{n} \gamma_i x_i$. Therefore, Step 1 of Alg. 10 generates a set of initial coefficients for $b$, and Step 4 updates the coefficients of $b$ based on Equation 4.1. In this way, $b \cdot x = \sum_{i=1}^{n} \gamma_i K(x_i, x)$, and $\|b\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_i \gamma_j K(x_i, x_j)$, where $K(\cdot, \cdot)$ is the kernel function. Next, notice that Alg. 10 and Alg. 9 are dependent on the examples only through the inner products or distances, which can be replaced by the kernel calculation.

## 4.4 Experimental Results

In this section, we present some experimental results showing the effectiveness of *RACH*.

### 4.4.1 Synthetic Data Set

Fig. 4.1(a) shows a synthetic data set where the majority class has 3000 examples drawn from a Gaussian distribution, and the 4 minority classes correspond to 4 different shapes with 84, 150, 267, and 280 examples respectively. In this figure, the green circles represent labeled examples from the minority classes, and the red pluses represent labeled examples from the majority class. Here we construct 4 binary problems (the majority class vs. each minority class). Fig. 4.1(b) shows the classification result where the green dots represent the rare examples, and the red dots represent the majority class examples. From this figure, we can see that almost all the rare examples have been correctly identified except for a few points on the boundary.



(a) Input data set: the green circles represent labeled examples from the minority classes, and the red pluses represent labeled examples from the majority class.

(b) Classification results: the examples in green are classified as rare examples, and the examples in red are classified as majority class examples.

Figure 4.1: Rare category characterization on the synthetic data set. (Best viewed in color.)

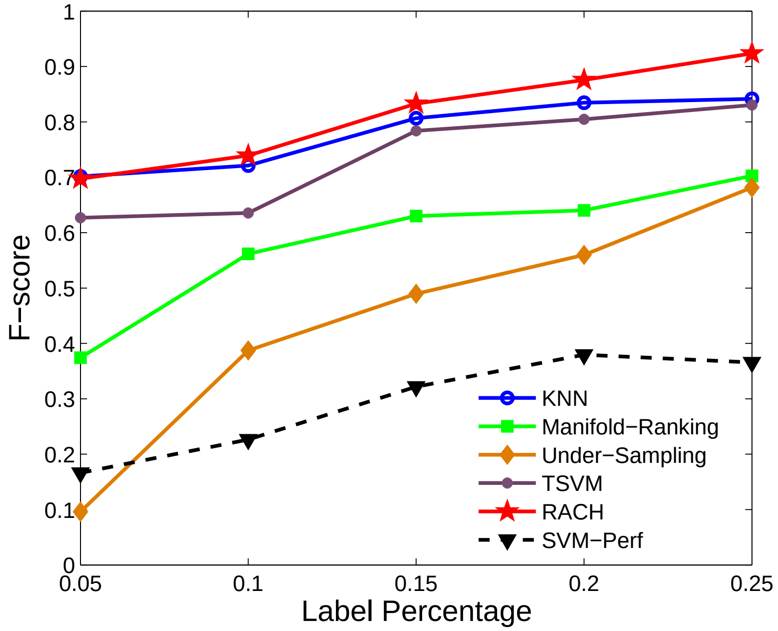
### 4.4.2 Real Data Sets

We also did experiments on 7 real data sets, which are summarized in Table 4.1. For each data set, we construct several binary problems consisting of one majority class and one minority class, and vary the percentage of labeled examples. For the sake of comparison, we also tested the following methods: (1) KNN (K-nearest neighbor); (2) Manifold-Ranking [Zhou *et al.*, 2003b]; (3) Under-Sampling; (4) TSVM [Joachims, 1999] with different costs for the examples from different classes; (5) SVM-Perf [Joachims, 2005]. We used the RBF kernel in *RACH*. All the parameters are tuned by cross validation. The comparison results in terms of the F-score (harmonic mean of precision and recall) of the minority class are shown in Fig. 4.2 to Fig. 4.8 where the x-axis is the percentage of labeled examples. Under each figure, the numbers outside the brackets are the class indices included in the binary problem, and the numbers inside the brackets are the number of examples from each class.

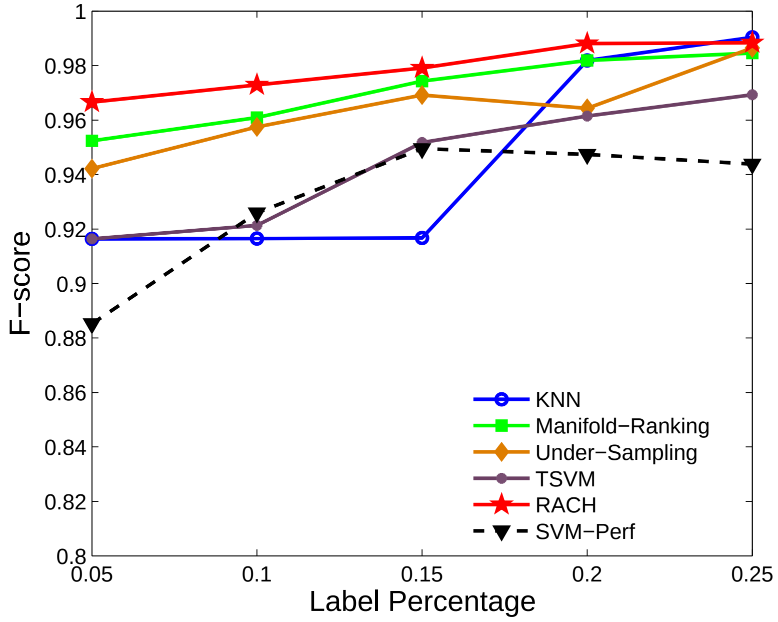
From these figures, we can see that the performance of *RACH* is consistently better than the other methods across all the data sets, especially when the percentage of labeled examples is small, which is the case we are most interested in. In particular, the performance of *RACH* is better than SVM-Perf in most cases, although the latter directly optimizes the F-score. This might be due to the fact that the objective function of SVM-Perf is only an upper bound of the training loss regularized by the $L_2$ norm of the weight vector. And also, SVM-Perfs is designed for the purpose of a general classification problem; and it might ignore the skewness and the compactness properties of the minority class. On the other hand, the performance of the other methods varies a lot across the different data sets. For example, in Fig. 4.2(a), the performance of KNN is only worse than *RACH*; whereas in Fig. 4.3(b), the performance of KNN is worse than TSVM, and as the percentage of labeled examples increases, KNN performs not as good as Under-Sampling.

Table 4.1: Summary of the data sets

| Data Set | Abalone | Ecoli | Glass | Yeast | Page Blocks | Shuttle | 20 Newsgroups |
|---|---|---|---|---|---|---|---|
| No. of Examples | 4177 | 336 | 214 | 1484 | 5473 | 4515 | 18774 |
| No. of Features | 7 | 7 | 9 | 8 | 10 | 9 | 61188 |

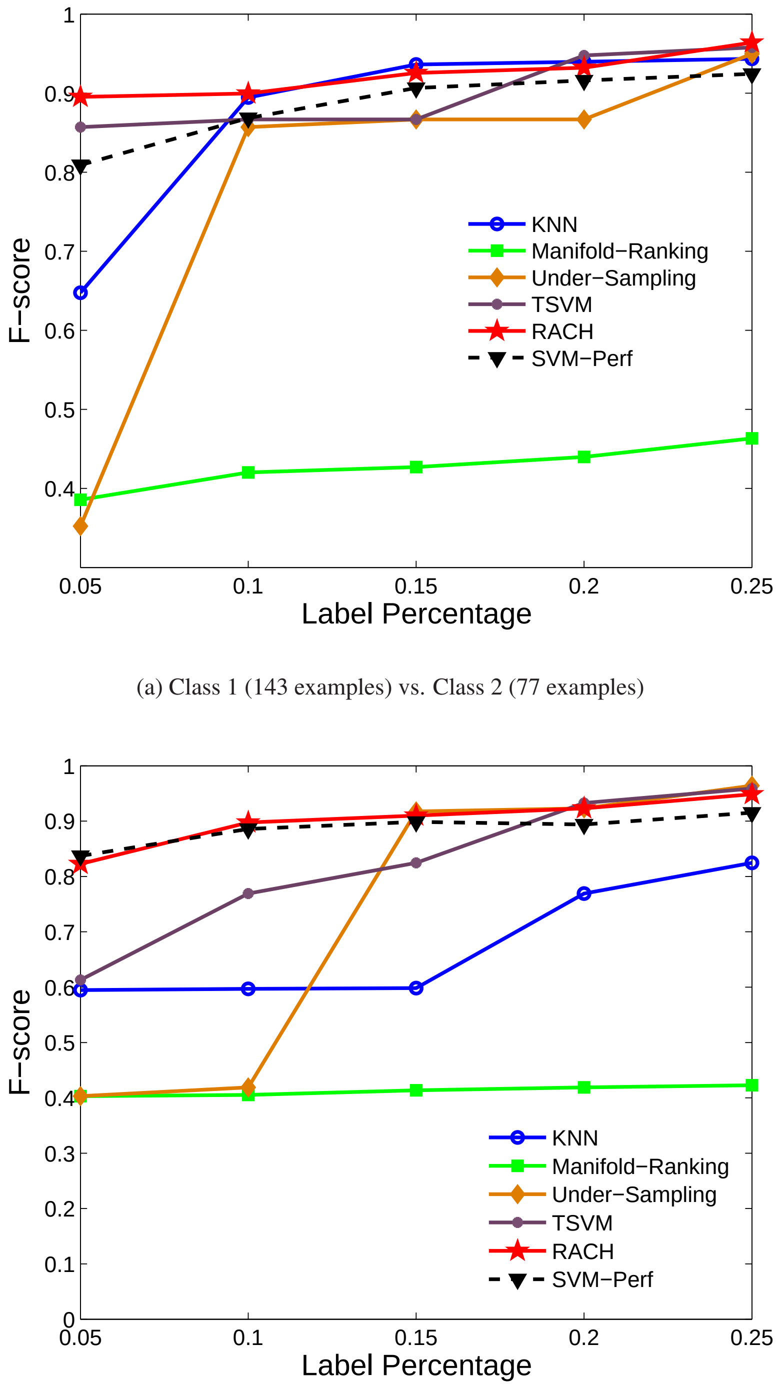


(a) Class 1 (689 examples) vs. Class 14 (67 examples)



(b) Class 2 (634 examples) vs. Class 4 (57 examples)

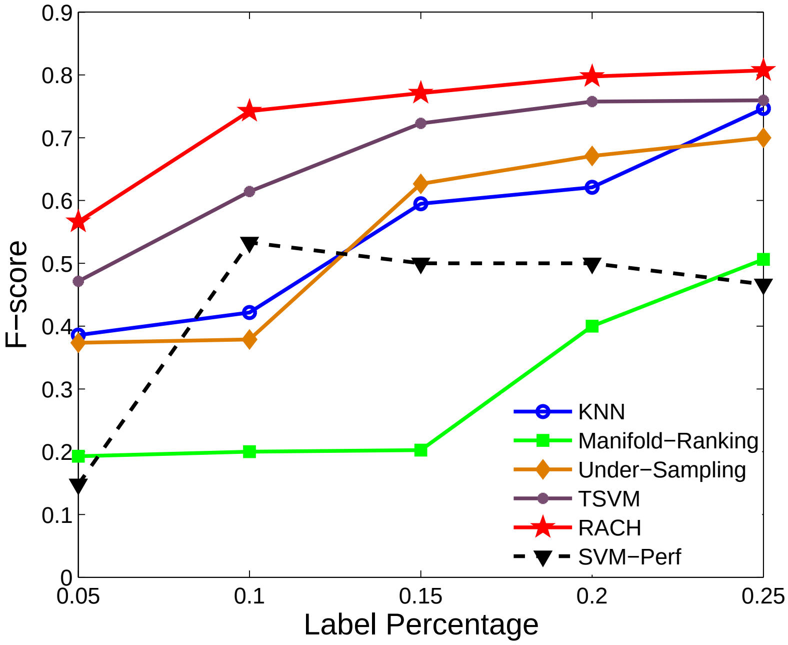
Figure 4.2: Results on Abalone data set



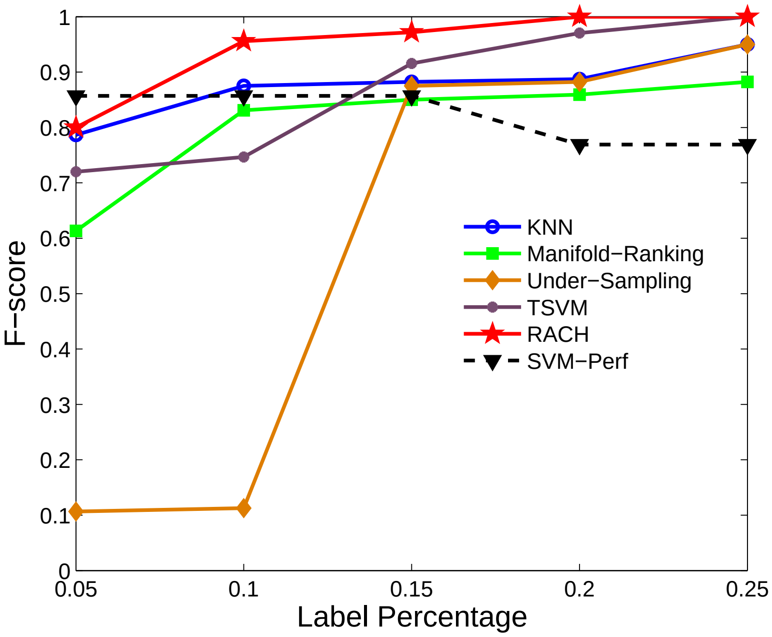
(a) Class 1 (143 examples) vs. Class 2 (77 examples)

(b) Class 2 (77 examples) vs. Class 3 (52 examples)

Figure 4.3: Results on Ecoli data set

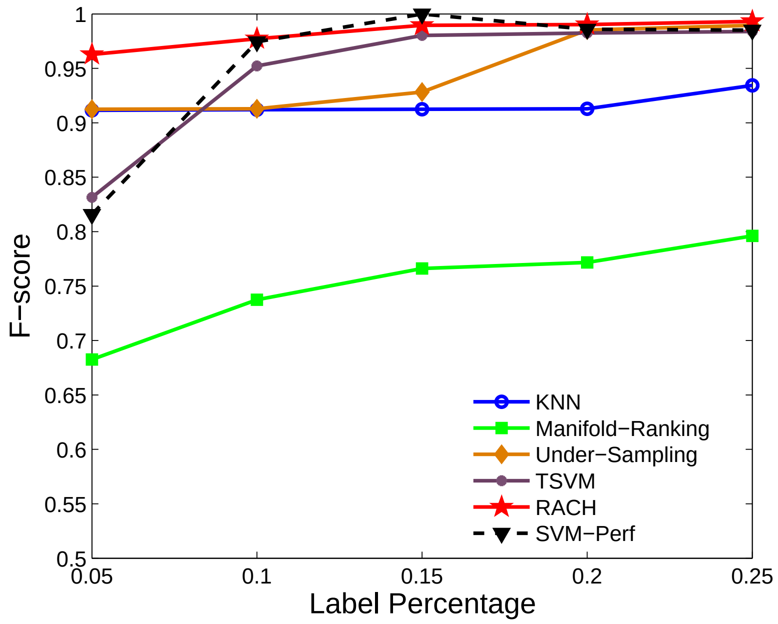


(a) Class 1 (70 examples) vs. Class 3 (17 examples)

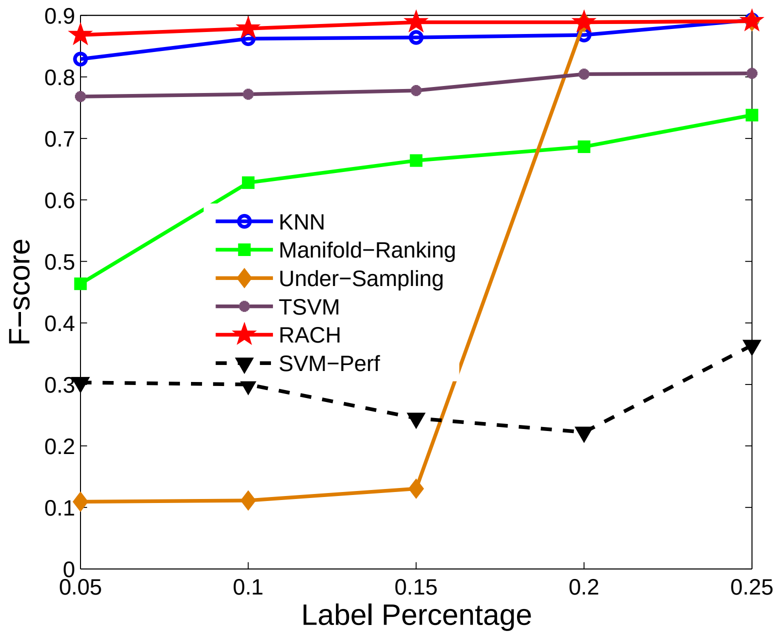


(b) Class 1 (70 examples) vs. Class 5 (9 examples)

Figure 4.4: Results on Glass data set

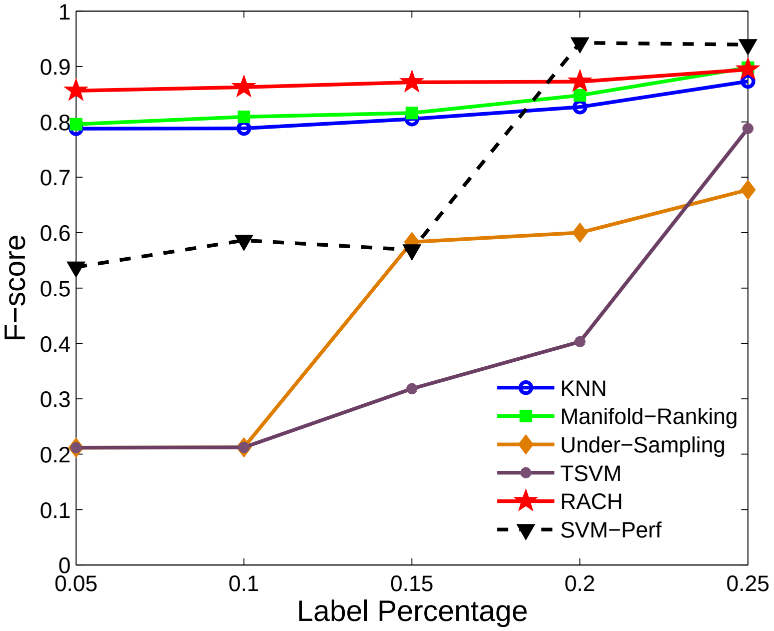


(a) Class 1 (463 examples) vs. Class 6 (44 examples)

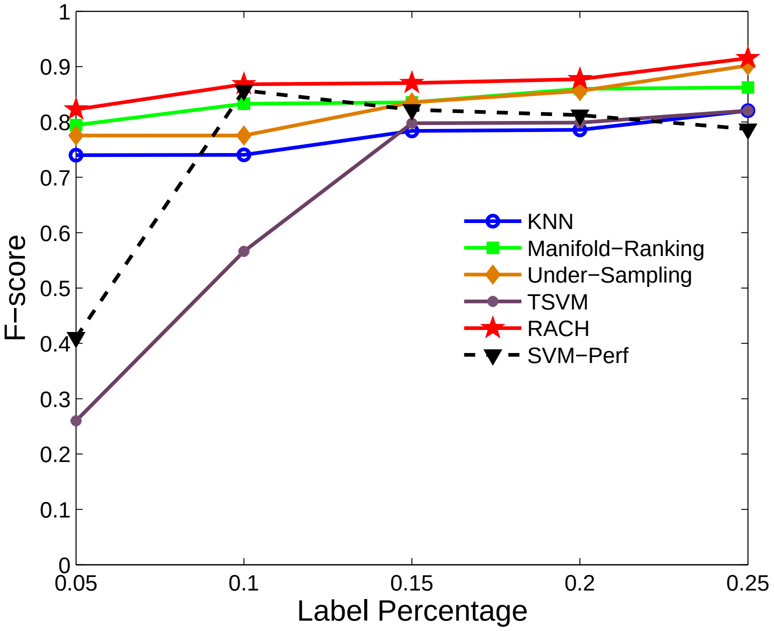


(b) Class 3 (244 examples) vs. Class 8 (30 examples)

Figure 4.5: Results on Yeast data set

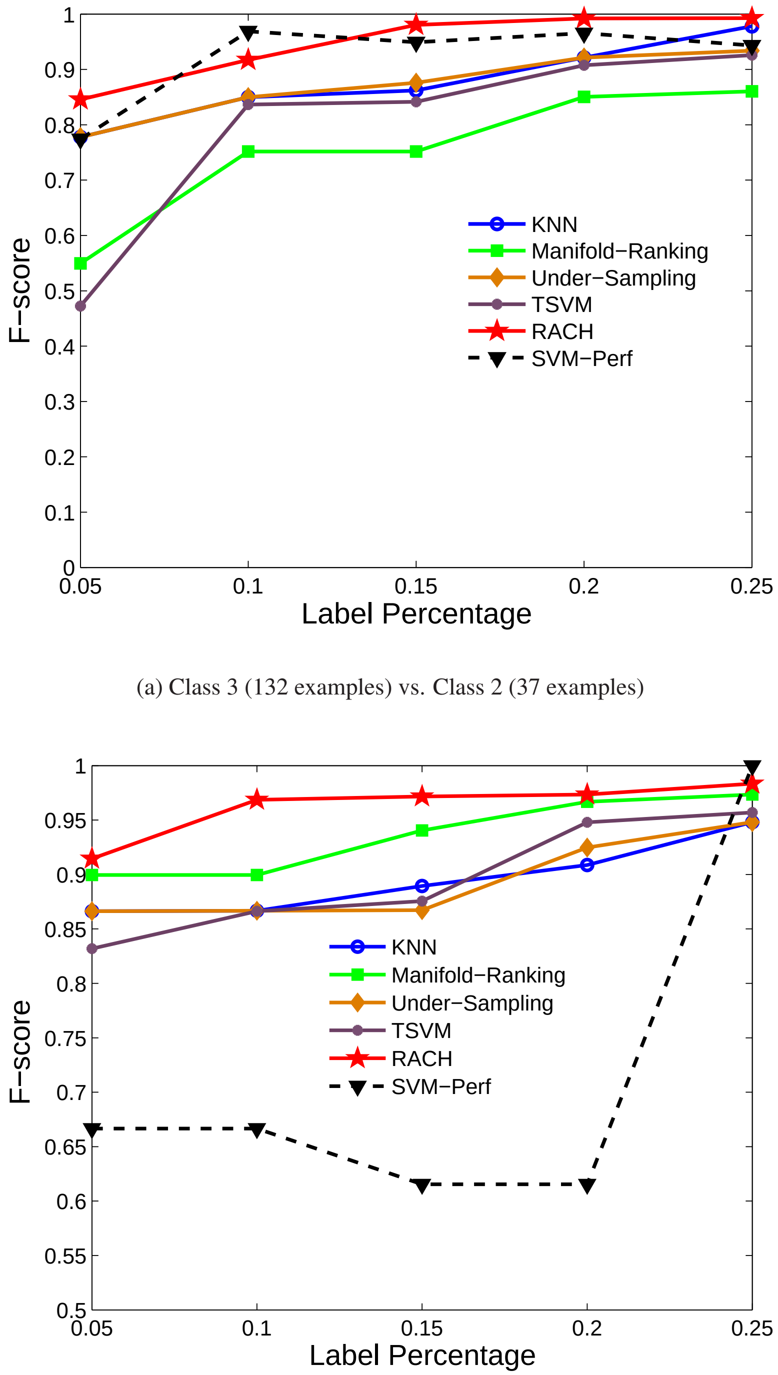


(a) Class 2 (329 examples) vs. Class 4 (88 examples)



(b) Class 2 (329 examples) vs. Class 5 (115 examples)

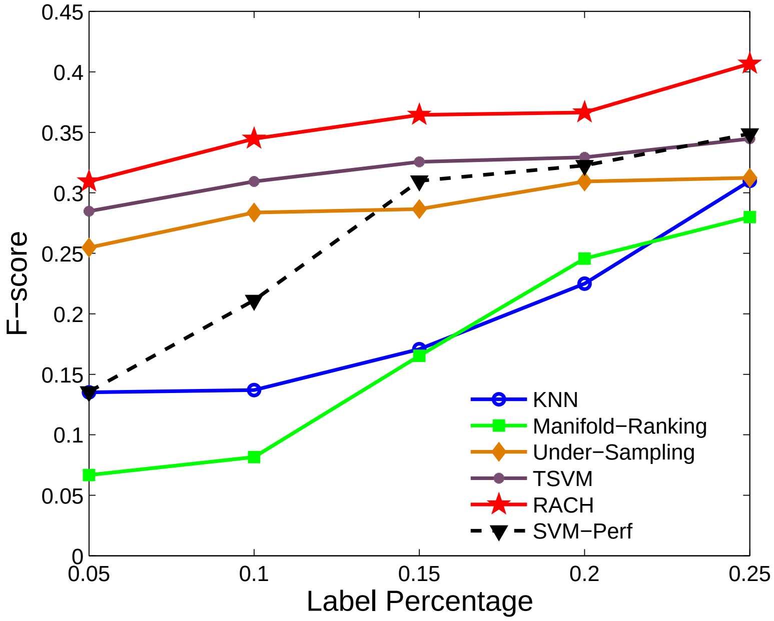
Figure 4.6: Results on Page Blocks data set



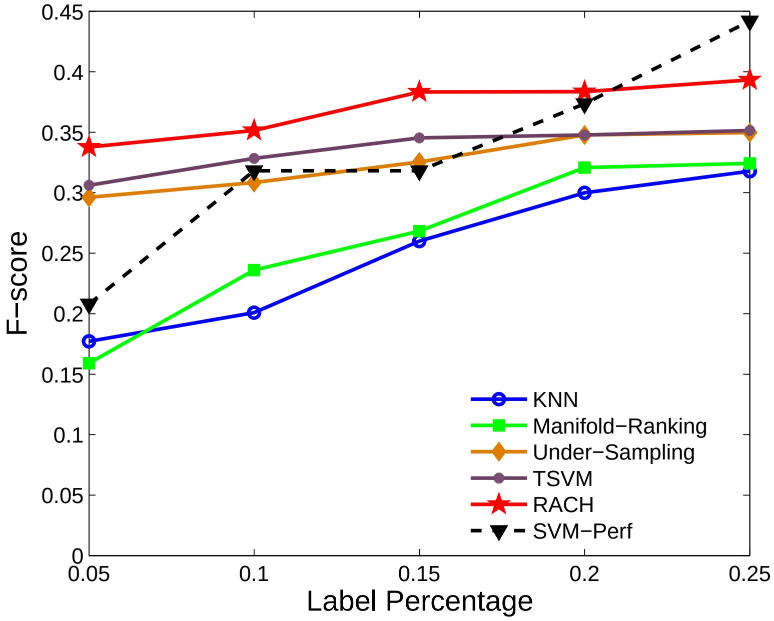
(a) Class 3 (132 examples) vs. Class 2 (37 examples)

(b) Class 2 (37 examples) vs. Class 7 (11 examples)

Figure 4.7: Results on Shuttle data set



(a) comp (4852 examples) vs. misc.forsale (964 examples)



(b) rec (3968 examples) vs. comp.os.ms-windows.misc (963 examples)

Figure 4.8: Results on 20 Newsgroups data set

## 4.5 Summary of Rare Category Characterization

In this chapter, we have discussed our work on rare category characterization. It follows the task of rare category detection, and makes use of labeled data from all the classes to find a compact representation for the minority classes. The goal is to identify all (or nearly all) the rare examples in the data set with high precision and recall. Different from the large amount of work on imbalanced classification, we only focus on the rare examples, and we are able to generate a compact representation for the minority classes for the sake of interpretability.

In our algorithm, the basic idea is to enclose the rare examples with a minimum-radius hyper-ball based on the clustering property of the minority classes. We formulate this idea as an optimization problem and present the effective optimization algorithm *RACH* to find its solution. In *RACH*, we repeatedly (1) convert the original problem into a convex optimization problem, and (2) solve it in its dual form by a projected subgradient method. Furthermore, we generalize *RACH* to the high-dimensional feature space induced by kernels. Experimental results demonstrate the effectiveness of the proposed *RACH*.

### 4.5.1 Limitations

The limitations of the current work on rare category characterization can be summarized as follows.

1. Interactions with domain experts. In *RACH*, the compact representations for the minority classes are used to help domain experts better understand the learning results. However, in our current work, we did not perform experiments to validate the effectiveness of these representations.
2. Application to data sets with multiple minority classes. The current version of *RACH* can only be applied to the binary settings, i.e., there is one majority class and one minority class in the data set. The generalization of *RACH* to multiple majority and minority classes in the data set may greatly improve its applicability to real world problems.
3. Computational issues. The proposed *RACH* algorithm does not scale well to very large data sets (see the analysis on its computational complexity in Subsection 4.2.3). This may be addressed by parallelism and simple yet effective heuristics, such as localized analysis.

# Chapter 5

# Unsupervised Rare Category Analysis

In this chapter, we focus on unsupervised rare category analysis, i.e., no label information is available in the learning process, and address the following two problems: (1) *rare category selection*, i.e., selecting a set of examples which are likely to come from the minority class; (2) *feature selection*, i.e., selecting the features that are relevant to identify the minority class.

The key observation is that the above two tasks are correlated with each other. On one hand, the analysis of the minority class examples helps us identify the relevant features; on the other hand, the identification of the relevant features is crucial to the selection of the minority class examples. Therefore, we propose to jointly deal with the two tasks so that they can benefit from each other. To this end, we formulate the problem as a well justified optimization framework, which co-selects the relevant features and the examples from the minority class. Furthermore, we design an effective search procedure based on augmented Lagrangian method. The basic idea is to alternatively find the relevant features and the minority class examples. Finally, we demonstrate the performance of the proposed method by experimental results on some benchmark data sets.

The main contributions of this chapter can be summarized as follows.

**Problem Definition.** To the best of our knowledge, we are the first to address the two important tasks in unsupervised rare category analysis; and we propose to jointly deal with them;

**Problem Formulation.** We design an optimization framework for the co-selection of features and examples, which is well justified theoretically;

**Search Procedure.** We develop an effective algorithm to solve the optimization problem which is based on augmented Lagrangian.

The rest of this chapter is organized as follows. In Section 5.1, we present the optimization framework with theoretical justification. Section 5.2 introduces the algorithm for solving the optimization problem. Experimental results are given in Section 5.3. Finally, we conclude this chapter with a brief summary in Section 5.4.

## 5.1 Optimization Framework

In this chapter, we focus on the binary case, i.e., one majority class and one minority class, and our goal is to (1) select a set of examples which are likely to come from the minority class, and (2) identify the features relevant to this minority class. In this section, we formulate this problem as an optimization framework, and provide some theoretical justification.

### 5.1.1 Additional Notation

In this chapter, we are dealing with the binary cases, i.e., $m = 2$. Therefore, $p_1 = 1 - p_2$. Furthermore, of the $d$ features, only $d_r$ features are relevant to the minority class. In other words, the examples from the minority class have very similar values on those features, and their values on the other features may be quite diverse. For the sake of simplicity, assume that the $d_r$ features are independent to each other. Therefore, the examples from the minority class are tightly clustered in the $d_r$-dimensional subspace spanned by the relevant features, which we call the relevant subspace.

Let $\mathbb{S}_{d_r}$ denote the set of all $d_r$-dimensional subspaces of $\mathbb{R}^d$, and let $S_{min}$ denote the relevant subspace, $S_{min} \in \mathbb{S}_{d_r}$. Let $f(x)$ denote the probability density function (pdf) of the data in $\mathbb{R}^d$, i.e., $f(x) = (1 - p_2)f_1(x) + p_2 f_2(x)$, where $f_1(x)$ and $f_2(x)$ are the pdf of the majority and minority classes in $\mathbb{R}^d$ respectively. Given feature subspace $S \in \mathbb{S}_{d_r}$ and $x \in \mathbb{R}^d$, let $x^{(S)}$ denote the projection of $x$ on $S$, and $f^{(S)}(x^{(S)})$, $f_1^{(S)}(x^{(S)})$ and $f_2^{(S)}(x^{(S)})$ denote the projection of $f(x)$, $f_1(x)$ and $f_2(x)$ on $S$ respectively.

To co-select the minority class examples and the relevant features, we define two vectors: $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^d$. Let $a_i$ and $b_j$ denote the $i^{\text{th}}$ and $j^{\text{th}}$ elements of $a$ and $b$ respectively. $a_i = 1$ if $x_i$ is from the minority class, and 0 otherwise; $b_j = 1$ if the $j^{\text{th}}$ feature is relevant to the minority class, and 0 otherwise.

### 5.1.2 Objective Function

Given the prior $p_2$ of the minority class and the number of relevant features $d_r$, we hope to find $np_2$ data points whose corresponding $a_i = 1$, and $d_r$ features whose corresponding $b_j = 1$. Intuitively, the $np_2$ points should form a compact cluster in the relevant subspace, and due to the characteristic of the minority class, this cluster should be more compact than any other $np_2$ data points in any $d_r$-dimensional subspace. To be more strict, we have the following optimization problem.

**Problem 5.1**

$$\begin{aligned} &\min \ f(a,b) = \frac{1}{np_2}\sum_{i=1}^{n}\sum_{k=1}^{n} a_i a_k (\sum_{j=1}^{d} b_j (x_i^j - x_k^j)^2) \\ &\text{s.t.,} \sum_{i=1}^{n} a_i = np_2, a_i = 0, 1 \\ &\qquad \sum_{j=1}^{d} b_j = d_r, b_j = 0, 1 \end{aligned}$$

In the objective function $f(a, b)$, $\sum_{j=1}^{d} b_j (x_i^j - x_k^j)^2$ is the squared distance between $x_i$ and $x_k$ in the subspace $S_b$ spanned by the features with non-zero $b_j$. This squared distance contributes to $f(a, b)$ if and only if both $a_i$ and $a_k$ are equal to 1. Given a set of $np_2$ points, define the set distance of every data point to be the sum of the squared distances between this point and all the points within this set. Therefore, by solving this optimization problem, we aim to find a set of $np_2$ points and $d_r$ features such that the average set distance of these points to this set in the corresponding subspace $S_b$ is the minimum.

Problem 5.1 can be easily applied to the case where either $a$ or $b$ is known, and we want to solve for the other vector. To be specific, if $a$ is known, i.e., we know the examples that belong to the minority class, and we want to find the $d_r$-dimensional subspace where the minority class can be best characterized, we can use the same objective function $f(a, b)$, and solve for $b$ using the minority class examples. Similarly, if $b$ is known, i.e., we know which features are relevant to the minority class, and we want to find the examples from the minority class, we can also use $f(a, b)$, and solve for $a$ in the subspace $S_b$ spanned by the relevant features.

### 5.1.3 Justification

The optimization problem we introduced in the last subsection is reasonable intuitively. Next, we look at it from a theoretical perspective.

$\forall S \in \mathbb{S}_{d_r}$, define function $\psi^S$ as follows. $\forall S \in \mathbb{S}_{d_r}$, $x \in \mathbb{R}^d$, let $\psi^S(x^{(S)}) = \min_{D_{np_2} \subset D, |D_{np_2}|=np_2} \frac{1}{np_2} \sum_{y \in D_{np_2}} \|x^{(S)} - y^{(S)}\|^2 = \frac{1}{np_2} \sum_{i=1}^{np_2} \|x^{(S)} - z^{(i)}_{x^{(S)}}\|^2$, where $z^{(i)}_{x^{(S)}}$ denotes the $i^{\text{th}}$ nearest neighbor of $x^{(S)}$ within $x_1^{(S)}, \ldots, x_n^{(S)}$, i.e., $\psi^S(x^{(S)})$ is the average squared distance between $x^{(S)}$ and its $np_2$ nearest neighbors. Furthermore, define function $\phi^S$ as follows. $\phi^S(x^{(S)}) = E(\psi^S(x^{(S)}))$. Here, the expectation is with respect to $z^{(i)}_{x^{(S)}}$, $i = 1, \ldots, np_2$.

Based on the above definitions, we have the following theorem.

**Theorem 7.** *If*

1. *In $S_{min}$, the support region of the minority class is within hyper-ball $B$ of radius $r$;*
2. *The support region of $f$ in any $d_r$-dimensional subspace is bounded, i.e., $\max_{S \in \mathbb{S}_{d_r}} \max_{x,y \in \mathbb{R}^d, f^{(S)}(x^{(S)})>0, f^{(S)}(y^{(S)})>0} \|x^{(S)} - y^{(S)}\| = \alpha < +\infty$;*
3. *The density of the majority class in hyper-ball $B$ is non-zero, i.e., $\min_{y \in \mathbb{R}^d, y^{(S_{min})} \in B} (1-p_2) f_1^{(S_{min})}(y^{(S_{min})}) = f_0 > 0$;*
4. *The function value of $\phi^S$ is big enough if the projection of the data point in the $d_r$-dimensional subspace $S$ is not within $B$, i.e., $\min_{S \in \mathbb{S}_{d_r}, x \in \mathbb{R}^d, x^{(S)} \notin B} \phi^S(x^{(S)}) - 4r^2 > \beta > 0$;*
5. *The number of examples is sufficiently large, i.e., $n \geq \max\{\frac{1}{2(V_B f_0)^2} \log \frac{2}{\delta}, \frac{\alpha^8}{4p_2^2 \beta^4} \log \frac{2C_d^{d_r}}{\delta}\}$, where $V_B$ is the volume of hyper-ball $B$, and $C_d^{d_r}$ is the number of $d$ choose $d_r$;*

*then with probability at least $1-\delta$, in the solution to Problem 5.1, the subspace $S_b$ spanned by the features with $b_j = 1$ is the relevant subspace $S_{min}$, and the data points with $a_i = 1$ are within $B$.*

*Proof.* The basic idea of the proof is to show that if the selected feature subspace $S_b$ is NOT $S_{min}$, or at least one point in the set of $np_2$ points is outside $B$ in $S_{min}$, we can always use $S_{min}$, and find another set of $np_2$ points such that all the points are within $B$, and its objective function is smaller than the original set. To do this, first, notice that according to condition (3), the expected proportion of data points falling inside $B$, $E(\frac{n_B}{n}) \geq p_2 + V_B f_0$, where $n_B$ denotes the number of points within $B$. Second, according to condition (2), $\forall x \in D$, $\Pr[0 \leq \|x^{(S)} - z^{(i)}_{x^{(s)}}\|^2 \leq \alpha^2] = 1$, $i = 1, \ldots, np_2$. Therefore,

$$
\begin{aligned}
&\Pr[\frac{n_B}{n} < p_2 \text{ or } \exists x \in D, \exists S \in \mathbb{S}_{d_r}, \text{ s.t., } \psi^S(x^{(S)}) < \phi^S(x^{(S)}) - \beta] \\
&\leq \Pr[\frac{n_B}{n} < p_2] + \Pr[\exists x \in D, \exists S \in \mathbb{S}_{d_r}, \text{ s.t., } \psi^S(x^{(S)}) < \phi^S(x^{(S)}) - \beta] \\
&\leq \Pr[\frac{n_B}{n} - E(\frac{n_B}{n}) < -V_B f_0] + nC_d^{d_r} \Pr[\psi^S(x^{(S)}) < \phi^S(x^{(S)}) - \beta] \\
&\leq \Pr[\frac{n_B}{n} - E(\frac{n_B}{n}) < -V_B f_0] + nC_d^{d_r} \int_{z^{(np_2+1)}_{x^{(S)}}} \Pr[\psi^S(x^{(S)}) < \phi^S(x^{(S)}) - \beta | z^{(np_2+1)}_{x^{(S)}}] d\Pr[z^{(np_2+1)}_{x^{(S)}}] \\
&\leq \exp(-2n(V_B f_0)^2) + nC_d^{d_r} \exp(-\frac{2np_2\beta^2}{\alpha^4})
\end{aligned}
$$

where $C_d^{d_r}$ is an upper bound on the number of subspaces in $\mathbb{S}_{d_r}$, and the last inequality is based on Hoeffding's inequality[1].

[1] Note that given $z^{(np_2+1)}_{x^{(S)}}$, $\psi^S(x^{(S)})$ can be seen as the average of $np_2$ independent items.

Let $\exp(-2n(V_B f_0)^2) \leq \frac{\delta}{2}$, and $nC_d^{d_r}\exp(-\frac{2np_2\beta^2}{\alpha^4}) \leq \frac{\delta}{2}$, we get $n \geq \frac{1}{2(V_B f_0)^2}\log\frac{2}{\delta}$, and $n \geq \frac{\alpha^8}{4p_2^2\beta^4}\log\frac{2C_d^{d_r}}{\delta}$. In other words, if the number of examples $n$ is sufficiently large, i.e.,

$$n \geq \max\{\frac{1}{2(V_B f_0)^2}\log\frac{2}{\delta}, \frac{\alpha^8}{4p_2^2\beta^4}\log\frac{2C_d^{d_r}}{\delta}\}$$

then with probability at least $1-\delta$, there are at least $np_2$ points within hyper-ball $B$, and $\forall x \in D$, $\forall S \in \mathbb{S}_{d_r}$, $\psi^S(x^{(S)}) \geq \phi^S(x^{(S)}) - \beta$. Furthermore, according to condition (4), $\forall x \in D$, $\forall S \in \mathbb{S}_{d_r}$, if $x^{(S)} \notin B$, $\psi^S(x^{(S)}) > 4r^2$.

Notice that $\forall a, \forall b$, $f(a,b) \geq \sum_{i:a_i=1}\psi^{S_b}(x_i^{(S_b)})$. On the other hand, if $S_b = S_{min}$, and the points with $a_i = 1$ are within $B$ in $S_{min}$, then $f(a,b) < 4np_2r^2$. This is because the squared distance between any two points within $B$ in $S_{min}$ is no bigger than $4r^2$.

Given $a$ and $b$, if $S_b$ is not $S_{min}$, we can design $a'$ and $b'$ in such a way that $S_{b'}$ is $S_{min}$, and the points that correspond to $a'_i = 1$ are within $B$ in $S_{min}$. We can always find such a vector $a'$ since we have shown that there are at least $np_2$ points within $B$. Therefore, $f(a,b) \geq \sum_{i:a_i=1}\psi^{S_b}(x_i^{(S_b)}) > 4np_2r^2 > f(a',b')$. On the other hand, if $S_b$ is $S_{min}$, but at least one point with $a_i = 1$ is outside $B$, we can design $a'$ and $b'$ in such a way that $b' = b$, and $a'$ replaces the points with $a_i = 1$ that are outside $B$ with some points within $B$ that are different from existing points in $a$. For the sake of simplicity, assume that only $x_t$ is outside $B$. Therefore, $f(a,b) = \frac{1}{np_2}\sum_{i\neq t}\sum_{k\neq t}a_ia_k\|x_i^{(S_{min})} - x_k^{(S_{min})}\|^2 + \frac{2}{np_2}\sum_{i=1}^{n}a_i\|x_i^{(S_{min})} - x_t^{(S_{min})}\|^2 \geq \frac{1}{np_2}\sum_{i\neq t}\sum_{k\neq t}a_ia_k\|x_i^{(S_{min})} - x_k^{(S_{min})}\|^2 + 2\psi^{S_{min}}(x_t^{(S_{min})}) > \frac{1}{np_2}\sum_{i\neq t}\sum_{k\neq t}a_ia_k\|x_i^{(S_{min})} - x_k^{(S_{min})}\|^2 + 8r^2 \geq f(a',b')$. The above derivation can be easily generalized to the case where more than one point with $a_i = 1$ are outside $B$. Therefore, in the solution to Problem 5.1, $S_b$ is the relevant subspace $S_{min}$, and the data points with $a_i = 1$ are within $B$. □

The conditions of Theorem 7 are straight-forward except condition (4). According to this condition, $\forall S \in \mathbb{S}_{d_r}$, if $x^{(S)} \notin B$ and $y^{(S_{min})} \in B$, $\phi^S(x^{(S)})$ is bigger than $\phi^{S_{min}}(y^{(S_{min})})$ by at least $\beta$ when there are at least $np_2$ points within $B$ in $S_{min}$. Therefore, this condition can be roughly interpreted as follows. The density around $x^{(S)}$ is smaller than the density around $y^{(S_{min})}$ such that the expected average squared distance between $x^{(S)}$ and its $np_2$ nearest neighbors is much larger than that between $y^{(S_{min})}$ and its $np_2$ neighbors. In this way, assuming the other conditions in Theorem 7 are also satisfied, with high probability, we can identify the relevant subspace and pick the examples within $B$ according to $a$.

It should be pointed out that if we want to select $np_2$ points from the minority class, picking them from hyper-ball $B$ is the best we can hope for. In this way, each selected example has a certain probability of coming from the minority class. On the other hand, if some selected points are outside $B$, their probability of coming from the minority class is 0.

## 5.2 Partial Augmented Lagrangian Method

In this section, we introduce the Partial Augmented Lagrangian Method (*PALM*) to effectively solve Problem 4.1. In our method, we alternate the optimization of $a$ and $b$, i.e., given the current estimate of $a$, we solve for $b$ that leads to the minimum value of $f(a,b)$; given the current estimate of $b$, we solve for $a$ that decreases the value of $f(a,b)$ as much as possible.

To be specific, $f(a,b)$ can be rewritten as $f(a,b) = \sum_{j=1}^{d}b_j\sum_{i=1}^{n}\sum_{k=1}^{n}\frac{1}{np_2}a_ia_k(x_i^j - x_k^j)^2$. Therefore, given $a$, we can solve for $b$ as follows. For each feature $j$, calculate its score $s_j^a = \frac{1}{np_2}\sum_{i=1}^{n}\sum_{k=1}^{n}a_ia_k(x_i^j - x_k^j)^2$. Then find the $d_r$ features with the smallest scores, and set their corresponding $b_j = 1$. It is easy to show that this vector $b$ minimizes $f(a, b)$ given $a$. On the other hand, given $b$, solving for $a$ is not straightforward, since $f(a, b)$ is not a convex function of $a$. In this chapter, we first relax the constraints on $a$: instead of requiring that $a_i$ be binary, we require that $a_i \in [0, 1]$, i.e., we solve the following optimization problem of $a$:

**Problem 5.2**

$$\min\ g_b(a) = \frac{1}{np_2}\sum_{i=1}^{n}\sum_{k=1}^{n} a_i a_k (\sum_{j=1}^{d} b_j (x_i^j - x_k^j)^2)$$

$$\text{s.t.}, \sum_{i=1}^{n} a_i = np_2, a_i \in [0, 1]$$

Next we use augmented Lagrangian method [Nocedal & Wright, 1999] to solve Problem 5.2 in an iterative way. The reason for using augmented Lagrangian method is the following: it is a combination of Lagrangian and quadratic penalty methods; the addition of the penalty terms to the Lagrangian function does not alter the stationary point of the Lagrangian function, and can help damp oscillations and improve convergence. Here, we define the following augmented Lagrangian function

$$\mathcal{L}_A(a, \lambda, \sigma) = \frac{1}{np_2}\sum_{i=1}^{n}\sum_{k=1}^{n} a_i a_k (\sum_{j=1}^{d} b_j (x_i^j - x_k^j)^2) - \sum_{i=1}^{2n+1} \lambda_i d_i(a) + \frac{\sigma}{2}\sum_{i=1}^{2n+1} d_i^2(a) \tag{5.1}$$

where $\lambda_i$, $i = 1, \ldots, 2n + 1$ are the Lagrange multipliers, $\sigma$ is a positive parameter, and $d_i(a)$, $i = 1, \ldots, 2n + 1$ are a set of functions defined as follows.

$$d_i(a) = \begin{cases} c_i(a) & \text{if } i \le 1 \text{ or } c_i(a) \le \frac{\lambda_i}{\sigma} \\ \frac{\lambda_i}{\sigma} & \text{otherwise} \end{cases}$$

$$c_1(a) = \sum_{i=1}^{n} a_i - np_2 = 0$$

$$c_{j+1}(a) = a_j \ge 0,\ 1 \le j \le n$$

$$c_{k+n+1}(a) = 1 - a_k \ge 0,\ 1 \le k \le n$$

Here $c_i(a)$, $i = 1, \ldots, 2n + 1$, denote the original constraints on $a$, both equality and inequality, and $d_i(a)$ are truncated versions of $c_i(a)$, i.e., $d_i(a)$ is equal to $c_i(a)$ if and only if the corresponding constraint is active or near-active; it is fixed at $\frac{\lambda_i}{\sigma}$ otherwise.

We minimize $\mathcal{L}_A(a, \lambda, \sigma)$ based on Algorithm 4.20 in [Madsen *et al.*, 2004]. Putting together the optimization of $a$ and $b$, we have the Partial Augmented Lagrangian Method, which is presented in Alg. 11.

The algorithm works as follows. Given the initial values $\lambda_0$ and $\sigma_0$ of $\lambda$ and $\sigma$, and the maximum number of iteration steps $step_{\max}$, it will output vectors $a$ and $b$ that correspond to a local minimum of $f(a, b)$. In Step 1, we initialize $a$ and $b$. Next, in Step 2, we assign $\lambda$ and $\sigma$ to their initial values, and calculate $K_{prev}$, which is the maximum absolute value of all the $d_i(a)$ functions, $i = 1, \ldots, 2n + 1$. Then Step 4 to Step 16 are repeated $step_{\max}$ times. In Step 4, we minimize the augmented Lagrangian function with respect to $a$, given the current estimates of $\lambda$, $\sigma$, and $b$. To be specific, we use gradient descent to update $a$, and gradually decrease the step size until convergence. Once we have obtained an updated estimate of $a$, calculate $K$, which is the maximum absolute value of the current $d_i(a)$ functions. If the value of $K$ is less than a half of $K_{prev}$, then we update the Lagrange multipliers using the formula in Step 7, which is called the steepest ascent formula in [Madsen *et al.*, 2004]. Furthermore, we update $K_{prev}$ using the smaller value of $K$ and $K_{prev}$. Otherwise, if the value $K$ is bigger than a half of $K_{prev}$, we double the value of $\sigma$. Next, we update the value of $b$ based on the current estimate of $a$. To be specific, for each feature, we calculate its score based on the formula in Step 14. Then in Step 16, we pick $d_r$ features with the smallest scores, and set the corresponding $b_j$ to 1, which minimizes $f(a, b)$ given $a$. In our experiments, the algorithm always converges around 20 iteration steps, so we set $step_{\max} = 30$.

**Algorithm 11** Partial Augmented Lagrangian Method (*PALM*)

```
Input: Initial values of $\lambda$ and $\sigma$: $\lambda_0$ and $\sigma_0$, $step_{\max}$
Output: $a$ and $b$
 1: Initialize $a$ and $b$
 2: $\lambda = \lambda_0$, $\sigma = \sigma_0$, $K_{prev} = \|d(a)\|_\infty$
 3: for $step = 1$ to $step_{\max}$ do
 4:     $a := \arg\min_a \mathcal{L}_A(a, \lambda, \sigma)$, $K := \|d(a)\|_\infty$
 5:     if $K \leq \frac{K_{prev}}{2}$ then
 6:         for $i = 1$ to $2n + 1$ do
 7:             $\lambda_i := \lambda_i - \sigma d_i(a)$
 8:         end for
 9:         $K_{prev} := \min(K, K_{prev})$
10:     else
11:         $\sigma := 2 \times \sigma$
12:     end if
13:     for $j = 1$ to $d$ do
14:         Calculate the score for the $j^{\text{th}}$ feature $s_j^a = \frac{1}{np_2} \sum_{i=1}^n \sum_{k=1}^n a_i a_k (x_i^j - x_k^j)^2$
15:     end for
16:     Pick $d_r$ features with the smallest scores, and set their corresponding $b_j$ to 1
17: end for
```

Notice that the vectors $a$ and $b$ generated by *PALM* correspond to a local minimum of $f(a, b)$. To improve its performance, we can run *PALM* with different initializations of $a$ and $b$ in Step 1 of Alg. 11, and pick the best values of $a$ and $b$ that correspond to the smallest $f(a, b)$.

The vectors $a$ and $b$ can be interpreted as follows. For $b$, its $d_r$ non-zero elements correspond to the relevant features. For $a$, ideally the minority class examples should correspond to $a_i = 1$. However, this may not be the case in practice. Therefore, we rank the elements of $a$ from large to small, and hope to find all the minority class examples from the top of the ranked list.

## 5.3 Experimental Results

In this section, we demonstrate the performance of *PALM* from the following perspectives: (1) the quality of rare category selection; (2) the quality of feature selection; (3) the benefit of co-selecting features and instances.

In our experiments, we retrieve the minority class examples from the ranked list generated by different methods, and use the following performance measures: (1) the precision-scope curve, i.e., the percentage of the minority class examples when a certain number of examples are retrieved, such as $10\% \times np_2, \ldots, 100\% \times np_2$; (2) the recall-scope curve, i.e., the percentage of the minority class examples when a certain number of MINORITY class examples are retrieved, such as $10\% \times np_2, \ldots, 100\% \times np_2$.

### 5.3.1 Synthetic Data Sets

To demonstrate the performance of *PALM*, we first use a simple synthetic data set shown in Fig. 5.1. In this figure, there are 1000 examples from the majority class, denoted as blue dots, which are uniformly distributed in the feature space, and only 10 examples from the minority class, denoted as red balls, whose features on $Z$ are uniformly distributed. Of the 3 features, only 2 features ($X$ and $Y$) are relevant to the minority class, i.e., the minority class examples have very similar values on these features; and 1 feature ($Z$) is irrelevant to the minority class, i.e., the minority class examples spread out on this feature. Using *PALM*, given the number of minority class examples and the number of relevant features, we are able to identify the relevant features, with the corresponding $b_j = 1$. Of the 10 examples with the largest $a_i$ values, 9 examples are from the minority class, and the remaining minority class example has the 11th largest $a_i$ value.

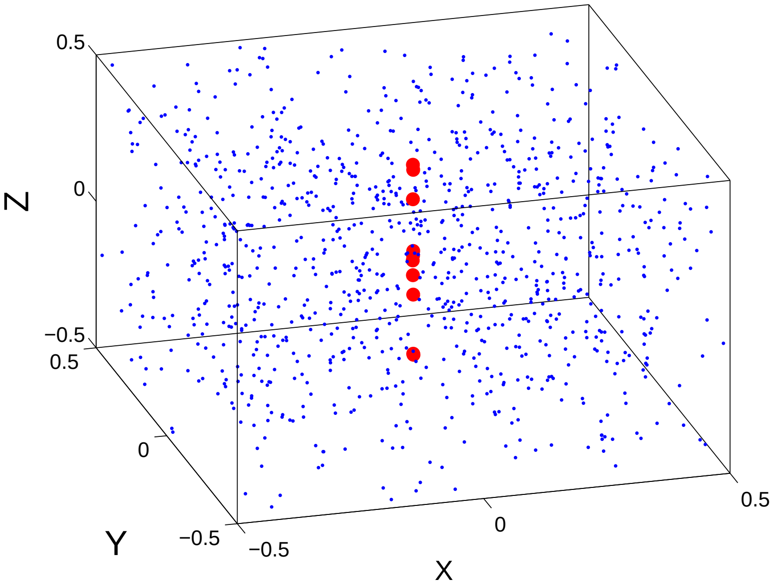
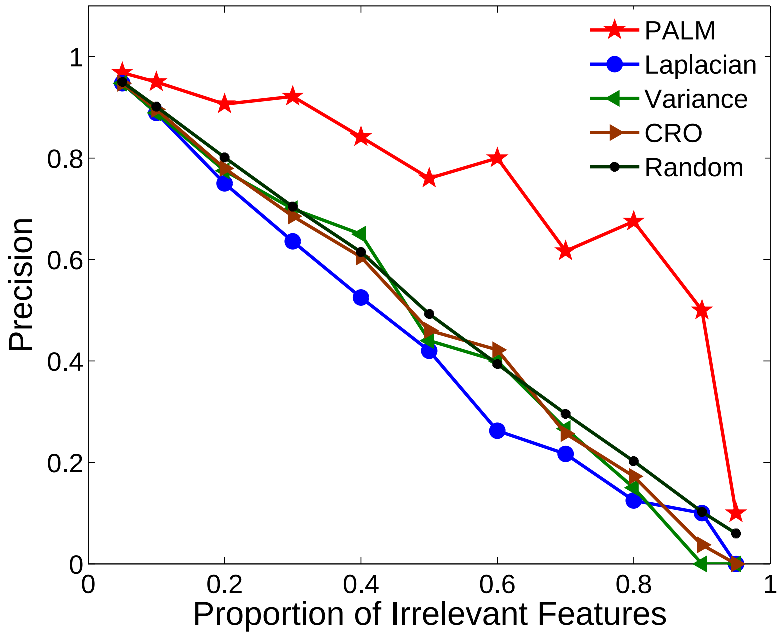
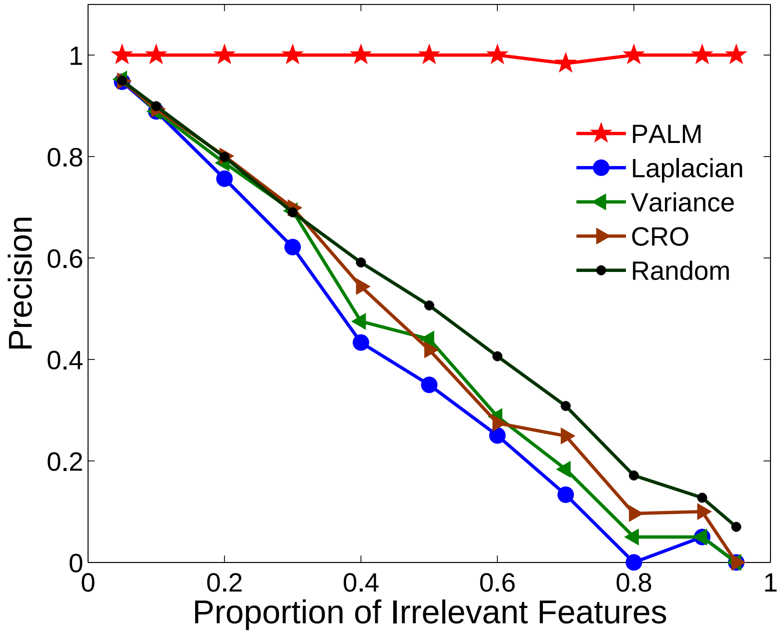


Figure 5.1: Synthetic data set: there are 1000 examples from the majority class, denoted as blue dots, and only 10 examples from the minority class, denoted as red balls. (Best viewed in color)

Next we test the precision of the selected features of *PALM* using synthetic data sets with different prior $p_2$. Fig. 5.2, Fig. 5.3, and Fig. 5.4 show the comparison results of *PALM* with Laplacian score method [He *et al.*, 2005a], feature variance method (selecting the features with the largest variance), CRO [Kim & Choi, 2007], and random sampling. The x-axis is the proportion of irrelevant features, and the y-axis is the precision of the selected features. From these results, we can see that *PALM* is much better than the other 4 methods especially when the prior $p_2$ is small. As for Laplacian score method, although it is comparable with *PALM* for large $p_2$, its performance quickly deteriorates as $p_2$ decreases (e.g., Fig. 5.2a and b), which is the case we are interested in for rare category analysis.

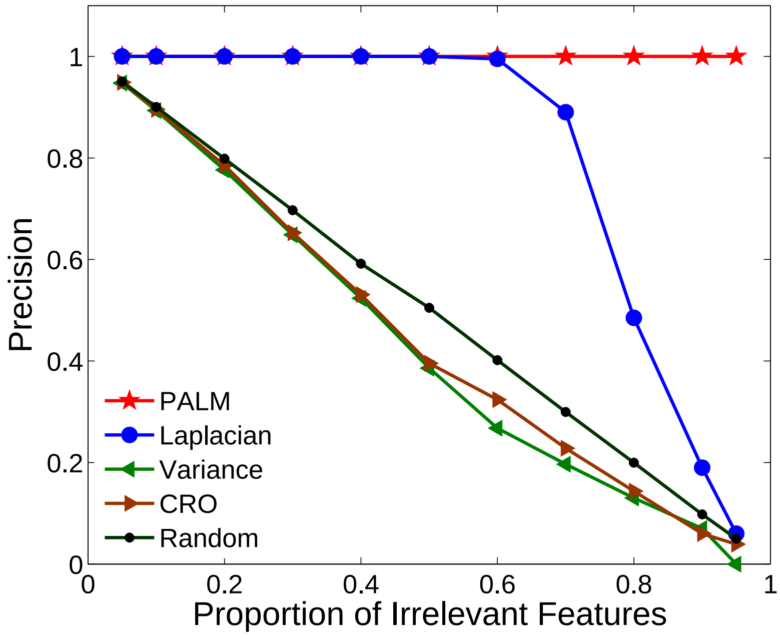


(a) $p_2 = 0.01$

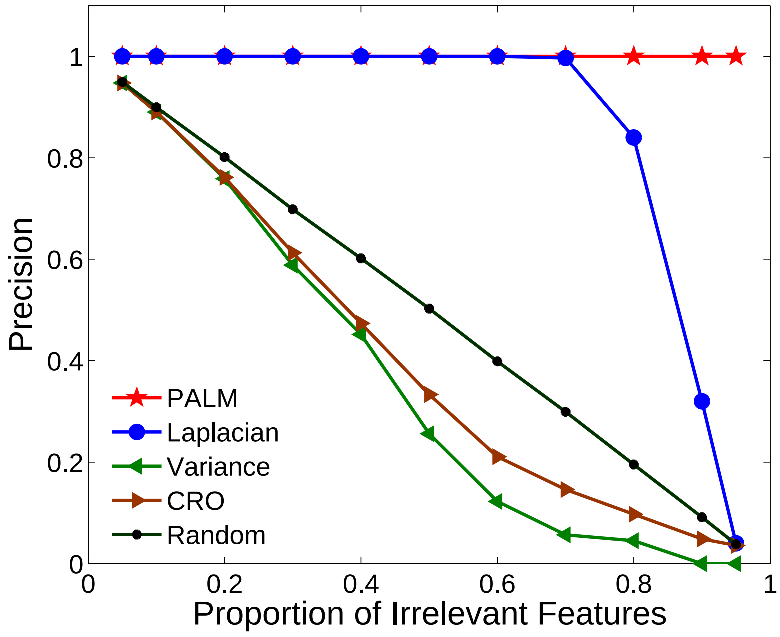


(b) $p_2 = 0.015$

Figure 5.2: Precision of selected features on synthetic data sets (part 1).

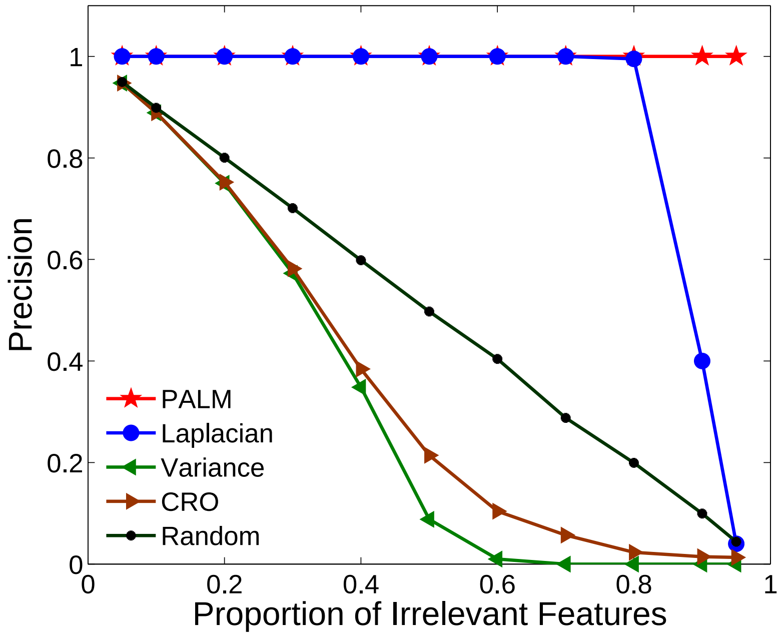


(a) $p_2 = 0.02$

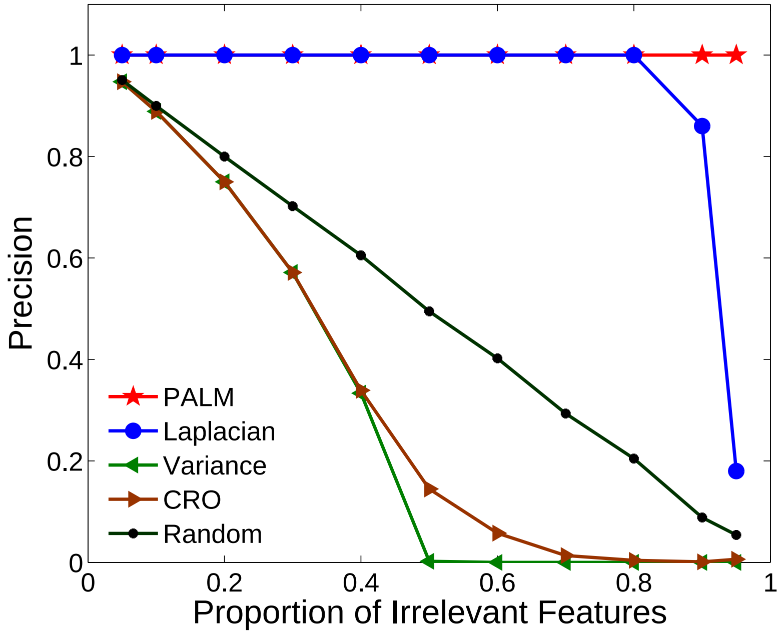


(b) $p_2 = 0.05$

Figure 5.3: Precision of selected features on synthetic data sets (part 2).



(a) $p_2 = 0.1$



(b) $p_2 = 0.2$

Figure 5.4: Precision of selected features on synthetic data sets (part 3).

### 5.3.2 Real Data Sets

In this subsection, we test the performance of *PALM* on rare category selection. To the best of our knowledge, there are no existing methods for this task. Therefore, we have designed the following methods for the sake of comparison.

1. Random: randomly rank all the examples, and select the first $np_2$ points from the ranked list as the minority class examples.
2. NNDB-based: calculate the score of each example using NNDB [He & Carbonell, 2007]. Note that no feedback from the labeling oracle is available, so the scores are not updated.
3. Interleave-based: calculate the score of each example using the Interleave principle [Pelleg & Moore, 2004]. Similar as the NNDB-based method, the scores of the examples are not updated in this method.
4. *PALM*-full: assume that all the features are relevant to the minority class, i.e., $b_j = 1$, $j = 1, \ldots, d$, and run *PALM* with $d_r = d$.

Note that NNDB-based method and Interleave-based method are both derived from rare category detection methods.

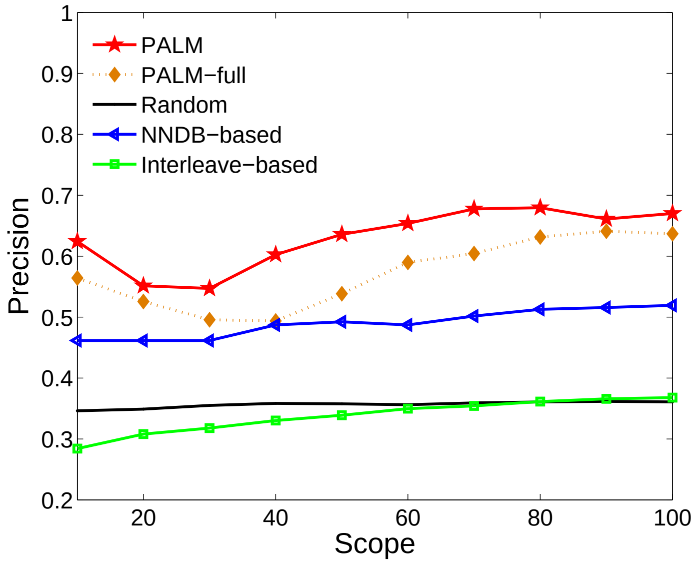
Here we use 4 real data sets, which are summarized in Table 5.1. In this chapter, we focus on binary problems, i.e., there is only one majority class and one minority class in the data set. Therefore, for each data set, we construct several subproblems as follows. We combine the examples from two different classes into a smaller binary data set, using the larger class as the majority class, the smaller class as the minority class, and test the different methods on these binary data sets. For *PALM*, we tune the number of relevant features $d_r$ without any label information. For each data set, we present the results on 2 binary subproblems, which are shown in Fig. 5.5 to Fig. 5.12. On the other binary subproblems, similar results are observed and therefore omitted for space. In these figures, the left figure shows precision vs. scope, and the right figure shows recall vs. scope.

On all the data sets, *PALM* performs the best: the precision and recall sometimes reach 100%, such as Fig. 5.10 and Fig. 5.11. As for the other methods (Interleave-based, NNDB-based, and *PALM*-full), their performance depends on different data sets, and none of them is consistently better than Random. Comparing with Random, Interleave-based, and NNDB-based, we can see that *PALM* does a better job at selecting the minority class examples; comparing with *PALM*-full, we can see that the features selected by *PALM* indeed help improve the performance of rare category selection.

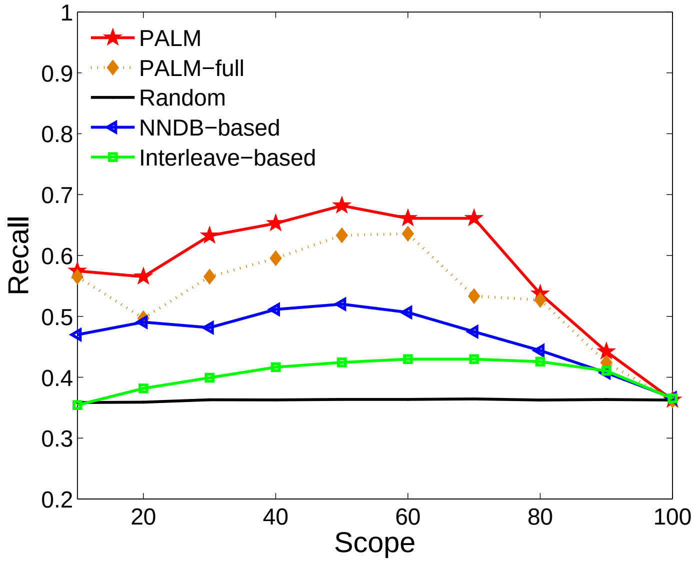
Notice that in some figures (Fig. 5.5b, Fig. 5.6b, Fig. 5.7b, Fig. 5.9b, and Fig. 5.10b), at the end of the recall curves, the different methods seem to overlap with each other. This is because with no supervision, it is sometimes difficult to retrieve all the examples from the minority class, and the last example from the minority class tends to appear towards the end of the ranked list. Therefore, the recall value at $100\% np_2$ is often close to the prior of the minority class in the data set.

Table 5.1: Properties of the data sets [Asuncion & Newman, 2007] used.

| Data Set | $n$ | $d$ | Largest Class | Smallest Class |
|---|---|---|---|---|
| Ecoli | 336 | 7 | 42.56% | 2.68% |
| Glass | 214 | 9 | 35.51% | 4.21% |
| Abalone | 4177 | 7 | 16.50% | 0.34% |
| Yeast | 1484 | 8 | 31.20% | 1.68% |

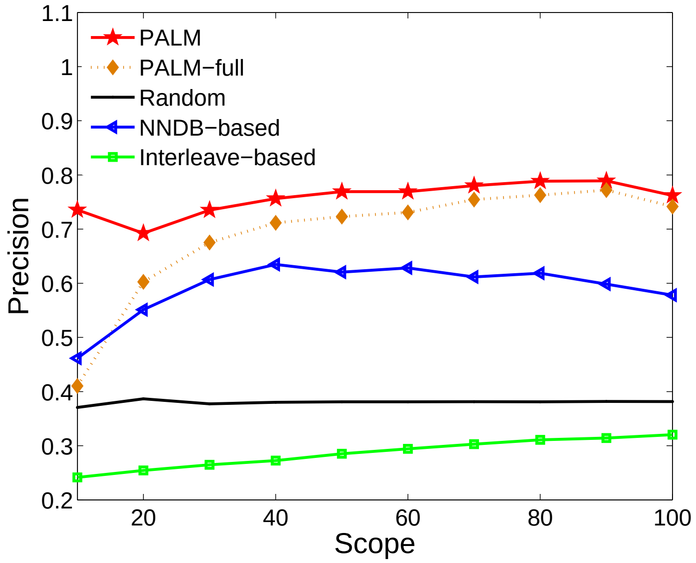


(a)

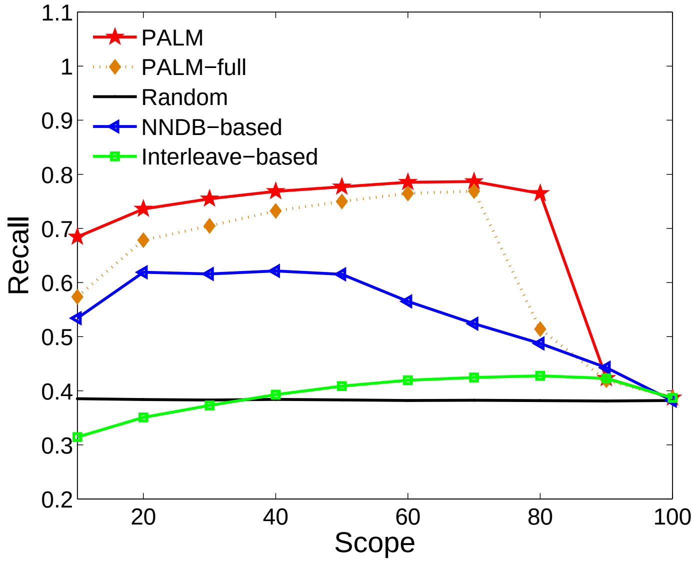


(b)

Figure 5.5: Abalone data set: class 1 vs. class 7, $p_2 = 0.362$, 4 features selected by *PALM*.

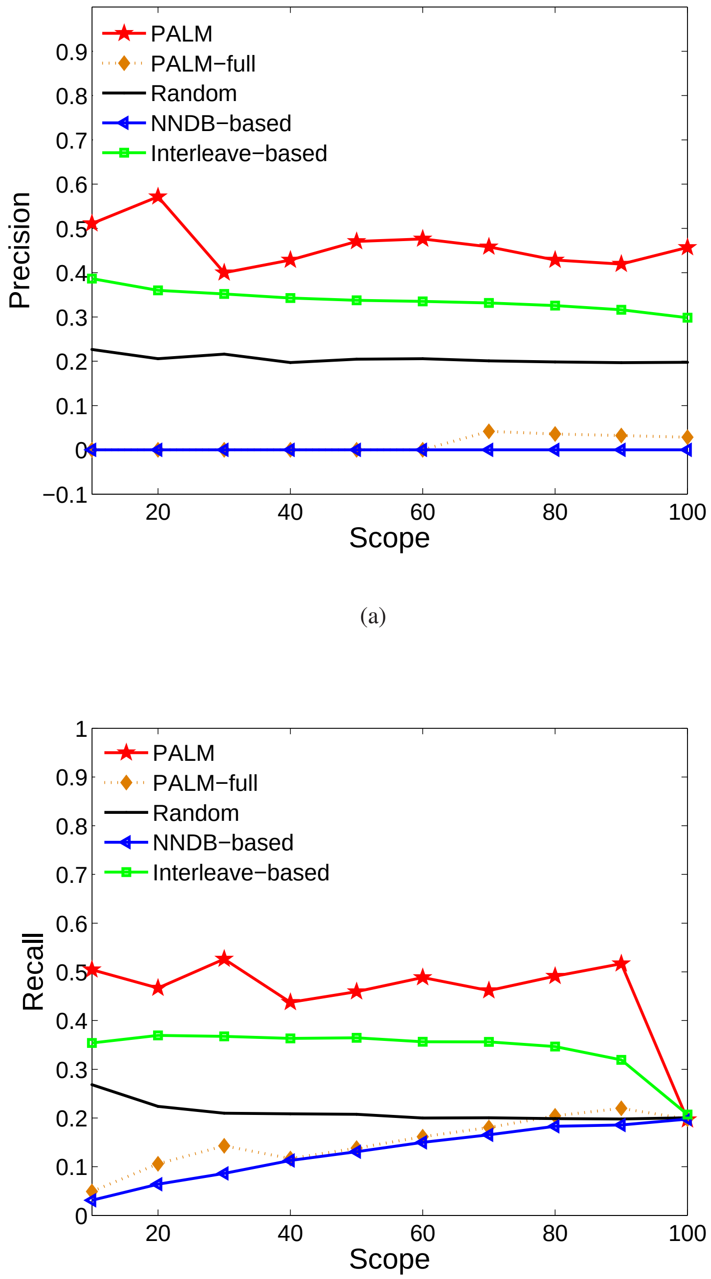


(a)



(b)

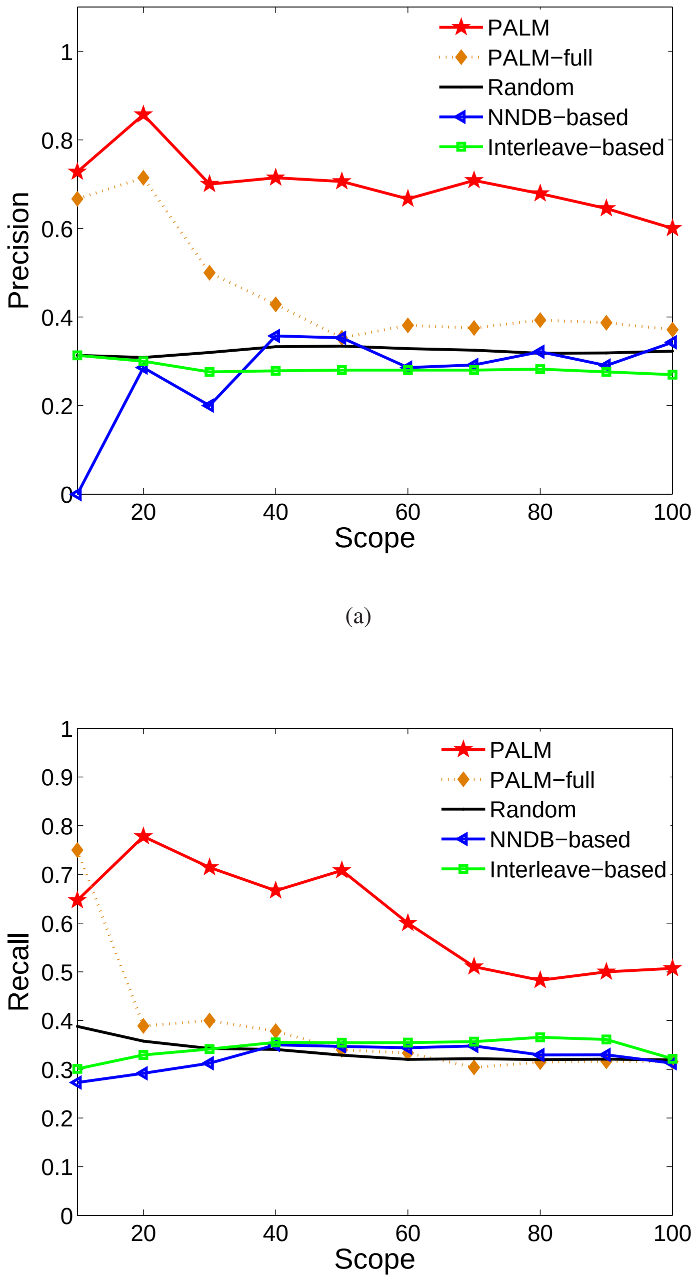
Figure 5.6: Abalone data set: class 2 vs. class 7, $p_2 = 0.381$, 4 features selected by *PALM*.



(a)

(b)

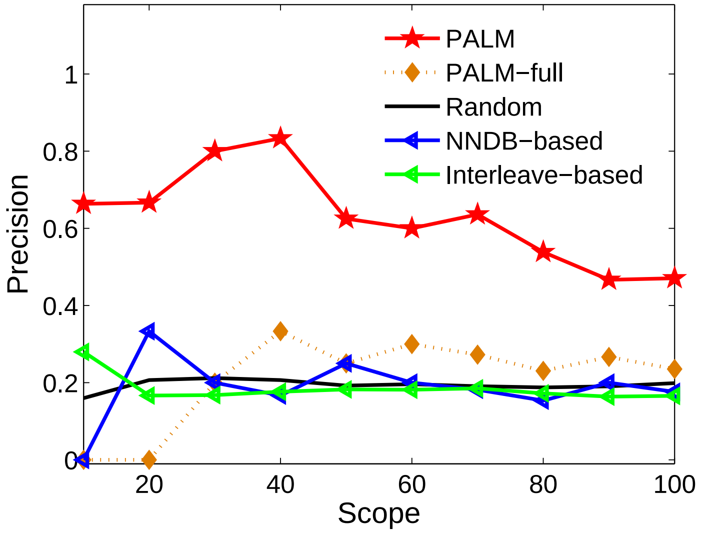
Figure 5.7: Ecoli data set: class 1 vs. class 4, $p_2 = 0.197$, 3 features selected by *PALM*.



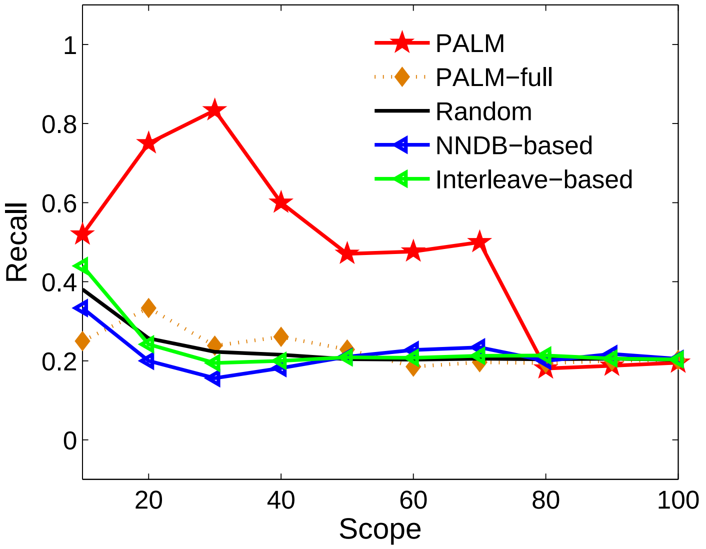
(a)

(b)

Figure 5.8: Ecoli data set: class 2 vs. class 4, $p_2 = 0.313$, 4 features selected by *PALM*.

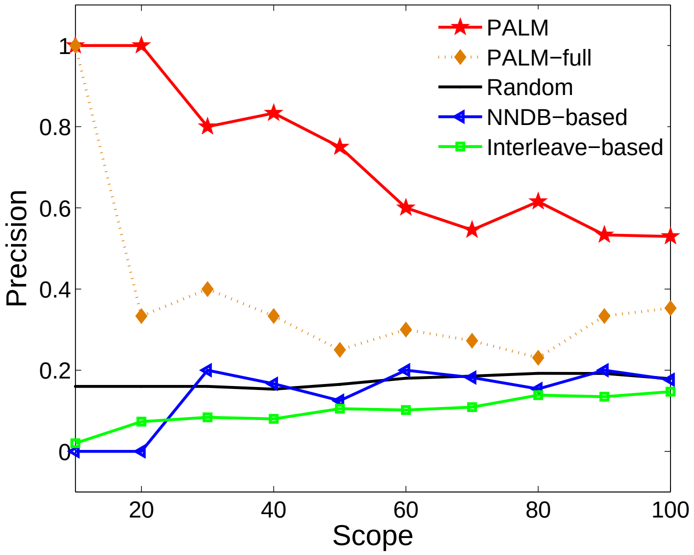


(a)

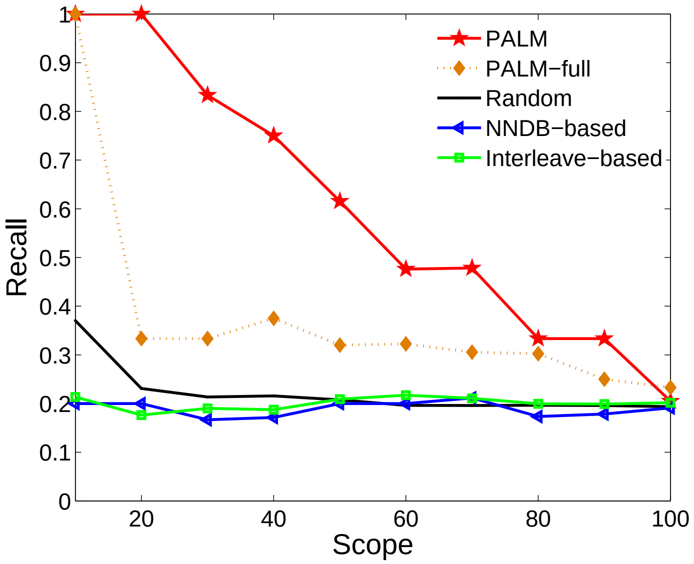


(b)

Figure 5.9: Glass data set: class 1 vs. class 3, $p_2 = 0.195$, 2 features selected by *PALM*.

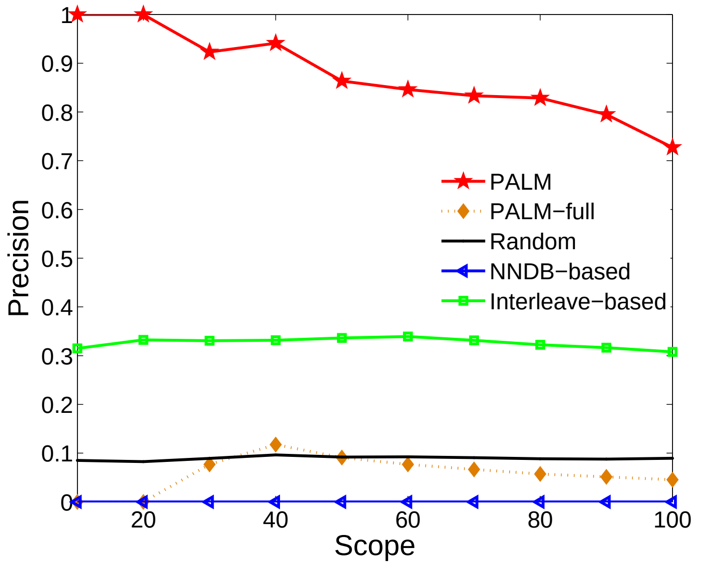


(a)

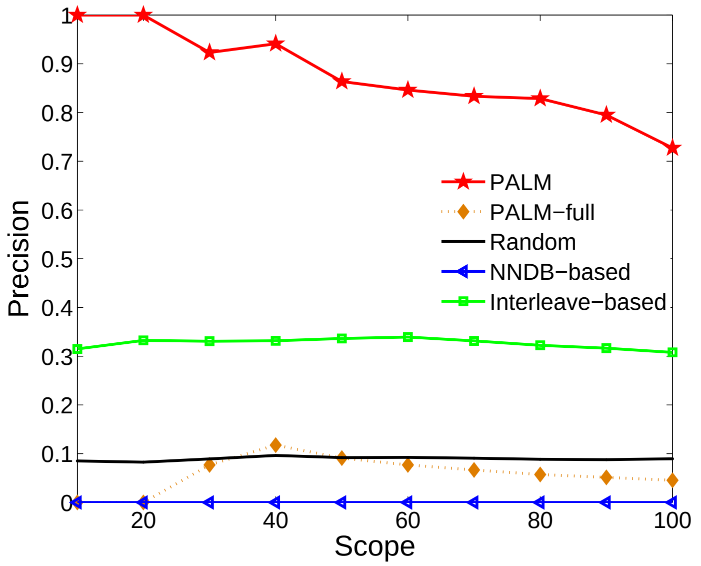


(b)

Figure 5.10: Glass data set: class 2 vs. class 3, $p_2 = 0.183$, 3 features selected by *PALM*.

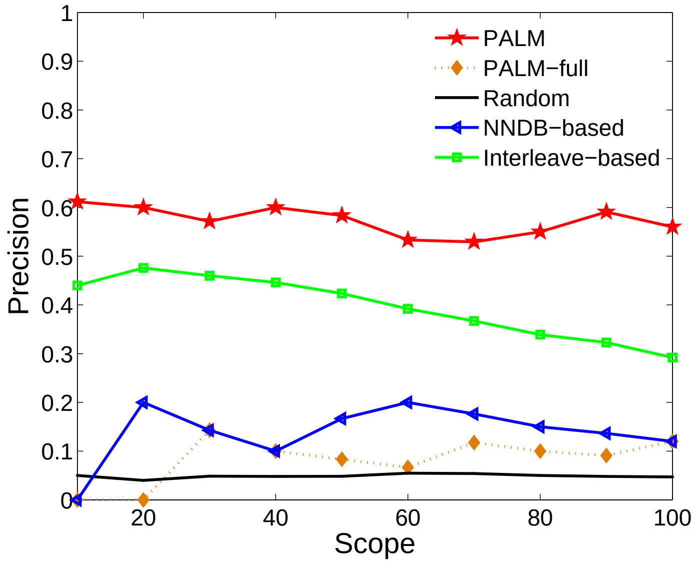


(a)

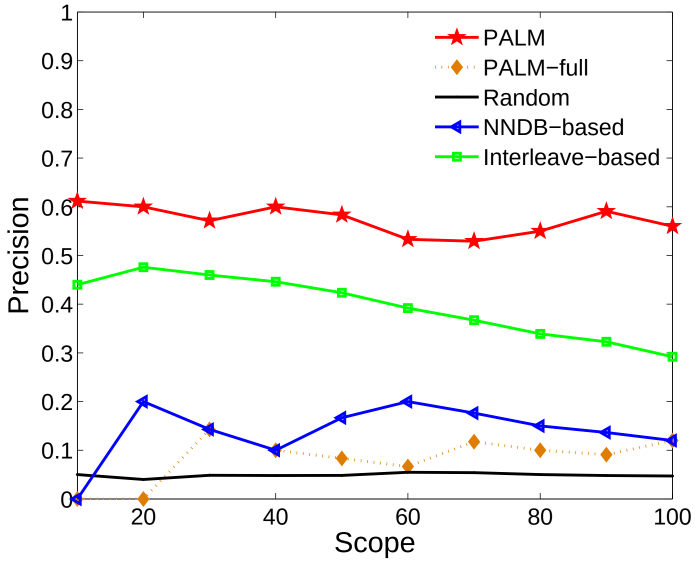


(b)

Figure 5.11: Yeast data set: class 2 vs. class 6, $p_2 = 0.093$, 2 features selected by *PALM*.



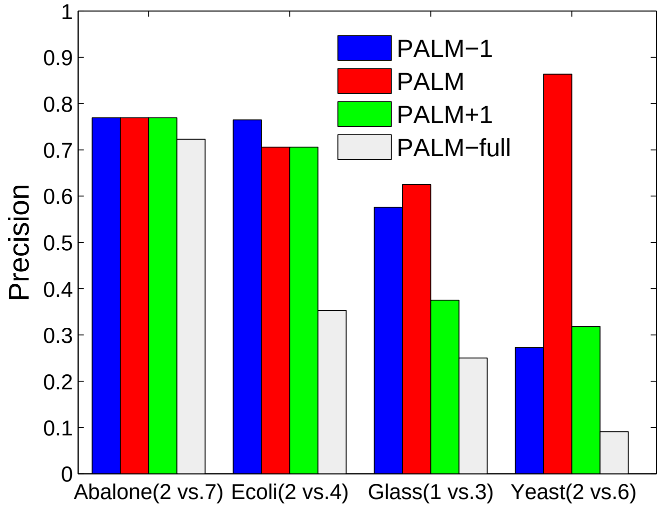
(a)



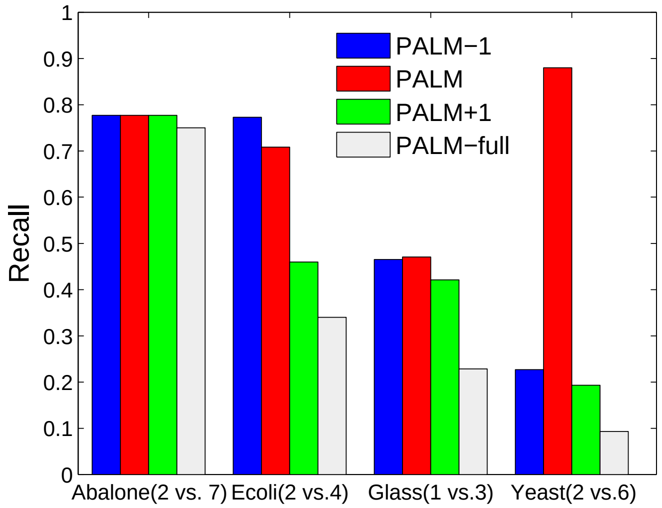
(b)

Figure 5.12: Yeast data set: class 2 vs. class 9, $p_2 = 0.055$, 3 features selected by *PALM*.

Finally, we test the performance of *PALM* when there are small perturbations in the number of relevant features. To this end, we run *PALM* with $d_r$ increased by 1 (*PALM*+1) and decreased by 1 (*PALM*-1) respectively, and compare their performance with *PALM* and *PALM*-full in Fig. 5.13. From this figure, we can see that *PALM* is quite robust against small perturbations in $d_r$ in most cases (Abalone, Ecoli, and Glass), and in all the cases, the performance of *PALM*+1 and *PALM*-1 is better than *PALM*-full (i.e., *PALM* without feature selection).



(a)



(b)

Figure 5.13: Perturbation on the number of relevant features. (Best viewed in color)

## 5.4 Summary of Unsupervised Rare Category Analysis

In this chapter, we have discussed our work on unsupervised rare category analysis. Different from the supervised settings, in the unsupervised settings, we do not have access to the labeling oracle. Under certain assumptions, we are able to address the following two problems: (1) rare category selection, and (2) feature selection. Here, our key observation is that jointly dealing with the two tasks benefits both of them. To this end, we propose an optimization framework, which is well justified theoretically. To solve this optimization problem, we design the Partial Augmented Lagrangian Method (*PALM*), which alternatively finds the relevant features and the minority class examples. The effectiveness of *PALM* is demonstrated by experimental results.

### 5.4.1 Limitations

The limitations of the current work on unsupervised rare category analysis can be summarized as follows.

1. Violations of the assumptions. It is not clearly how the performance of *PALM* will deteriorate if one of more of the assumptions are not satisfied. For example, the support region of the minority class is elongated.
2. Application to data sets with multiple minority classes. The current version of *PALM* can only be applied to the binary settings, i.e., there is one majority class and one minority class in the data set. *PALM* can be generalized to multiple majority and minority classes in the data set by targeting the minority classes one at a time. However, no experiments have been performed in this respect.

# Chapter 6

# Conclusion and Future Directions

Rare categories are of key importance in many real applications: although the occurrence of such examples is rare, their impact is significant. Applications of rare category analysis include: financial fraud detection, Medicare fraud detection, network intrusion detection, astronomy, spam image detection and health care. Based on our empirical studies, we make the following two assumptions: (1) smoothness assumption for the majority classes, and (2) compactness assumption for the minority classes. Notice that we do not assume the majority and minority classes are separable / near-separable from each other in the feature space, which is assumed by most existing work on rare category analysis. In other words, we target the more challenging cases where the support regions of the majority and minority classes overlap with each other in the feature space (although some algorithms work well in both the separable and non-separable cases), since many real applications exhibit the overlapping phenomenon.

In this thesis, we have focused on rare category analysis in both the supervised and unsupervised settings. In particular, the following three tasks are addressed.

1. **Rare category detection**: discovering at least one example from each minority class with the least label requests from the labeling oracle. For data with feature representations, we propose the *NNDB*, *ALICE*, and *MALICE* algorithms that need full prior information of the data set as input, including the number of classes and the proportions of different classes; we also propose the *SEDER* algorithm, which is prior-free. For graph data (relational data), we propose the *GRADE* algorithm that needs full prior information as input; we also generalize this algorithm to produce the *GRADE-LI* algorithm that only needs an upper bound on the proportions of all the minority classes. For each of these algorithms, we provide theoretical justifications as well as empirical evaluations, showing their effectiveness on our data sets.
2. **Rare category characterization**: given labeled examples from all the classes, finding a compact representation for the minority classes in order to identify all the rare examples with high precision and recall. To this end, we propose to enclose the rare examples with a minimum-radius hyper-ball based on the clustering property of the minority classes. This idea is further formulated as an optimization problem, and we design the *RACH* algorithm to find its solution. There are two key components in *RACH*: (1) converting the original problem into a convex optimization problem, and (2) solving it in its dual form by a projected subgradient method. *RACH* can be easily kernelized. Experimental results demonstrate the effectiveness of *RACH*. Furthermore, with the compact representation, we are able to let domain experts (who have no knowledge in machine learning and data mining) better understand the learning results.
3. **Unsupervised rare category analysis**: selecting the examples that are likely to come from the minority classes and selecting the features relevant to the minority classes in an unsupervised fashion. We

propose to co-select the rare examples and the relevant features, which benefits both tasks. To this end, we design an optimization framework, which is well justified theoretically. To solve this optimization problem, we propose the Partial Augmented Lagrangian Method (*PALM*), which alternatively finds the relevant features and the minority class examples. The effectiveness of *PALM* is demonstrated by experimental results.

Rare category analysis can be extended in multiple dimensions. For example,

1. **Understanding the dynamics of rare categories.** To be specific, how does a rare category emerge and evolve over time? Take emerging research topics as an example. A new research topic may start from a single paper exploring a new research direction, which can be followed by more researchers working in the same area, forming a minority class. Finally, it may become a major research topic. Understanding the dynamics of rare categories can help us gain deeper insights of these categories and provide tools for predicting the occurrence of these categories. To address this problem, as a first step, we may want to monitor the number of papers on a new research topic over time to see if there is any change in the distribution once the research topic has become a minority class. Then we may use the above observation to predict if a new research topic will eventually become a major research topic or gradually disappear. These techniques can also be used in disease evolution to monitor possible variations of a certain disease over time.

2. **Understanding complex fraud.** In many real world problems, the fraud patterns may be more complex than a single fraudulent transaction or a bogus claim. For example, in eBay, the fraudsters and accomplices form a bipartite core. The fraud identities are eventually used to carry out the actual non-delivery fraud, while the accomplices exist only to help the fraudsters by boosting their feedback rating [Chau *et al.*, 2006]. Such bipartite cores are a tell-tale sign of a popular fraud scheme, which has resulted in the total loss in the order of millions. To discover these fraud patterns, we need to focus on the subgraph level. As a first step, we may apply the *GRADE* or *GRADE-LI* algorithms proposed in Section 3.3 on the whole transaction graph to discover the complex fraud. Then we can incorporate the bipartite nature of the subgraphs into the algorithms. For example, we may try to modify the global similarity so that vertices in the same bipartite graph tend to have larger global similarity.

3. **Transfer learning for rare category analysis.** In this aspect, we focus on both inter-domain and intra-domain transfer learning. In the first case, the goal is to leverage the information of rare categories in a source domain to help us understand the rare categories in the target domain. Here, the major challenge is the lack of label information related to the rare categories in the target domain. For example, compared with financial fraud, Medicare fraud is less studied, but the different areas may share common fraud patterns. To address this problem, first of all, in order to leverage the label information from the source domain, we need to build some connection between the rare categories in the source domain and the target domain. Do they have similar distributions in the feature space? Are certain parameters shared by these rare categories across different domains? Do they evolve over time in a similar way? Then based on the shared properties of these rare categories, we can effectively transfer the label information from the source domain to the target domain.

   In the second case, we work in a single domain, and the goal is to make use of known rare categories to help us detect new rare categories. Here, the major challenge is the lack of label information related to the new rare categories. For example, in disease diagnosis, studying patients with known flu types may help us detect new flu variants. Similar as before, to address this problem, we first need to find the shared properties of the rare categories, both known and unknown. Then we may have a better clue of the support regions for the new rare categories. By sampling in these regions, we may be able to reduce the number of label requests from the labeling oracle.

4. **Rare category exploration system.** We would like to build a complete system for rare category exploration. The goal here is to provide the domain experts, who have little or no engineering knowledge, with an intuitive user interface. In this way, it may be easier for the domain experts to interact with the data, convert domain knowledge into certain parameters, understand the learning results, and provide feedback. On the input side, the system should be able to deal with different types of data, make use of different amount of prior information, and if some label information is available, it should also be able to use this information to improve the performance. Some major modules of the system include detection, characterization, prediction, feature selection, transfer learning, modeling, etc. On the output side, we hope to discover new minority class as well as new examples from known minority classes; identify key features to the minority classes as well as find their compact representations; built statistical models for the rare categories, etc. Furthermore, we plan to incorporate relevant feedback into the system to better serve the users' need.

# Bibliography

[Aggarwal & Yu, 2001] Aggarwal, C. C. & Yu, P. S. (2001). Outlier detection for high dimensional data. in *SIGMOD Conference* pp. 37–46.

[Angluin, 1987] Angluin, D. (1987). Queries and concept learning. Machine Learning *2*(4), 319–342.

[Angluin, 2001] Angluin, D. (2001). Queries revisited. in *ALT* pp. 12–31.

[Asuncion & Newman, 2007] Asuncion, A. & Newman, D. (2007). UCI machine learning repository.

[Balcan *et al.*, 2006] Balcan, M.-F., Beygelzimer, A., & Langford, J. (2006). Agnostic active learning. in *ICML* pp. 65–72.

[Barbará *et al.*, 2001] Barbará, D., Wu, N., & Jajodia, S. (2001). Detecting novel network intrusions using bayes estimators. in *Proceedings of the First SIAM Conference on Data Mining*.

[Bay *et al.*, 2006] Bay, S., Kumaraswamy, K., Anderle, M. G., Kumar, R., & Steier, D. M. (2006). Large scale detection of irregularities in accounting data. in *ICDM* pp. 75–86.

[Bekkerman *et al.*, 2005] Bekkerman, R., El-Yaniv, R., & McCallum, A. (2005). Multi-way distributional clustering via pairwise interactions. in *ICML* pp. 41–48 New York, NY, USA. ACM.

[Boyd & Vandenberghe, 2004] Boyd, S. & Vandenberghe, L. (2004). (Cambridge University Press ).

[Chandola *et al.*, 2009] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Comput. Surv. *41*(3).

[Chau *et al.*, 2006] Chau, D. H., Pandit, S., & Faloutsos, C. (2006). Detecting fraudulent personalities in networks of online auctioneers. in *PKDD* pp. 103–114.

[Chawla, 2009] Chawla, N. (2009). Mining when classes are imbalanced, rare events matter more, and errors have costs attached. in *SDM*.

[Chawla *et al.*, 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR) *16*, 321–357.

[Chawla *et al.*, 2003] Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting. in *PKDD* pp. 107–119.

[Cohn *et al.*, 1992] Cohn, D., Atlas, L., & Ladner, R. (1992). Improving generalization with active learning.

[Cohn *et al.*, 1996] Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH *4*, 129–145.

[Coppersmith & Winograd, 1987] Coppersmith, D. & Winograd, S. (1987). Matrix multiplication via arithmetic progressions. in *STOC* pp. 1–6 New York, NY, USA. ACM.

[Dagan & Engelson, 1995] Dagan, I. & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. in *In Proceedings of the Twelfth International Conference on Machine Learning* pp. 150–157. Morgan Kaufmann.

[Dasgupta, 2005] Dasgupta, S. (2005). Coarse sample complexity bounds for active learning. in *NIPS*.

[Dasgupta & Hsu, 2008] Dasgupta, S. & Hsu, D. (2008). Hierarchical sampling for active learning. in *ICML* pp. 208–215.

[Dash *et al.*, 2002] Dash, M., Choi, K., Scheuermann, P., & Liu, H. (2002). Feature selection for clustering - a filter solution. in *ICDM* pp. 115–122.

[Dhillon *et al.*, 2003] Dhillon, I., Mallela, S., & Modha, D. (2003). Information-theoretic co-clustering. in *KDD* pp. 89–98.

[Duchi *et al.*, 2008] Duchi, J., Shalev-Shwartz, S., Singer, Y., & Chandra, T. (2008). Efficient projections onto the $l_1$-ball for learning in high dimensions. in *ICML* pp. 272–279.

[Duda *et al.*, 2000] Duda, R. O., Hart, P. E., & Stork, D. G. (2000). (Wiley-Interscience Publication ).

[Dunn, 1973] Dunn, J. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Cyber. and Sys. *3*, 32–57.

[Dutta *et al.*, 2007] Dutta, H., Giannella, C., Borne, K. D., & Kargupta, H. (2007). Distributed top-k outlier detection from astronomy catalogs using the demac system. in *SDM*.

[Dy & Brodley, 2000] Dy, J. G. & Brodley, C. E. (2000). Feature subset selection and order identification for unsupervised learning. in *ICML* pp. 247–254.

[Efron & Tibshirani, 1996] Efron, B. & Tibshirani, R. (1996). Using specially designed exponential families for density estimation. in *Proc. of the Annals of Statistics* pp. 2431–2461.

[EURODIS, 2005] EURODIS (2005). Rare diseases: Understanding this public health priority.

[Fine & Mansour, 2006] Fine, S. & Mansour, Y. (2006). Active sampling for multiple output identification. in *COLT* pp. 620–634.

[Flake *et al.*, 2000] Flake, G., Lawrence, S., & Giles, C. (2000). Efficient identification of web communities. in *KDD* pp. 150–160.

[Gao *et al.*, 2007] Gao, B., Liu, T., Zheng, X., Cheng, Q., & Ma, W. (2007). Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. in *KDD* pp. 41–50.

[Gibson *et al.*, 2005] Gibson, D., Kumar, R., & Tomkins, A. (2005). Discovering large dense subgraphs in massive graphs. in *VLDB* pp. 721–732.

[Glenn & Fung, 2001] Glenn, C. A. & Fung, G. (2001). A comprehensive overview of basic.

[Greco *et al.*, 2007] Greco, G., Guzzo, A., & Pontieri, L. (2007). An information-theoretic framework for high-order co-clustering of heterogeneous objects. in *SEBD* pp. 397–404.

[Guo & Viktor, 2004] Guo, H. & Viktor, H. L. (2004). Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. SIGKDD Explorations *6*(1), 30–39.

[HART, 1968] HART, P. (1968). The condensed nearest neighbor rule. IEEE Trans. Inform. Th. *14*, 515–516.

[He & Carbonell, 2008] He, J. & Carbonell, J. (2008). Rare class discovery based on active learning. in *ISAIM*.

[He & Carbonell, 2007] He, J. & Carbonell, J. G. (2007). Nearest-neighbor-based active learning for rare category detection. in *NIPS*.

[He *et al.*, 2005a] He, X., Cai, D., & Niyogi, P. (2005a). Laplacian score for feature selection. in *NIPS*.

[He *et al.*, 2005b] He, Z., Xu, X., & Deng, S. (2005b). An optimization model for outlier detection in categorical data. CoRR *abs/cs/0503081*.

[Hofmann & Buhmann, 1997] Hofmann, T. & Buhmann, J. M. (1997). Pairwise data clustering by deterministic annealing. IEEE Trans. Pattern Anal. Mach. Intell. *19*, 1–14.

[Huang *et al.*, 2004] Huang, K., Yang, H., King, I., & Lyu, M. R. (2004). Learning classifiers from imbalanced data based on biased minimax probability machine. in *CVPR (2)* pp. 558–563.

[Joachims, 1999] Joachims, T. (1999). Transductive inference for text classification using support vector machines. in *ICML* pp. 200–209.

[Joachims, 2005] Joachims, T. (2005). A support vector method for multivariate performance measures. in *ICML* pp. 377–384.

[Johnson, 1967] Johnson, S. (1967). Hierarchical clustering schemes. Psychometrika *32*(3), 241–254.

[Kim *et al.*, 2000] Kim, Y., Street, W. N., & Menczer, F. (2000). Feature selection in unsupervised learning via evolutionary search. in *KDD* pp. 365–369.

[Kim & Choi, 2007] Kim, Y.-D. & Choi, S. (2007). A method of initialization for nonnegative matrix factorization. in *ICASSP* pp. II–537–II–540.

[Kubat & Matwin, 1997] Kubat, M. & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. in *ICML* pp. 179–186.

[Kumar *et al.*, 2003] Kumar, R., Novak, J., Raghavan, P., & Tomkins, A. (2003). On the bursty evolution of blogspace. in *WWW* pp. 568–576.

[Law *et al.*, 2002] Law, M. H. C., Jain, A. K., & Figueiredo, M. A. T. (2002). Feature selection in mixture-based clustering. in *NIPS* pp. 625–632.

[Lehoucq & Sorensen, 1996] Lehoucq, R. & Sorensen, D. (1996). Deflation techniques for an implicitly restarted arnoldi iteration. SIAM J. Matrix Anal. Appl. *17*(4), 789–821.

[Long *et al.*, 2006] Long, B., Zhang, Z., Wu, X., & Yu, P. (2006). Spectral clustering for multi-type relational data. in *ICML* pp. 585–592.

[Madsen *et al.*, 2004] Madsen, K., Nielsen, H. B., & Tingleff, O. (2004). Optimization with constraints, 2nd ed.

[Mccallum, 1998] Mccallum, A. K. (1998). Employing em in pool-based active learning for text classification. in *In Proceedings of the 15th International Conference on Machine Learning* pp. 350–358. Morgan Kaufmann.

[Mitra *et al.*, 2002] Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Anal. Mach. Intell. *24*(3), 301–312.

[Moore, 1991] Moore, A. (1991). A tutorial on kd-trees Technical report University of Cambridge Computer Laboratory.

[Nocedal & Wright, 1999] Nocedal, J. & Wright, S. J. (1999). (Springer ).

[Papadimitriou *et al.*, 2003] Papadimitriou, S., Kitagawa, H., Gibbons, P. B., & Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. in *ICDE* pp. 315–327.

[Pelleg & Moore, 2004] Pelleg, D. & Moore, A. W. (2004). Active learning for anomaly and rare-category detection. in *NIPS*.

[Ramaswamy *et al.*, 2000] Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. in *SIGMOD* (Chen, W., Naughton, J. F., & Bernstein, P. A., eds.) pp. 427–438. ACM.

[Reynolds & Rose, 1995] Reynolds, D. & Rose, R. (1995). Robust text-independent speaker identification using gaussian mixture speaker models. IEEE Trans. Speech and Audio Processing *3*, 72–83.

[Schölkopf *et al.*, 2001] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural Computation *13*(7), 1443–1471.

[Scott, 1992] Scott, D. (1992). (Wiley-Interscience ).

[Settles, 2010] Settles, B. (2010). Active learning literature survey Computer Sciences Technical Report 1648 University of Wisconsin–Madison.

[Sheikholeslami *et al.*, 1998] Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. in *VLDB* pp. 428–439.

[Singh & Gordon, 2008a] Singh, A. P. & Gordon, G. J. (2008a). Relational learning via collective matrix factorization. in *KDD* pp. 650–658.

[Singh & Gordon, 2008b] Singh, A. P. & Gordon, G. J. (2008b). A unified view of matrix factorization models. in *ECML/PKDD (2)* pp. 358–373.

[Sun *et al.*, 2007] Sun, J., Faloutsos, C., Papadimitriou, S., & Yu, P. (2007). Graphscope: parameter-free mining of large time-evolving graphs. in *KDD* pp. 687–696.

[Sun *et al.*, 2006] Sun, Y., Kamel, M. S., & Wang, Y. (2006). Boosting for learning multiple classes with imbalanced class distribution. in *ICDM* pp. 592–602.

[Tang *et al.*, 2009] Tang, Y., Zhang, Y.-Q., Chawla, N. V., & Krasser, S. (2009). Svms modeling for highly imbalanced classification. Trans. Sys. Man Cyber. Part B *39*(1), 281–288.

[Tomek, 1976] Tomek, I. (1976). Two modifications of cnn. IEEE Trans. Systems, Man and Cybernetics *6*, 769–772.

[Tong *et al.*, 2001] Tong, S., Koller, D., & Kaelbling, P. (2001). Support vector machine active learning with applications to text classification. in *Journal of Machine Learning Research* pp. 999–1006.

[Vatturi & Wong, 2009] Vatturi, P. & Wong, W.-K. (2009). Category detection using hierarchical mean shift. in *KDD* pp. 847–856.

[Wang *et al.*, 2007] Wang, Z., Josephson, W. K., Lv, Q., Charikar, M., & Li, K. (2007). Filtering image spam with near-duplicate detection. in *CEAS*.

[Wasserman, 2005] Wasserman, L. (2005). (Springer ).

[Wu & Chang, 2003] Wu, G. & Chang, E. Y. (2003). Adaptive feature-space conformal transformation for imbalanced-data learning. in *ICML* pp. 816–823.

[Wu *et al.*, 2007] Wu, J., Xiong, H., Wu, P., & Chen, J. (2007). Local decomposition for rare class analysis. in *KDD* pp. 814–823.

[Yu *et al.*, 2002] Yu, D., Sheikholeslami, G., & Zhang, A. (2002). Findout: Finding outliers in very large datasets. Knowl. Inf. Syst. *4*(4), 387–412.

[Yuster & Zwick, 2005] Yuster, R. & Zwick, U. (2005). Fast sparse matrix multiplication. ACM Trans. Algorithms *1*(1), 2–13.

[Zhou *et al.*, 2003a] Zhou, D., Bousquet, O., Lal, T., Weston, J., & Scholkopf, B. (2003a). Learning with local and global consistency. in *NIPS*.

[Zhou *et al.*, 2003b] Zhou, D., Weston, J., Gretton, A., Bousquet, O., & Schölkopf, B. (2003b). Ranking on data manifolds. in *NIPS*.

ML
MACHINE LEARNING
DEPARTMENT

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon