1-1-2013

Learning With An Insufficient Supply Of Data Via Knowledge Transfer And Sharing

Samir Al-Stouhi
Wayne State University,

LEARNING WITH AN INSUFFICIENT SUPPLY OF DATA VIA KNOWLEDGE TRANSFER AND SHARING

by

SAMIR AL-STOUHI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2013

MAJOR: COMPUTER ENGINEERING

Approved by:

_____

Advisor Date

_____

Co-advisor Date

_____

_____

_____

DEDICATION

To my Parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

This dissertation addresses a learning problem where the training set does not contain enough information for learning because it has an inadequate supply of training instances with a complex feature space and skewed label distributions. Learning on such a dataset will be referred to as "Learning with Absolute Rarity". In the absence of the evidence required for making a conclusion (learning), our algorithms will apply transfer learning methods to learn form different (but related) datasets or will share knowledge between a set of small datasets in a multi-task learning paradigm. The complexity of the data and the rarity of training examples prohibit learning (hypothesis construction) by standard algorithms and thus our solutions present a set of "last resort" methods to be applied to an area of research that has been identified as very important but has received little attention.

Many problems related to medical diagnosis, fault monitoring and fraud detection have small datasets. For example, there are 6500 rare diseases where each rare disease occurs in fewer than 200,000 individuals in the USA. Rare diseases are a substantial public health burden as $6 - 8\%$ of people have a rare disease at some point of their life and most of these diseases do not have an International Classification of Diseases (ICD) code or even belong to a registry [41]. Improvements, even minor, from methods optimized specifically for "Absolute Rarity" can have significant financial and social impact within domains, such as healthcare, where only human expertise are currently applicable. Other problems such as cancer diagnosis (benign or malignant) for minority demographics or classification of seismic waves (earthquake or nuclear detonation) are datasets that are

small and are also imbalanced. Imbalance is especially true in healthcare where a fraction of the population are critical (unhealthy) patients [92] and a vast majority are healthy. For example, in one healthcare study, breast cancer diagnosis was active in 1.66% of the population while peripheral atherosclerosis was active in 3.16% and aneurysm in 0.74% [59]. Even more rare diseases are active in a very low ratio of the population and include Testis Cancer (0.046%) and male genital disease (0.01%). The imbalanced problem is also prevalent in other rare datasets including the detection of oil spills in satellite radar images [16] and in-flight gearbox fault monitoring [54].

"Absolute Rarity" has been identified by several prominent researchers as an interesting research problem. In 2009, Haibo He and Edwardo Garcia [50] described absolute rarity as "*a relatively new research topic that requires much needed attention in the community*". Gary M Weiss highlights the problem by stating that: "*obtaining additional training data is the most direct way of addressing the problems associated with mining rare cases*" and he states that "*the usage of additional data when the absolute number of samples is rare is an approach that warrants additional research*" [116]. In her 2012 book "Analysis of Rare Categories" [51], Jingrui He singles out transfer learning as a possible future direction. She states that "*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*" or "*to make use of known rare categories to help us detect new rare categories*". Figure 1.1 gives an overview as to how "Absolute Rarity" fits with other machine learning domains. "Absolute Rarity" is the study of learning where training examples are scarce and the class distribution is possibly skewed. When data is not readily available and class labels are equally distributed, transfer learning or multi-task learning methods can compensate for the lack of data with the addition of auxiliary data or the concatenation of similar tasks. Alternatively, when the classes are not equally distributed with an abundance of training samples, imbalanced methods are applied and are less necessi-

tated as the number of training examples increases [117]. The figure depicts how standard machine learning techniques are applicable when data is abundant and class labels are equally distributed. Finally, a new set of "Big Data" methods extend the machine learning domain where the efficient processing of data is of more relevance than the choice of algorithms.

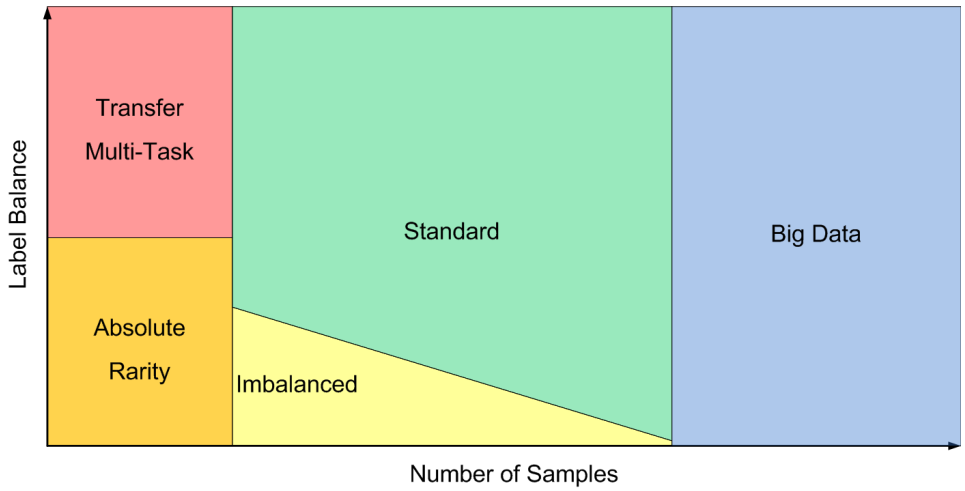


Figure 1.1: Summary of the Learning Domains.

1.2 Thesis Contribution

A dataset with the high dimensionality, small sample size and class-imbalance creates an extremum situation for learning that we aim to address in our thesis. Learning is the science of generating conclusions (or hypotheses) from observed or "training" data. A training dataset serves as the evidence that supports the conclusion and thus a judgment cannot be applied with insufficient evidence. Similarly a hypothesis cannot be generated with an inadequate supply of data and attempting to learn with "Absolute Rarity" is a problem that can only be tackled with the addition of a selected set of knowledge from a secondary source of data. Extending knowledge is not outside the realm of statistical machine learning as it is the science of applying a conclusion to unseen data from knowledge assembled with some previously collected data. This can be described as a transfer of knowledge from what has been observed to what has yet to be observed. This idea of

knowledge extension is actually the foundation of machine learning and we specifically optimize algorithms for this using a bias-variance trade-off [44]. Traditional learning fails when the evidence does not depict the task at hand (target task). For example, determining if an alien in a movie is good or bad requires a dataset of similar aliens, a task that is complicated and not required for human learning as we can instantaneously transfer knowledge from similar experiences. Watching a movie, there is no need to specify which alien creature is evil, it is just instinctive to us as we take cues from what we already know to something that we have never encountered. That knowledge comes from a mix of unrelated experiences and each individual fuses their own set of experiences to transfer. It takes one good act from that seemingly evil alien for us to realize that what we knew about good and evil was negative transfer that does not apply to this specific task and we instantly reject (or reverse) that transfer. Selecting only a subset of evidence (the concept to what an alien should look like) and rejecting or modifying another subset of evidence (how a good or evil alien should look like) is a type of knowledge transfer that is fundamentally different from standard learning theory since a volume of evidence can be instantly rejected, transformed or reversed if a very small amount of data from the target task deviates from what was already learned.

Motivated with real-world problems, we present the first body of work to address a problem with significant financial and social impact and we aim to extend the machine learning domain to an area that has been been identified but never specifically addressed. Our main contributions:

1. **Chapter 2:** We describe "Rare Datasets" from different views. We highlight related fields and describe the theoretical limitations that prohibit learning. We highlight the difficulties encountered and the need for a new set of algorithms and evaluation methods to solve an important problem.

2. **Chapter 3:** We highlight the problems with current boosting methods for instance

transfer learning algorithms and propose a novel supervised transfer learning algorithm to improve upon the most popular transfer learning method.

3. **Chapter 4:** We extend the transfer learning algorithm proposed in Chapter 2 to address "Absolute Rarity". This algorithm addresses a "Rare Dataset" learning problem with a transfer learning approach. This is the first method to specifically address "Absolute Rarity".

4. **Chapter 5:** We compare the performance of our methods with several real-world demographics datasets. We also extend our work with an algorithm to address "Absolute Rarity" using an oversampling based approach.

5. **Chapter 6:** We propose an unsupervised multi-task learning algorithm. Instead of extending knowledge from an auxiliary domain with an ample supply of samples, we concatenate the knowledge from several small datasets with an unsupervised multi-clustering technique.

Figure 1.2 presents an overview of the thesis's organization.

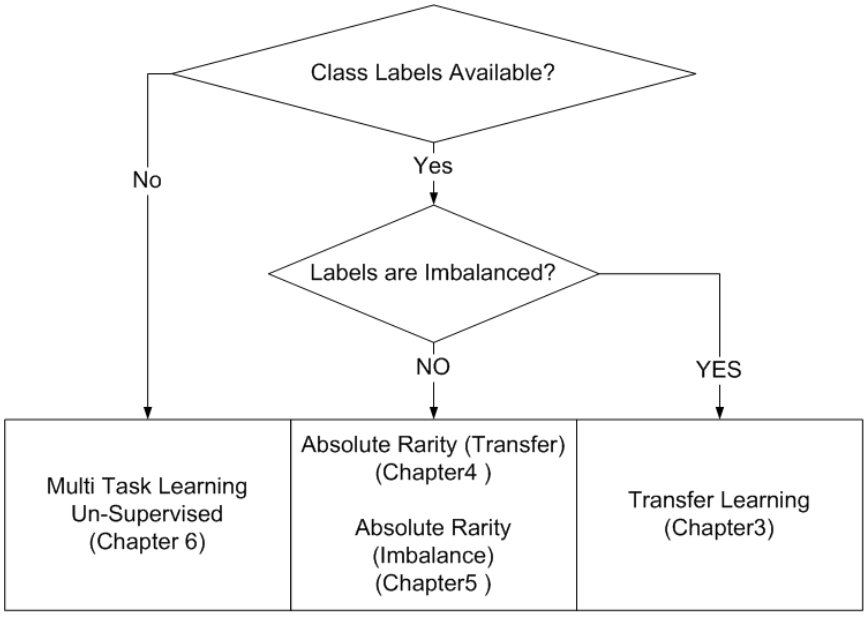


Figure 1.2: Thesis Organization

CHAPTER 2

ANALYSIS OF "ABSOLUTE RARITY"

2.1 Introduction

"Absolute Rarity" refers to the problem of learning when the number of examples associated with a a class is too small to construct a hypothesis capable of generalizing to unseen data. It is an understudied problem because the lack of representative training samples, especially within the minority class, impede learning. In this chapter, we:

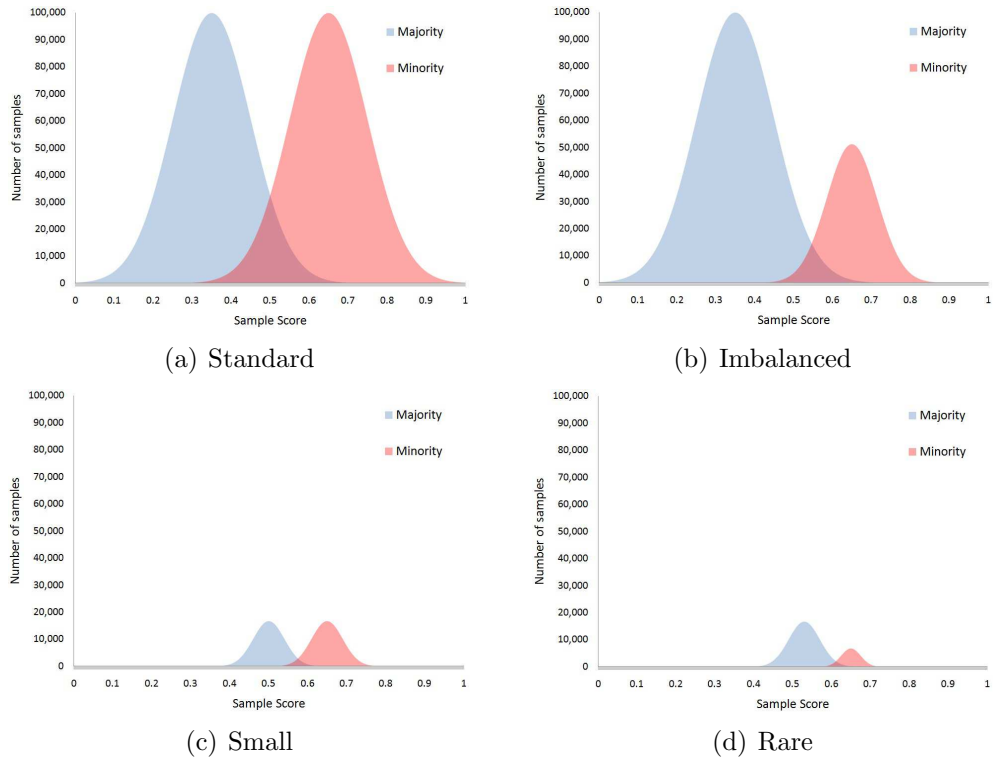
1. Describe "Absolute Rarity" with a Label-Dependent Distribution view.
2. Highlight the difference and relation to other machine learning domains.
3. Give an overview of learning methods and evaluation metrics that are necessary for learning with "Absolute Rarity".
4. Analyze the issues that impede learning.
5. Propose an evaluation plot.

2.2 Label-Dependent view

To describe datasets in terms of both size and imbalance, we use the "Label-Dependent" view in Figure 2.1. The sub-figures present a binary classification problem with normally distributed samples within each class[1] (thus we describe it as label-dependent since the distributions are normal within each label). Figure 2.1 illustrates the different datasets with an overview of the related machine learning fields[2] that can improve learning.

1. **Standard dataset:** Figure 2.1-a depicts a standard dataset with a relatively equal

[1]The terms class and label are used interchangeably in our discussion.
[2]Only concepts that are relevant for "Absolute Rarity" are discussed.



(a) Standard (b) Imbalanced

(c) Small (d) Rare

Figure 2.1: Label-Dependent view of different type of datasets.

number of samples within each class (balanced class distribution) and an adequate number of samples for generalization. To learn from balanced datasets, researchers assign equal importance to all classes and thus maximizing the overall arithmetic accuracy is the chosen optimization objective. A variety of standard machine learning and data mining approaches can be applied for standard datasets as such methods serve as the foundation for the algorithms that are modified for any peculiar feature set or distribution.

2. **Imbalanced dataset:** The dataset in Figure 2.1-b is a relatively-imbalanced dataset. It is relatively-imbalanced because there is a between-class imbalance where one class encompass the majority of the training set. The balance is relative since both minority and majority training subsets contain adequate training examples. For example, email spam classification is a relatively imbalanced problem since

97% (majority) of emails sent over the net are considered unwanted emails [113] and with around 200 billion messages of spam sent per day [48], the number of non-spam emails (minority) is also a large dataset. Researchers refer to a relatively imbalanced dataset as an "imbalanced dataset" with the postulation that the imbalance is relative [50]. Because the majority class overwhelms the minority class, imbalanced learning models are biased to improve learning on the minority class (without any consideration to the availability of training examples).

3. **Small dataset:** The dataset in Figure 2.1-c is a balanced dataset with a training sample size that is inadequate for generalization. One method to determine the number of samples required for training is to rely on the "Probably Approximately Correct (PAC)" learning theory [58]. PAC is applied to determine if the ratio of the dimensions of the data to the number of training samples is too high where the hypothesis space would thus be exceedingly large. If that ratio is too high, learning is difficult and prone to model over-fitting. PAC gives a theoretic relationship between the number of samples needed in terms of the size of hypothesis space and the number of dimensions. The simplest example is a binary dataset with binary classes and d dimensions with hypothesis space of size 2^{2^d}, requiring $O\left(2^n\right)$ samples [74].

4. **"Absolute Rarity":** The dataset in Figure 2.1-d is small and imbalanced and thus its imbalance is termed as "Absolute Rarity". Weiss [115] presents a good overview of the problems encountered when analyzing and evaluating such datasets. Different solutions are outlined for handling "Absolute Rarity" with a discussion of solutions for segmentation, bias and noise associated with these datasets. In [51], an end-to-end investigation of rare categories in imbalanced datasets in both the supervised and unsupervised settings is presented.

2.3 Methods for Learning with "Absolute Rarity"

Figure 2.2 depicts a summary of the machine learning domains that are leveraged for different label distributions and sample sizes. The figure will demonstrate how our work will apply the transfer learning paradigm to address the problem of "Absolute Rarity". This research presents the first set of methods that are specifically optimized for "Rare Dataset" learning with a transfer and multi-task learning paradigm.

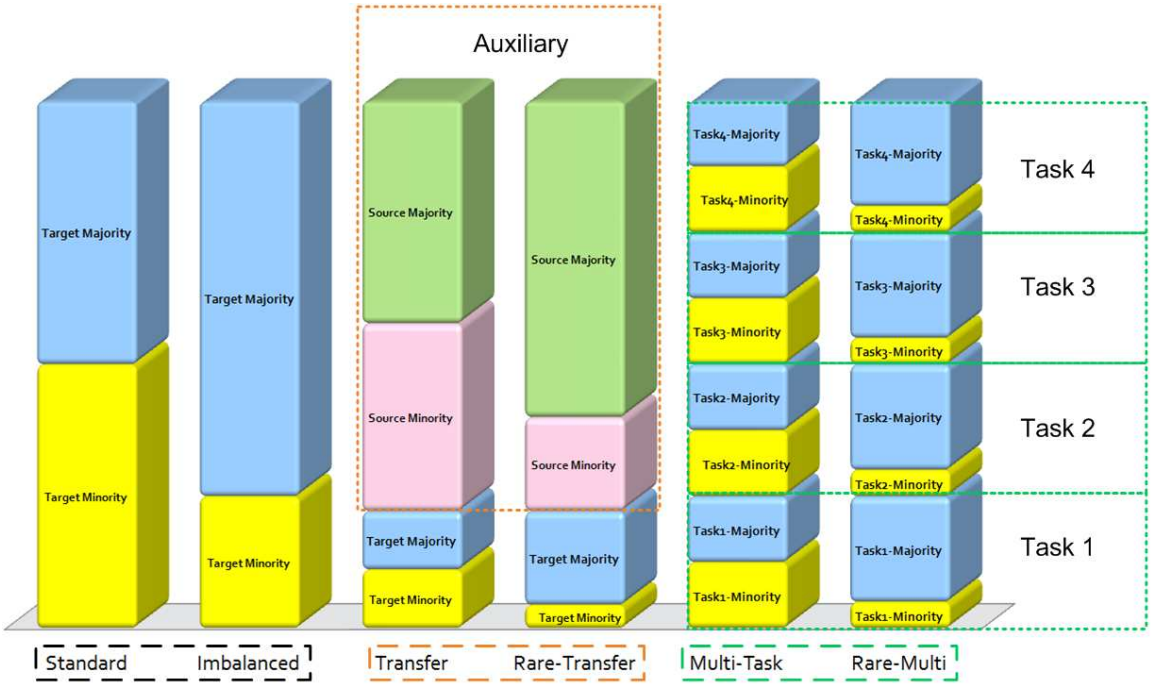


Figure 2.2: Learning methods for different datasets' distribution and size.

1. Standard Learning: Any Standard learning method can be applied and that includes decision trees, k-means, support vector machines and others [120].

2. Imbalanced Learning: Section 2.4.1 presents an overview of the imbalanced learning domain. Chapter 5 presents our method for balancing a dataset using an oversampling approach that leverages an auxiliary domain for its minority samples.

3. Transfer Learning: Methods for transfer learning improve a learning task by leveraging helpful knowledge from an auxiliary task or dataset. Section 2.4.2 presents an

overview of transfer learning. Chapter 3 presents our supervised transfer learning method.

4. Rare-Transfer Learning: This refers to the extension of transfer learning to "Absolute Rarity". Rare-Transfer extends transfer learning to imbalanced learning and Chapter 4 presents the first learning method to specifically optimize for "Absolute Rarity" with a transfer learning approach.

5. Multi-Task Learning: Given a combination of small and balanced datasets. Learning can be improved if multiple related tasks can share knowledge. Multi-task learning aims at improving the generalization performance where several related problems can be simultaneously optimized by utilizing the intrinsic relationships among multiple tasks. Chapter 6 presents our unsupervised multi-task learning method.

6. Rare-Multi-Task Learning: Extending the multi-task learning paradigm, multi-task imbalanced datasets can be learned simultaneously to improve balanced measures. This will be presented as a possible extension to Chapter 6.

2.4 Related Domains

2.4.1 Imbalanced Learning

Traditional learning methods maximize accuracy and thus fail to generalize to imbalanced datasets because the generated classifiers are biased towards the majority class [83, 50, 76, 38]. Imbalanced classification is a well-studied problem and many sampling methods, cost-sensitive learning methods, kernel-based learning techniques, and active learning methods have been proposed [49, 98]. It is typically assumed that the minority class is more important and more difficult to classify and thus the class imbalanced algorithms and evaluations give the minority class more importance [38].

Example: Breast Cancer Detection

In this section, an example is presented to demonstrate how an imbalanced learning algorithm's bias is dependent on the desired outcome. Breast cancer detection is an imbalanced classification problem where a majority of tests are Negative (patient is healthy) while a minority of tests are Positive (patient has breast cancer). In a dataset that is collected from patients' records, an overwhelming majority of data belongs to healthy patients and a standard classifier could achieve high accuracy by mistakenly classifying all patients as healthy. To correct for the imbalance in the data, the classifier's output should be biased to improve the balanced classification metrics including the Balanced Accuracy [47], the Geometric Mean [63] and the Harmonic Mean [89].

First Step (Screening) The first step for breast cancer detection is a screening test. Figure 2.3 illustrates an example of a classifier optimized for breast cancer screening. Breast cancer screening tests include Mammography [99] and Optical Spectroscopy [110] and these tests are cost effective, routine and non-invasive. Once data is collected, a constructed classifier should be biased to minimize false negatives since a false classification of a healthy patient as having breast cancer triggers a diagnostic test. While it is an undesired outcome to classify a healthy patient as cancerous, it is a preferred outcome when compared to a false classification of a cancerous patient as healthy.

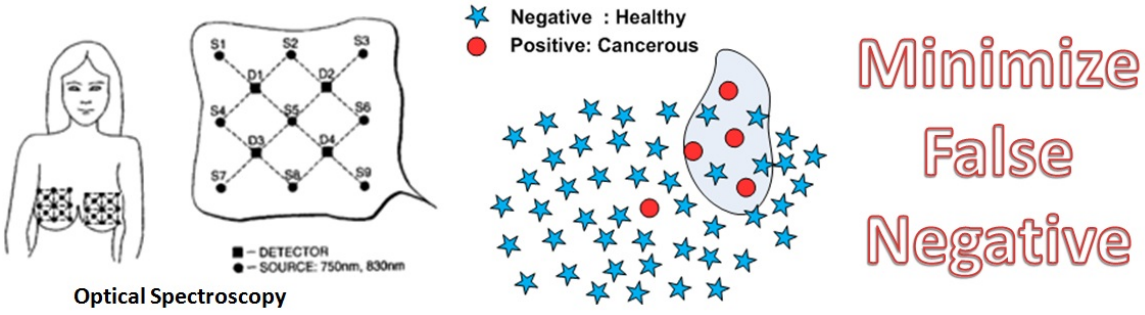


Figure 2.3: Breast Cancer Screening Test

Last Step (Diagnostic) The last step for breast cancer detection is a diagnostic test. Figure 2.4 presents an example of a classifier optimized for a breast cancer diagnostic test. Breast cancer diagnostic tests include Biopsy [108] and Raman Spectroscopy [118] and these test are generally more expensive and invasive than screening tests. Once data is collected, a constructed classifier should be biased to minimize false positives since a false classification of a healthy patient as having breast cancer triggers a false surgical procedure. While it is an undesired outcome to miss the detection of cancer in an unhealthy patient, it is considered a preferred outcome when compared to a false classification where a healthy patient undergoes a surgical procedure to remove a non existing tumor.

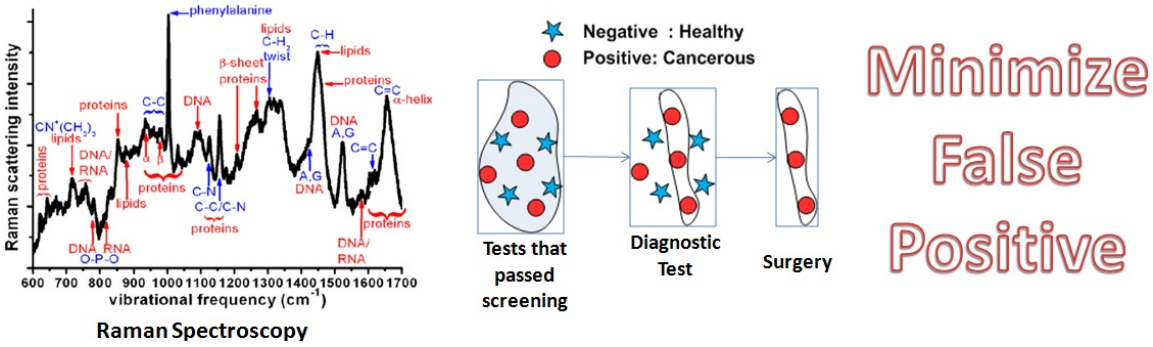


Figure 2.4: Breast Cancer Diagnostic Test

This example demonstrated how the same problem requires two different specialty classifiers that not only optimize for class imbalance but also optimize for opposing class labels.

Evaluation Metrics

A confusion matrix [60] contains the information required for evaluating a learning system. In a confusion matrix, a True Negative (TN) is a negative instance that is correctly classified and a True Positive (TP) is a correctly classified positive sample. On the other hand, a False Positive (FP) is a negative instance that is incorrectly classified

as positive and a False Negative (FN) is a positive instance that is incorrectly classified as negative. Table 2.1 presents the confusion matrix for a two-class classifier. In con-

Table 2.1: Confusion Matrix

		Predicted	
		Positive	Negative
Actual		TP	FN
		FP	TN

vention, the class that occupies a minor fraction of data is called the positive class, and instances that belong to this class are called positive instances (or positive samples). The other class that takes a majority proportion of data is called the negative class and its instances are negative instances. This is the standard convention in imbalanced learning methods as the domain stems from the information retrieval science. The medical domain has a reverse label definition where the majority instance's label is considered negative (healthy) while the minority instance's label is considered positive (un-healthy). Given the definitions of positive and negative classes, we will only refer to a label as a minority or a majority label without referencing if it is positive of negative.

Measures of Imbalance

Imbalance Ratio: The degree of imbalance in the class distribution is the ratio of the sample size of the minority class to that of the majority class. In real-world applications, this ratio varies drastically and can range from 1:10 to 1:1000 or even smaller [18]. A study investigated the correlation between the degree of class imbalance within a training dataset and the classification performance of decision trees classifiers [117]. In this study, 26 UCI [42] datasets were used to determine the range of imbalance ratios that causes a deterioration in a classifier's performance and it was found that there is no specific ratio. In some applications, a drop in classification performance occurs at a ratio as low as 1:35

while other applications can only require a ratio of around 1:10 or even higher [57].

Distance Measures: Other measure that are used in imbalance learning techniques include distance measures such as Hellinger distance [104]. Let P and Q denote two probability measures that are absolutely continuous with respect to a third probability measure λ. The squared Hellinger distance is the quantity given by:

$$H^2(P,Q) = \frac{1}{2} \int \left(\sqrt{\frac{\mathrm{d}P}{\mathrm{d}\lambda}} - \sqrt{\frac{\mathrm{d}Q}{\mathrm{d}\lambda}} \right)^2 \mathrm{d}\lambda \tag{2.1}$$

Hellinger distance was used as the decision tree splitting criterion in [20] where it was demonstrated that this measure produces, under class imbalance, decision trees that are superior to standard decision trees such as C4.5 [86] and CART [15].

2.4.2 Transfer Learning

To overcome the theoretical bounds of learning with an inadequate number of samples, "Transfer Learning" methods [79] can be applied to develop learning models for a small dataset (referred to as target set) by including a similar and possibly larger auxiliary dataset (referred to as the source set). This knowledge transfer is achieved by integrating relevant source knowledge into the training model or by mapping the source data or models to the target. The knowledge assembled can be transferred across domain tasks and domain distributions with the assumption that the auxiliary data is relevant. Pan and Yang [79] present a comprehensive survey of transfer learning methods and discuss the relationship between transfer learning and other related machine learning domains.

History of Transfer Learning Research

Transfer of learning started as the study of the dependency of human conduct, learning, or performance on prior experience. The notion was originally introduced in a 1901 study as "Transfer of Practice"[100]. The study explored how individuals would transfer

learning in one context to another context that shared similar characteristics or more formally how "improvement in one mental function could influence another related one". This study found that transfer of learning depends on the proportion to which the learning task and the transfer task are similar, or where "identical elements are concerned in the influencing and influenced function" in what is now known as "Identical Element Theory" [100, 102].

Transfer learning as a machine learning domain gained attention in the 1990's with the first comprehensive survey in Thrun's "Learning To Learn"[101]. Researchers in transfer learning give the domain different names including: learning to learn, life-long learning, knowledge transfer, inductive transfer, multi-task learning, knowledge consolidation, context-sensitive learning, knowledge-based inductive bias, meta-learning, and incremental/cumulative learning [101, 79]. In NIPS 1995, a two-day workshop on "Learning to Learn", focused on the need for lifelong machine learning methods that retain and reuse learned knowledge. Fast forward to NIPS 2005, a workshop titled "Inductive Transfer: 10 Years Later", examined the progress that has been made in ten years and identified the questions and challenges that remain and the opportunities for new applications of inductive transfer systems. The Defense Advanced Research Projects Agency (DARPA) Information Processing Technology Office (IPTO) also identified transfer learning in 2005 with solicitation number BAA05-29 as: *"The goal of the Transfer Learning Program solicited by this BAA is to develop, implement, demonstrate and evaluate theories, architectures, algorithms, methods, and techniques that enable computers to apply knowledge learned for a particular, original set of tasks to achieve superior performance on new, previously unseen tasks. This goal reflects the observation that key cognitive abilities of humans include the abilities to generalize, abstract, reuse, reorganize and apply knowledge learned in previous life experiences to novel situations."*.

Near and Far Transfer

Near transfer refers to the transfer of knowledge between similar tasks or tasks with common elements [82]. For example, an auto mechanic can transfer his knowledge of vehicle engines to repair a small boat's engine. Far transfer refers to transfer between tasks that seem more remote and on the surface have little in common. For example, a military commander can make a good CEO or a chess player can make a good poker player. Far transfer is much more difficult and generally requires additional knowledge. For instance, the majority of people stay within one area of research as moving to a new area generally takes re-training.

The same concept of Near vs Far transfer is applicable to machine learning. The transfer learning distance is generally quantified with distance based measures. The most popular measure of transfer distance is the Kullback-Leibler (KL) divergence [66]. KL divergence is a non-symmetric measure of the distance between two probability distributions P and Q. This divergence is also referred to as information divergence and relative entropy and is defined as:

$$D_{\mathrm{KL}}(P||Q) = \sum_{x} P(x) \ln \left(\frac{P(x)}{Q(x)} \right) \tag{2.2}$$

If the densities P and Q exist with respect to a Lebesgue measure [8], then the KL divergence of Q from P gives a measure of the information lost when Q (model) is approximating P (true distribution). The KL-divergence is a specific example of a Bregman divergence [13] and is useful for estimating whether two set of samples have been drawn from the same distribution. This is essential for transfer learning as KL divergence can be used to determine the distance between two distributions or this distance metric can be included as an optimization criterion in a transfer learning algorithm.

KL divergence was used for transfer learning in [103] and the minimization of this divergence in [97] was used to directly calculate a weight w for the ratio of two density

functions for covariance shift between a training (tr) and testing (te) distributions as:

$$\begin{aligned} D_{\mathrm{KL}}(p_{te}||wp_{tr}) &= \int p_{te}(x) \ln\left(\frac{p_{te}(x)}{w(x)p_{tr}(x)}\right) dx \\ &= \int p_{te}(x) \ln\left(\frac{p_{te}(x)}{p_{tr}(x)}\right) dx - \int p_{te}(x) \ln\left(w(x)\right) dx \end{aligned} \tag{2.3}$$

It should be noted that the Kullback-Leibler divergence can be lower bounded in terms of the Hellinger distance, H, (that is used in imbalanced learning methods) as [125]:

$$D_{KL}(Q||P) \geq 2H^2(P,Q) \tag{2.4}$$

A related measure is the two sample Kolmogorov-Smirnov (KS) distance and can be more applicable as a nonparametric distance measure that does not assume a Gaussian or any predefined distribution for the data. KS compares the cumulative distributions of two datasets where the cumulative distribution function $F_n(x)$ is defined as:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \leqslant x} \tag{2.5}$$

where $I_{X_i \leqslant x}$ is the indicator function, equal to 1 if $X_i \leqslant x$ and equal to 0 otherwise. The KS statistic for the given cumulative distribution function $F(x)$ is defined as:

$$D_n = \sup_x |F_n(x) - F(x)| \tag{2.6}$$

where $\sup_x$ is the supremum of the set of distances.

Other distance measures include "Information Gain" or "Mutual Information" which is popular as it used as the splitting criterion in CART. In probability theory and information theory, the mutual information of two random variables is a quantity that measures the mutual dependence of the two random variables. The most common unit of measurement of mutual information is the bit, when logarithms to the base 2 are used. Formally,

the mutual information of two discrete random variables X and Y can be defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{2.7}$$

Where p(x,y) is the joint probability distribution function of X and Y. The marginal probability distribution functions of X and Y are respectively p(x) and p(y). This splitting criterion relates directly to KL divergence as:

$$I(X;Y) = D_{\mathrm{KL}}(p(x,y)||p(x)p(y)) \tag{2.8}$$

An information-theoretic metric learning method in [27] minimized the LogDet divergence [80] as a Bregman optimization problem. This metric was used for transfer learning in [93] where they proposed to directly learn a distance metric across different domains. In [65], this work was extended with the addition of a non-linear kernel for improved performance. The authors in [84] proposed to learn metrics to leverage the information shared between the training data from two image categories to develop a cross-category ensemble for target classification. A transfer metric for learning task relationships was proposed in [130] to discover the task relationship between all source tasks and the target task. In [128], a distance metric is modified to map the target domain by using existing distance metrics which are pre-learned from the source domains.

A universal kernel mapping data into a Reproducing Kernel Hilbert Space (RKHS) can be used to estimate the high order momentums of the original data using only the first momentum, mean, of that RKHS. As the number of dimensions produced by the kernel increases, the capability of its mean to recover the moments of original data also increases and thus the mean can more accurately reconstruct that original data. This interesting trait of kernels equates minimizing the difference between the means of source and target domains in the RKHS, also known as Maximum Mean Discrepancy (MMD) [45] to

mapping the data to a shared distribution. MMD is a nonparametric estimate criterion of distance between distributions of datasets and is a relevant criterion for comparing distributions because it does not require explicit density estimation. The authors in [77] learned a low-dimensional space to reduce the distribution difference between different domains for transfer learning by exploiting Maximum Mean Discrepancy Embedding (MMDE) [11] which was originally designed for dimensionality reduction. This was improved in [78] where a more efficient feature extraction algorithm, known as Transfer Component Analysis (TCA), was applied to overcome the computationally expensive cost of MMDE.

Positive and Negative Transfer

Positive transfer occurs when learning in one context improves performance in some other context. For instance, speakers of one language find it easier to learn a related second languages [114]. For this multi-lingual language learning example, negative transfer occurs when learning in one context impacts negatively on performance in another. For example, despite the generally positive transfer among related languages, contrasts of pronunciation, vocabulary, and syntax generate stumbling blocks. Learners commonly assimilate a new language's phonetics to crude approximations in their native tongue and use word orders carried over from their native tongue (accent).

Transfer learning algorithms have to incorporate positive transfer while simultaneously rejecting negative transfer. A careful selection of source tasks and data is essential to avoid negative transfer as this knowledge transfer not only fails to improve learning, but actually hinders a learner's performance. An empirical study demonstrated that negative transfer occurs when tasks are too dissimilar [91]. Transfer learning algorithms should only select source knowledge that is relevant to the target task and the authors in [95] selected only source instances based on the likelihood that they will correctly label an

instance from the target domain. In [34], the source selection techniques were divided into methods that perform manual selection [119, 5] or ones that do so with task clustering [6, 123]. There is no set rule to determine how and if negative transfer will occur and the general rule is to assume that near transfer is less likely to induce negative transfer than far transfer and thus the same measures of transfer distance are applicable.

2.5 Learning with "Absolute Rarity"

A "Rare Dataset"[3] is a label-skewed and small dataset and presents a set of challenges that are not studied in existing literature. This section examines the parameters that are relevant for the study of "Rare Datasets".

2.5.1 Effect of Data Size on Learning

In a Balanced Dataset

The first impediment to learning with "Absolute Rarity" is the fact that the small size of the training set, regardless of imbalance, impedes learning. When the number of training examples is not *adequate* to generalize to instances not present in the training data, it is not theoretically possible to use a learning model as the model will only overfit the training set. The term "adequate" is a broad term as many factors have to be considered including data complexity, number of dimensions, data duplication, and overlap complexity [50]. Computational learning theory [74] provide a general outline to estimate the difficulty of learning a task, the required number of training examples, the expected learning and generalization error and the risk of failing to learn or generalize. A study in [7] found that the size of training set is the factor with the most significant impact on classification performance. Figure 2.5 depicts 4 different algorithms that are trained at different training set sizes and demonstrates that increasing the training sets' size improves the classification performance of all algorithms. To assert that increasing

[3] A "Rare Dataset" refers to a dataset with "Absolute Rarity"

the number of training examples, combined with an error minimizing classifier, yields results where the training and the generalization errors are similar is an intuitive and crucial finding as is demonstrates than the choice of a classification model is less important than the overall size of the training set. This is a major driving force in the machine learning domain where there is a shift to "Big Data" assisted with the availability of cluster computing frameworks such as MapReduce [28].

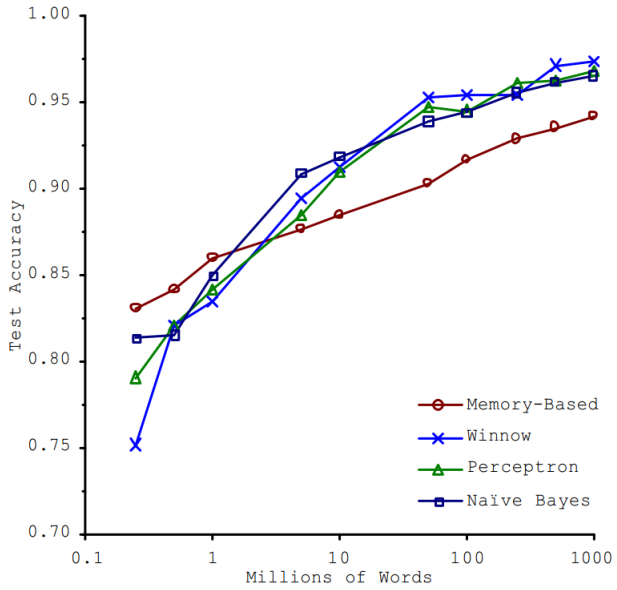


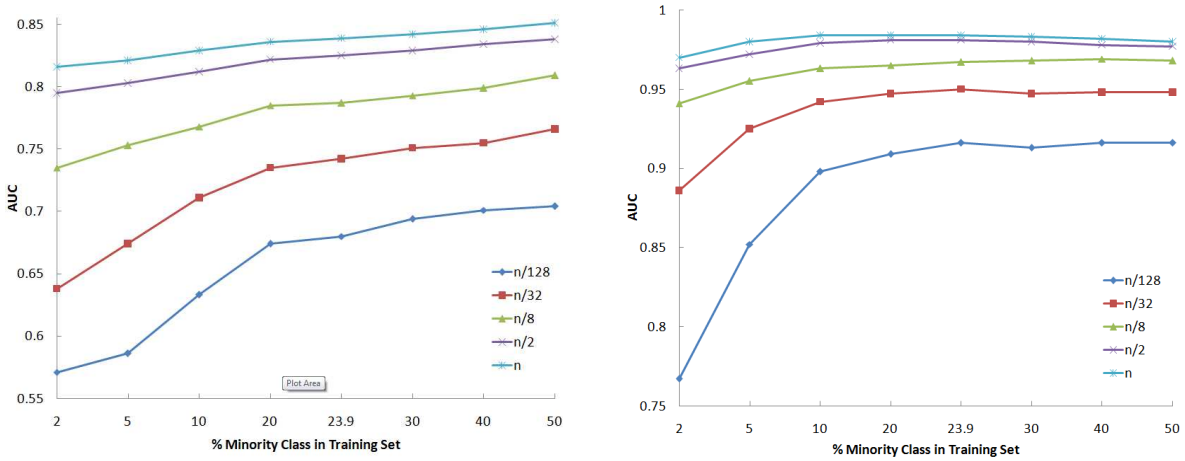
Figure 2.5: Learning Curves for Confusion Set Disambiguation.

In an Imbalanced Dataset

The second impediment to learning with "Absolute rarity" is the between-class imbalance where a majority of samples belong to an overrepresented class and a minority of samples belong to an underrepresented class [50]. The imbalanced classification study in [117] found that the most significant effect on a classifier's performance in an imbalanced classification problem is **not** the ratio of imbalance but it is the number of samples in the training set. This is an important finding as it demonstrates that the lack of data in "Absolute Rarity" intensifies the label imbalance problem. As the number of

the training examples increased, the error rate caused by imbalance decreased [55] and thus increasing the number of training samples makes the classifiers less sensitive to the between-class imbalance [117].

Figure 2.6 demonstrates how the lack of training examples degrades learning in an imbalanced dataset [117]. The ROC curve illustrates the performance of a binary classifier where the x-axis represents the False Positive Rate (1-Specificity) and the y-axis represents the True Positive Rate and is an accepted metric in imbalanced learning problems. AUC is a simple summary of the ROC performance and can be calculated by using the trapezoidal areas created between ROC points and is thus equivalent to the Wilcoxon-Mann-Whitney statistic [26]. Figure 2.6 presents the Area Under the ROC curve (AUC) [12] results in [117] where a classifier was trained for two imbalanced datasets [42] with different subsets of training sets (with a total of n samples). The results demonstrate that increasing the size of the training set directly improves learning for imbalanced datasets.



(a) Adult (b) Covertype

Figure 2.6: AUC for imbalanced datasets at different training sample sizes

2.5.2 Learning Bounds

Domain Adaptation Learning Bounds

Researchers have theoretically analyzed the target error bounds for the related problem of domain adaptation. By assuming the distribution of the target domain to be a weighted combination of the source distributions, the authors in [73] proved that the loss of the target classifier can be upper bound where there exists a *distribution weighted combining rule* that has a loss of at most ε with respect to any target mixture of the source distributions. Alternatively, the authors in [22] introduced a PAC-style model of learning from multiple sources where they assumed that the distributions of multiple input sources are the same across sources but each source may have its own deterministic labeling function. They derived a bound on the error of the target domain by minimizing the empirical error on the uniformly weighted data from any subset of the sources.

Hoeffding's Inequality for Imbalanced datasets

Let $Z_1, \ldots, Z_n$ be random independent, identically distributed variables with expected value $E[Z]$, such that $0 \leq Z_i \leq 1$, than:

$$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - E[Z]\right| > \varepsilon\right] \leq \delta = 2^{-2m\varepsilon^2} \tag{2.9}$$

To get a confidence of δ for error ε, we calculate the required number of samples m where $2^{-2m\varepsilon^2} \geq \delta$. Thus, given a hypothesis space with $|H|$ complexity, we require:

$$m \geq \frac{1}{2\varepsilon^2}\left(\ln(|H|) + \ln\left(\frac{1}{\delta}\right)\right) \tag{2.10}$$

Once m is set using Equation 2.10, we can calculate with probability at least $1-\delta$ the difference between the empirical and actual mean as:

$$\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - E\left[Z\right]\right| \leq \varepsilon \tag{2.11}$$

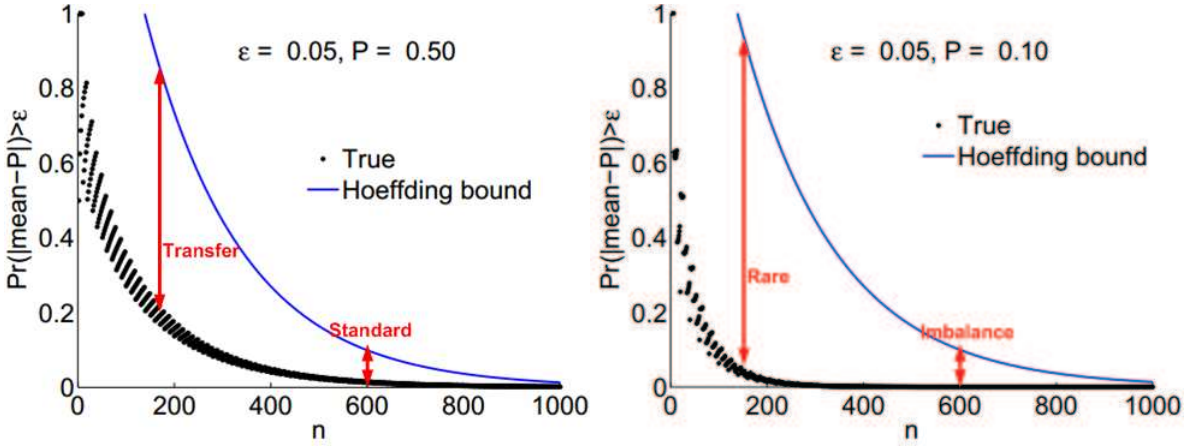
If we draw m samples, then with probability at least $1-\delta$, the difference between the empirical mean, $\frac{1}{m}\sum_{i=1}^{m} Z_i$, and the true mean, $E\left[Z\right]$, is at most ε, where:

$$\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - E\left[Z\right]\right| \leq \varepsilon \leq \sqrt{\frac{1}{2m}\left(\ln\left(|H|\right) + \ln\left(\frac{1}{\delta}\right)\right)} \tag{2.12}$$

Observing Equation 2.12, it be be deduced that Hoeffding's inequality ignores information about the variance. Hoeffding's inequality does not use any distributional properties, such as the distribution's mean or variance as it assumes binomial random variables. On the other hand, imbalanced data does not follow a binomial distribution and thus the Hoeffding's bounds are altered.

In most problems, the aim is not to obtain a function that performs well on training data but rather to estimate a function (using training data) that performs well on future unseen test data. This is accomplished by minimizing empirical risk on the training set while choosing a function of small complexity. The rationale behind this approach is that the empirical risk converges (uniformly) to the true unknown risk. Figure 2.7 shows how the different datasets relate to Hoeffding's Inequality. Both figures have the same error rate, or deviation, but Figure 2.2-a depicts a balanced distribution while Figure 2.2-b depicts an imbalanced distibution. In an imbalanced learning problem, the error bounds are label-dependent [62]. Give the label dependent error, ε^l, the error bounds are calculated as:

$$m^l \geq \frac{1}{2(\varepsilon^l)^2}\left(\ln\left(|H|\right) + \ln\left(\frac{1}{\delta}\right)\right) \tag{2.13}$$



(a) (b)

Figure 2.7: (a) Generalization Bound on a Balanced Distribution [37]. (b) Generalization Bound on an Imbalanced Distribution.

This is an important property, as the label-dependent error bounds require a label-dependent error minimization. In a datasets with "Absolute Rarity", a label independent error bound error has to be formulated.

Theorem 1: *The number of minority samples bounds the structural risk.*

Proof. Assuming no knowledge of training error, we can apply Hoeffding's Inequality for an imbalanced dataset by treating each class as an independent binomial distribution where:

$$\begin{aligned} \Pr\left[\left|\frac{1}{m_{majority}}\sum_{i=1}^{m} Z_i - E\left[Z\right]\right| > \varepsilon\right] &\leq \delta = |H|\, 2^{-2m_{majority}\varepsilon^2}, i \in majority \\ \Pr\left[\left|\frac{1}{m_{minority}}\sum_{i=1}^{m} Z_i - E\left[Z\right]\right| > \varepsilon\right] &\leq \delta = |H|\, 2^{-2m_{minority}\varepsilon^2}, i \in minority \end{aligned} \tag{2.14}$$

An imbalanced dataset has an un-equal distribution of labels as: $m_{minority} < m_{majority}$. Hoeffding's Inequality for a label-independent error will be bound by:

$$\begin{aligned} \Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - E\left[Z\right]\right| > \varepsilon\right] \leq \delta &= \min\left[|H|\, 2^{-2m_{majority}\varepsilon^2}, |H|\, 2^{-2m_{minority}\varepsilon^2}\right] \\ &= |H|\, 2^{-2m_{minority}\varepsilon^2} \end{aligned} \tag{2.15}$$

Equation 2.15 proves that the number of minority samples bounds the structural risk. Given the extremely low number of minority samples in a dataset with "Absolute Rarity", it is essential to get another source of information for generalization. Transfer learning provides a perfect opportunity to construct a hypothesis capable of generalization.

2.6 Visualization Diagram for Rare Datasets

For imbalanced datasets, F-measure [105] is a popular metric that is used to evaluate the performance of a classifier. It is calculated as:

$$F = \frac{2PR}{P + R} \tag{2.16}$$

P refers to *Precision* or *Positive Predicted Value(PPV)*. Precision generally refers to the fraction of retrieved instances that are relevant, and is calculated as:

$$P = PPV = \frac{TP}{TP + FP} \tag{2.17}$$

R refers to *Recall* or *True Positive Rate(TPR)*. Recall generally refers to the fraction of relevant instances that are retrieved, and is calculated as:

$$R = TPR = \frac{TP}{TP + FN} \tag{2.18}$$

F-measure is an acronym for the $F_1 - score$ which can be interpreted as a harmonic mean of the *Precision* and *Recall*. The $F_1 - score$ reaches its best value at 1 and worst score at 0. Equation 2.16 equally weighs Precision and Recall and this metric can be deceiving since a classifier with high Precision and low Recall give the same F-measure result of a classifier with low Precision and high Recall. Table 2.2 demonstrates how 3 algorithms can have the same $F_1 - score$ with different performance for individual classes.

For imbalanced dataset methods, researchers can make the assumption that an increase

Table 2.2: Classification Performance in an imbalanced dataset

	Precision	*Recall*	*F-Measure*
Algorithm 1	*0.2*	*0.6*	*0.3*
Algorithm 2	*0.3*	*0.3*	*0.3*
Algorithm 3	*0.45*	*0.225*	*0.3*

in the F-measure is satisfactory since there is enough data in the training set to assume that the majority instances are in such abundance that Recall is never degraded and the increased F-measure comes from a better Precision score. In a dataset with "Absolute Rarity", an increase in both Precision and Recall is required (although a degradation for the majority is almost unavoidable as the classification algorithm attempts to fit the minority labels). For a more appropriate measure that is suitable for "Absolute Rarity", we investigated the generalized F-Score which is calculated as:

$$\begin{aligned} F_\beta &= \frac{\left(1+\beta^2\right)(Specificity)(Sensitivity)}{Specificity+\beta^2 Sensitivity} \\ &= \frac{\left(1+\beta^2\right)\cdot \mathrm{TP}}{(1+\beta^2)\mathrm{TP}+\beta^2\mathrm{FN}+\mathrm{FP}} \end{aligned} \tag{2.19}$$

The F-Score (F_β) was derived so that β weighs the balance between Precision and Recall as its value attaches β times as much weight to Recall over Precision. For example, an F_2 measure weighs Recall twice as much as Precision and $F_{0.5}$ measure weighs Precision twice as much as Recall. For an all purpose metric where both Precision and Recall are important, it is important that an algorithm minimizes the variation between the F_β and the $F_{1/\beta}$ measures. To visualize how an algorithm learns on a dataset with "Absolute Rarity", we propose the F_λ-Plots. F_λ-Plots allow the visualization of how well a classification algorithm performs at different Precision and Recall weights. λ will indicate the scale of maximum importance for Precision and Recall. The x-axis would be set to $\left(\frac{\log_{10}(\beta)}{log_{10}(\lambda)}\right)$ and the Y-axis would be the F-measure value at the specific value of β. Using $\left(\frac{\log_{10}(\beta)}{log_{10}(\lambda)}\right)$ for the x-axis normalizes the scale between -1 and +1. For example, for an

F_{10}-Plot, the F-measure value at +1 would weigh recall 10 times more than precision, at value 0 they will have equal weights and at -1 the F-measure value would weigh precision 10 times more than recall.

In Table 2.2, the F-measure statistic gives the false conclusion that all 3 algorithms are equally fit to handle "Absolute Rarity" since they all have the same F-measure. The algorithms actually are not similar and handle Precision and Recall very differently.

Alternatively, Figure 2.8 presents the F_{∞}-Plot for the three algorithms. Figure 2.8 captures the retrieval effectiveness of an algorithm in one figure and thus presents a more comprehensive visual view of the performance of a classifier. The figure demonstrates visually how 3 algorithms have the same F-Measure (as when the x-axis is at zero) while they produce Precision and Recall results that are completely different. The plot also demonstrates that for an ($\lambda \rightarrow \infty$) plot, the plot converges to the actual Precision and Recall metrics as the F_{∞}-Plot converges to Recall at +1 and Precision at -1.

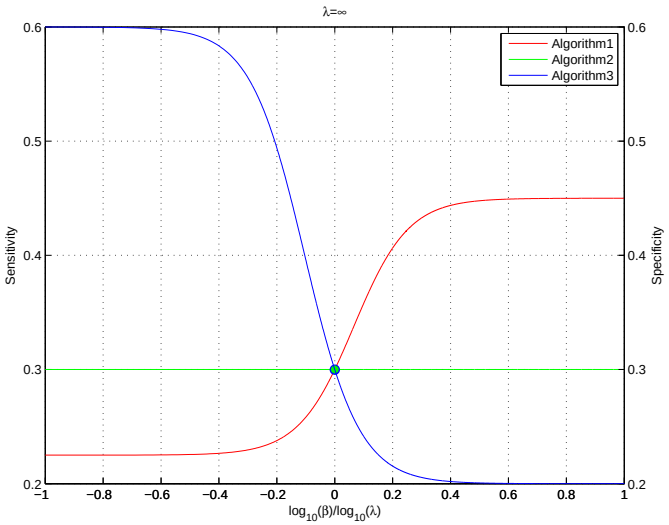


Figure 2.8: F_{∞} plot for 3 algorithms with same F-measure but different statistics

The figure demonstrates that while all three algorithms have the same F-measure, "Algorithm 2" handles positive and negative samples equally as it is the most balanced. On the contrary, "Algorithm 1" and "Algorithm 3" have opposite performances.

2.7 Conclusion

In this chapter, we provided an overview of the methods that are critical for learning and evaluation of datasets with "Absolute Rarity". We demonstrated that theoretically, learning can not generalize and an additional source of information is required to generalize as learning can only overfit.

CHAPTER 3

ADAPTIVE BOOSTING FOR TRANSFER LEARNING USING DYNAMIC UPDATES

3.1 Introduction

Instance transfer learning methods utilize labeled examples from one domain to improve learning performance in another domain via knowledge transfer. Boosting-based transfer learning algorithms are a subset of such methods and have been applied successfully within the transfer learning community. In this chapter, we address some of the weaknesses of such algorithms and extend the most popular transfer boosting algorithm, TrAdaBoost [24]. We incorporate a dynamic factor into TrAdaBoost to make it meet its intended design of incorporating both AdaBoost [43] and the "Weighted Majority Algorithm" [70]. We theoretically and empirically analyze the effect of this important factor on the boosting performance of TrAdaBoost and we apply it as a "Correction Factor" that significantly improves the classification performance. Our experimental results on several real-world datasets demonstrate the effectiveness of our framework in obtaining better classification results.

3.1.1 Notations

Consider a domain (D) comprised of instances ($X \in \mathbb{R}^d$) with d features. We can specify a mapping function, F, to map the feature space to the label space as "$X \rightarrow Y$" where $Y \in \{-1, 1\}$. We will denote the domain with n auxiliary instances as the source domain (X_{src}) and define (X_{tar}) as the target domain with $m \ll n$ instances. Instances that belong to the majority class will be defined as (X_{majority}) and those that belong to the minority class will be defined as (X_{minority}). n^l is the number of source samples that belong to label l while ε^l is the error rate for label l. N defines the total number of

boosting iterations, w is a weight vector. A weak classifier at a given boosting iteration (t) will be defined as $\ddot{f}^t$ and its classification error is denoted by ε^t. $\mathbb{1}$ is an indicator function and is defined as:

$$\mathbb{1}\left[y \neq \ddot{f}\right] = \begin{cases} 1 & y \neq \ddot{f} \\ 0 & y = \ddot{f} \end{cases} \tag{3.1}$$

Table 3.1 presents a summary of notations used for chapters 3-5.

Table 3.1: Summary of the Notations

Notation	Description
X	feature space, $X \in \mathbb{R}^d$
Y	label space $= \{-1, 1\}$
d	number of features
F	mapping function $X \rightarrow Y$
D	domain
src	source (auxiliary) instances
tar	target instances
maj	majority class
min	minority class
ε^t	classifier error at boosting iteration "t"
w	weight vector
N	number of iterations
n	number of source instances
m	number of target instances
t	index for boosting iteration
$\ddot{f}^t$	weak classifier at boosting iteration "t"
$\mathbb{1}$	Indicator function

3.2 Boosting for Instance Transfer

3.2.1 Boosting

Ensemble methods [30] are learning algorithms that construct a set of classifiers and then classify new data points by taking a weighted vote of the combination of all the classifiers' predictions. The original ensemble method is Bayesian averaging where an *Bayes Optimal Classifier* is an ideal ensemble where each generated hypothesis is given a vote proportional to the likelihood that the training dataset would be sampled from a

system where the hypothesis is correct. More recent algorithms include error-correcting output coding [31], Bagging [14] and Boosting [43]. Some intuitive explanations to the advantages of boosting with ensemble learners:

1. The training dataset might not be fit for a single best learner and thus a combination of learners can improve the overall classification outcome.

2. The error convergence of most algorithms is not perfect (nor unique for most of them). For example, even if there exists a unique optimal hypothesis, it might be difficult to discover as learning algorithms find sub-optimal (and local) hypotheses. Ensemble methods can compensate with the combination of several suboptimal hypotheses.

3. The hypothesis space of the training set might not exactly depict the true target function and thus an ensemble of learners can give better approximation .

4. Ensemble learners can boost simple classifiers (including linear classifiers) to produce complex (non-linear) classifiers. For example, classification boundaries of decision stumps [52] (one level decision trees) are linear segments and a single decision tree cannot lead to a good result yet a good approximation can be achieved by combining a set of decision trees.

3.2.2 Boosting for Transfer Learning

Boosting-based transfer learning algorithms apply ensemble methods to both source and target instances with an update mechanism that incorporates only the source instances that are useful for target instance classification. These methods perform this form of mapping by giving more weight to source instances that improve target training and decreasing the weights for instances that induce negative transfer.

TrAdaBoost [24] is the first and most popular transfer learning method that uses boost-

ing as a best-fit inductive transfer learner. As outlined in Algorithm 1[1], TrAdaBoost trains the base classifier on the weighted source and target set in an iterative manner. After every boosting iteration, the weights of misclassified target instances are increased and the weights of correctly classified target instances are decreased. This target update mechanism is based solely on the training error calculated on the normalized weights of the target set and uses a strategy adapted from the classical AdaBoost [43] algorithm. The Weighted Majority Algorithm (WMA) [70] is used to adjust the weights of the source set by iteratively decreasing the weight of misclassified source instances by a constant factor, set according to [70], and preserving the current weights of correctly classified source instances. The basic idea is that the source instances that are not correctly classified on a consistent basis would converge to zero and would not be used in the final classifier's output since that classifier only uses boosting iterations $\frac{N}{2} \rightarrow N$ for convergence [24].

Algorithm 1 TrAdaBoost

Require: Source and Target Instances : $\mathrm{D} = \{(x_{src_i}, y_{src_i}) \cup (x_{tar_i}, y_{tar_i})\}$, Maximum number of iterations(N), Base Learning algorithm($\ddot{f}$)

Ensure: Weak classifiers for boosting iterations : $\frac{\mathrm{N}}{2} \rightarrow \mathrm{N}$

Procedure:

1: **for** $t = 1$ to N **do**
2: Find the candidate weak learner for $\ddot{f}^t : X \rightarrow Y$ that minimizes error for D
3: Update source weights via WMA to decrease weights of misclassified instances
4: Update target weights via AdaBoost using target error rate (ε^t_{tar})
5: Normalize weights for D
6: **end for**

TrAdaBoost has been extended to many transfer learning problems. A multi-source learning [126] approach was proposed to import knowledge from many sources. Having multiple sources increases the probability of integrating source instances that are better fit to improve target learning and thus this method can reduce negative transfer. TrAdaBoost has also been extended in [81] by incorporating AdaBoost.R2 [32] for regression

[1]Detailed algorithm can be found in the referenced paper

transfer . A model-based transfer in "TaskTrAdaBoost" [127] extends this algorithm to transferring knowledge from multiple source tasks to learn a specific target task. Since closely related tasks share some common parameters, suitable parameters that induce positive transfer are integrated from multiple source tasks. Application of TrAdaBoost include multi-view surveillance for highway traffic [127] or head-pose estimation [109]. Other applications include visual tracking [71], cross category visual learning [84], text classification [24] and several other problems [79].

Some other methods use AdaBoost's update mechanism for target and source instances. In ExpBoost [88], a separate hypothesis is learned for each of the source datasets and one learner is constructed using only the target instances. At each boosting iteration, ExpBoost chooses to either use a hypothesis that is learned from a weighted source dataset or the one generated by the weighted target dataset. Picking a hypothesis is dependent on which learner produces the most accurate results. TransferBoost [35] is another method and is used for boosting when multiple source tasks are available. It boosts all source weights for instances that belong to tasks exhibiting positive transferability to the target task. TransferBoost calculates an aggregate transfer term for every source task as the difference in error between the target only task and the target plus each additional source task. TransferBoost claims some improvements over TrAdaBoost where boosting is applied by reweighing instances from each source task based on their aggregate transfer to the target task. TransferBoost adds the term, α_t^i, to the update weight of source samples and updates these source weights in a manner similar to AdaBoost but with a slight modification as:

$$\begin{aligned} w_{src_i}^{t+1} &= w_{src_i}^{t} e^{\left(\beta_{tar}^{t} y_{src_i} \ddot{f}_{src_i}^{t} + \alpha_t^i\right)} \\ \beta_{tar}^{t} &= \ln\left(\frac{\varepsilon_{tar}^{t}}{1-\varepsilon_{tar}^{t}}\right) \\ \alpha_i^t &= \varepsilon_{tar}^{t} - \varepsilon_{(tar+src_i)}^{t} \end{aligned} \tag{3.2}$$

AdaBoost was also extended in [107] for concept drift as a fixed cost is pre-calculated using Euclidean distance (as one of two options) as a measure of relevance between source and target distributions as

$$C_j = \frac{\sum_{i,\left[y_{src_i} \neq y_{tar_j}\right]}^{n} \left(\sqrt{\left(x_{src_i} - x_{tar_j}\right)^2}\right)}{\sum_{i,\left[y_{src_i} = y_{tar_j}\right]}^{n} \left(\sqrt{\left(x_{src_i} - x_{tar_j}\right)^2}\right)} \tag{3.3}$$

The resulting relevance ratio is then normalized to span $[0, 1]$. This relevance ratio thus gives more weights to data that is near in the feature space and share a similar label. This ratio is finally incorporated to the update mechanism via AdaCost [98] for AdaC1 as:

$$w_{tar_j}^{t+1} = w_{tar_j}^{t} e^{-C_j \beta^t \mathbb{I}\left[y_{tar_j} = y_{tar_j}\right]} \tag{3.4}$$

In Equation 3.4, the samples that have have a higher cost are reduced by a smaller factor. However, the difference is expressed in exponential terms and is generally small. For a linear impact, AdaC2 is used as it directly applies the cost factor so the weight change is directly related to the relevance of the sample as:

$$w_{tar_j}^{t+1} = C_j w_{tar_j}^{t} e^{-\beta^t \mathbb{I}\left[y_{tar_j} = y_{tar_j}\right]} \tag{3.5}$$

Finally AdaC3 can be applied as a combination of Equation 3.4 and Equation 3.5. It is calculated as:

$$w_{tar_j}^{t+1} = C_j w_{tar_j}^{t} e^{-C_j \beta^t \mathbb{I}\left[y_{tar_j} = y_{tar_j}\right]} \tag{3.6}$$

Since AdaBoost based methods update the source weights via AdaBoost's update mechanism, they create a conflict within this update mechanism. A source task that is unrelated to the target task will exhibit negative transferability and its instances' weights would

be diminished by a fixed [107] or dynamic rate [35, 36] within AdaBoost's update mechanism. This update mechanism will be simultaneously increasing these same weights since AdaBoost increases the weights of misclassified instances. Furthermore, it can be noted that the weight update in TransferBoost for the source domain instances is a special case of the cost-sensitive boosting algorithm. When the parameter α_t^i, in the TransferBoost algorithm is set to a constant value at the individual instance level, then TransferBoost is analogous to cost-sensitive boosting. Because of the conflict within AdaBoost's update strategy, a source update strategy based on the Weighted Majority Algorithm would be more appropriate.

3.2.3 Weaknesses of TrAdaBoost

The main weaknesses of TrAdaBoost are highlighted in the list below:

1. ***Weight Mismatch:*** As outlined in [81], when the size of source instances is much larger than that of target instances, many iterations might be required for the total weight of the target instances to approach that of the source instances. This problem can be alleviated if more initial weight is given to target instances.

2. ***Disregarding First Half of Ensembles:*** Eaton and desJardins [35] list the choice to discard the first half of the constructed classifiers as one of TrAdaBoost's weaknesses since it is these classifiers that fit the majority of the data, with later classifiers focusing on "harder" instances. Their experimental analyses along with the analyses reported by Pardoe and Stone [81] and our own investigation show mixed results. This is the outcome of a final classifier that makes use of all ensembles and thus infers negative transfer introduced from non-relevant source instances whose weights had yet to converge to zero.

3. ***Introducing Imbalance:*** In [34], it was noted that TrAdaBoost sometimes yields a final classifier that always predicts one label for all instances as it substantially

unbalances the weights between the different classes. Dai et al. [24] re-sampled the data at each step to balance the classes and we also examined their algorithm's implementation and found out that they do add a step were they balance the labels' weights.

4. ***Rapid Convergence of Source Weights:*** This seems to be the most serious problem with TrAdaBoost. Various researchers observed that even source instances that are representative of the target concept tend to have their weights reduced quickly and erratically. This quick convergence is examined by Eaton and desJardins [35] as they observe that in TrAdaBoost's reweighing scheme, the difference between the weights of the source and target instances only increases and that there is no mechanism in place to recover the weight of source instances in later boosting iterations when they become beneficial. This problem is exacerbated since TrAdaBoost, unlike AdaBoost, uses the second half of ensembles when the weights of these source instances have already decreased substantially from early iterations. These weights may be so small that they become irrelevant and will no longer influence the output of the combined boosting classifier. This rapid convergence also led Pardoe and Stone [81] to the use of an adjusted error scheme based on experimental approximation.

3.3 Dynamic-Transfer Algorithm

The pseudo code of "Dynamic-TrAdaBoost" is presented in Algorithm 2. The method exploits transfer learning concepts to improve learning by allocating higher weights to the subset of auxiliary instances that is most likely to improve learning with positive transfer. The framework effectively combines the power of two boosting algorithms with AdaBoost [43] updating the target instances' weights and the Weighted Majority Algorithm (WMA) [70] updating the source instances' weights. The two algorithms operate

separately and are only linked in:

1. Line 4 (Normalization): Both algorithms require normalization. The combined normalization causes an anomaly that we will address in subsequent analysis.

2. Line 5: Infusing source with target for training is how transfer learning is induced from the auxiliary dataset.

The target instances are updated on lines (7,9,11) as outlined by AdaBoost [43]. The weak learner on line 5 finds the separating hyperplane that forms the classification boundary and is used to calculate the target's error rate ($\varepsilon^t_{tar} < 0.5$) on line 7. This error is used on line 9 to calculate ($\beta_{tar} > 1$) which is used to update the target weights on line 11 as:

$$w^{t+1}_{tar_j} = w^t_{tar_j} \beta_{tar}^{\mathbb{1}\left[y_{tar_j} \neq \ddot{f}^t_j\right]} \tag{3.7}$$

Similar to AdaBoost, a misclassified target instance's weight increases after normalization and would thus acquire more influence in the next iteration. Once boosting is completed, ($t = N$), the weak classifiers ($\ddot{f}^t$) weighted by β_{tar} are combined to construct a committee capable of non-linear approximation.

The source instances are updated on lines (2,10) as outlined by the Weighted Majority Algorithm [70]. WMA is a meta-learning algorithm that constructs an additive set of weak learners, where the number of mistakes for n source samples ($n\varepsilon^{WMA}$) is bound by the number of mistakes made by the best performing of the N weak classifiers ($n\varepsilon^{best}$) as:

$$n\varepsilon^{WMA} \leq \frac{n\varepsilon^{best} \ln\left(\beta^{-1}_{src}\right) + \ln(N)}{1 - \beta_{src}} \tag{3.8}$$

The static WMA update rate ($\beta_{src} < 1$) is calculated on line 2 and updates the source weights as:

$$w^{t+1}_{src_i} = w^t_{src_i} \beta_{src}^{\mathbb{1}\left[y_{src_i} \neq \ddot{f}^t_i\right]} \tag{3.9}$$

Contrary to AdaBoost, WMA decreases the influence of an instance that is misclassified and give it lower relative weight in subsequent iterations. This property is beneficial for transfer learning as a source instance's contribution to the weak classifiers is dependent on its classification consistency. A consistently misclassified instance's weight converges[2] and its influence diminishes in subsequent iterations. In Algorithm 2, the WMA update mechanism in Equation (3.9) is actually modified on line 10 to incorporate the cost C^t as a dynamic "Correction Factor". We will prove that this "Correction Factor" prevents the source instances' weights from early and improper convergence and thus improve positive transfer.

3.4 Theoretical Analysis

3.4.1 Overview

In this section, we analyze how our algorithm improves transfer learning with improved convergence properties. The combined normalization for AdaBoost and WMA presents an anomaly where source weights improperly converge and the transfer from source to target would be diminished. This anomaly diminishes the weight of source instances that are beneficial for training and thus will be referred to as "Weight Drift". We present a high level overview of "Weight Drift" in Figure 3.1. The figure outlines how the two algorithms operate on the source and target datasets and gives an overview of the factors that control the rate of "Weight Drift". There are 4 factors that affect the rate of convergence and these variables can increase the convergence rate of source instances that are helpful for transfer learning and diminish positive transfer. These factors are:

1. Number of boosting iterations.
2. Number of target instances.
3. Number of source instances.
4. Choice of weak learner.

[2]All mentions of "convergence" refer to a sequence (weight) that converges to zero.

Algorithm 2 Dynamic-Transfer

Require:

- Source domain instances $D_{src} = \{(x_{src_i}, y_{src_i})\}$
- Target domain instances $D_{tar} = \{(x_{tar_i}, y_{tar_i})\}$
- Maximum number of iterations : N
- Base learner : $\ddot{f}$

Ensure: Target Classifier Output : $\left\{ \dot{f} : X \to Y \right\}$

$$\dot{f} = sign\left[\prod_{t=\frac{N}{2}}^{N}\left(\beta_{tar}^{t}{}^{-\ddot{f}^{t}}\right) - \prod_{t=\frac{N}{2}}^{N}\left(\beta_{tar}^{t}{}^{-\frac{1}{2}}\right)\right]$$

Procedure:

1: Initialize the weights for vector D $= \{D_{src} \cup D_{tar}\}$, where: $w_{src} = \{w_{src}^1, \ldots, w_{src}^n\}, w_{tar} = \{w_{tar}^1, \ldots, w_{tar}^m\}, w = \{w_{src} \cup w_{tar}\}$

2: Set $\beta_{src} = \frac{1}{1+\sqrt{\frac{2\ln(n)}{N}}}$

3: **for** $t = 1$ to N **do**

4: Normalize Weights: $w = \frac{\mathrm{w}}{\sum_{i}^{n} \mathrm{w}_{src_i} + \sum_{j}^{m} w_{tar_j}}$

5: Find the candidate weak learner $\ddot{f}^t : X \to Y$ that minimizes error for D weighted according to w

6: Calculate the error of $\ddot{f}^t$ on D_{src}: $\varepsilon_{src}^t = \sum_{j=1}^{n} \frac{\left[w_{src}^j\right] \mathbb{1}\left[y_{src_j} \neq \ddot{f}_j^t\right]}{\sum_{i=1}^{n}\left[w_{src}^i\right]}$

7: Calculate the error of $\ddot{f}^t$ on D_{tar}: $\varepsilon_{tar}^t = \sum_{j=1}^{m} \frac{\left[w_{tar}^j\right] \mathbb{1}\left[y_{tar_j} \neq \ddot{f}_j^t\right]}{\sum_{i=1}^{m}\left[w_{tar}^i\right]}$

8: Set $C^t = 2\left(1 - \varepsilon_{src}^t\right)$. (Chapter 3) or Set $C^l = \left(1 - \varepsilon_{src}^l\right)$. (Chapter 4)

9: Set $\beta_{tar} = \dfrac{1 - \varepsilon_{tar}^t}{\varepsilon_{tar}^t}$

10: $w_{src_i}^{t+1} = C^t w_{src_i}^t \beta_{src}^{\mathbb{1}\left[y_{src_i} \neq \ddot{f}_i^t\right]}$ where $i \in D_{src}$

11: $w_{tar_j}^{t+1} = w_{tar_j}^t \beta_{tar}^{\mathbb{1}\left[y_{tar_j} \neq \ddot{f}_j^t\right]}$ where $j \in D_{tar}$

12: **end for**

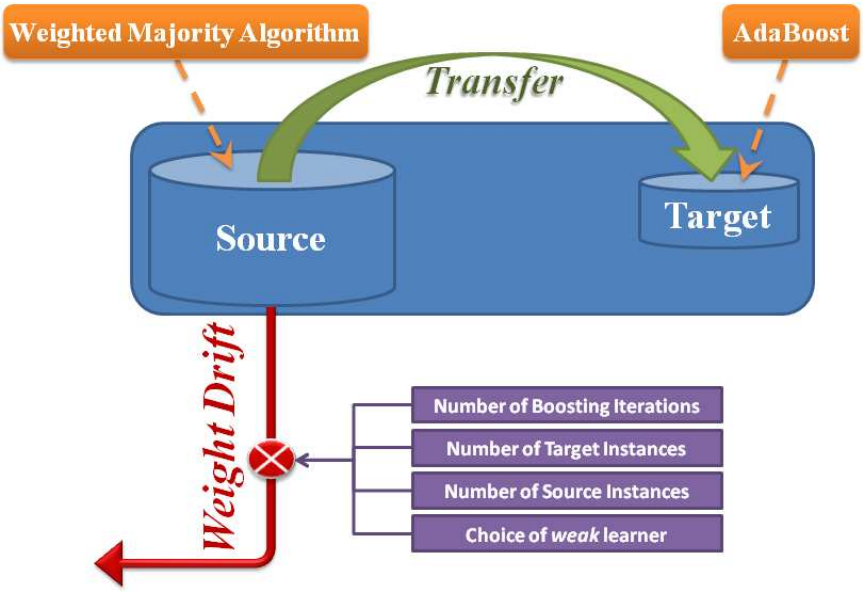


Figure 3.1: Overview of Weight Drift

3.4.2 "Weight Drift"

In this section, we theoretically analyze the factors that affect the convergence rate of source instances and present a method to prevent early convergence. We present an explanation of the weight update of instances after a single boosting iteration to aid in subsequent analysis and we introduce a proposition to allow for the calculation of the "Weight Drift" bounds.

Definition 1: Given k instances at iteration t with normalized weight w and update rate β, the sum of the weights after one boosting iteration with error rate (ε^t) is calculated as:

$$\sum_{i=1}^{k} w^{t+1} = kw^t(1-\varepsilon^t)+kw^t(\varepsilon^t)\beta \tag{3.10}$$

To demonstrate with an example, given $k = 10$ instances at iteration t with normalized weights $w = 0.1$, assume that weak learner $\ddot{f}$ correctly classifies 6 instances ($\varepsilon^t = 0.4$).

The sum of correctly classified instances at boosting iteration $t+1$ is calculated as:

$$\begin{aligned}\sum_{y=\ddot{f}^t} w^{t+1} &= 0.1\beta^0 + 0.1\beta^0 + 0.1\beta^0 + 0.1\beta^0 + 0.1\beta^0 + 0.1\beta^0 \\ &= 6(w^t)\beta^0 \left\{\text{since}\,(w^t = 0.1)\right\} \\ &= 10(0.6)(w^t) \\ &= kw^t(1-\varepsilon^t) \left\{\text{since}\,(\text{k} = 10, \varepsilon^t = 0.4)\right\}\end{aligned} \tag{3.11}$$

Alternatively, the sum of misclassified instances at boosting iteration $t+1$ is:

$$\begin{aligned}\sum_{y\neq\ddot{f}^t} w^{t+1} &= 0.1\beta^1 + 0.1\beta^1 + 0.1\beta^1 + 0.1\beta^1 \\ &= 4(w^t)\beta^1 \left\{\text{since}\,(w^t = 0.1)\right\} \\ &= 10(0.4)(w^t)\beta \\ &= kw^t(\varepsilon^t)\beta \left\{\text{since}\,(k = 10, \varepsilon^t = 0.4)\right\}\end{aligned} \tag{3.12}$$

Thus, the sum of weights at boosting iteration *"t+1"* is calculated as:

$$\begin{aligned}\sum_{i=1}^{k} w^{t+1} &= \sum_{y=\ddot{f}^t} w^{t+1} + \sum_{y\neq\ddot{f}^t} w^{t+1} \\ &= kw^t(1-\varepsilon^t) + kw^t(\varepsilon^t)\beta\end{aligned} \tag{3.13}$$

Proposition 1: All source instances are correctly classified by the weak learner:

$$y_{src_i} = \ddot{f}_i^t, \forall i \in \ \{1, \ldots, n\} \tag{3.14}$$

Equation (3.14) is analogous to:

$$\sum_{i=1}^{n} w^{t+1} = nw_{src}^t \left(1 - \varepsilon_{src}^t\right) + nw_{src}^t \left(\varepsilon_{src}^t\right) \beta_{src} = nw_{src}^t \tag{3.15}$$

Proposition 1 is not a realistic condition as it is not logical to assume an ideal classifier. We hold it as true to theoretically demonstrate that even under ideal conditions, ideal source samples, source weights still converge when they should not. A "Correction Factor" is calculated to correct for this improper convergence. It will be later demonstrated that this correction is inversely proportional to the weak learner's error and approaches unity (no correction needed) as error increases and deviates from this proposition.

While Proposition 1 can not be exactly manifested, there are three methods to approximate Equation 3.15 in Proposition 1 (hold the proposition as true) and minimize the weight differential between subsequent iterations:

$$\left(\sum_{i=1}^{n} w_{src_i}^{t+1} \approx \sum_{i=1}^{n} w_{src_i}^{t}\right) \tag{3.16}$$

1. Maximize the number of boosting iterations(N)

The total sum of source weights at iteration $t+1$ is calculated as the sum of P and Q where:

$$\begin{aligned} P &= \text{Sum of correctly classified source weights at iteration "t + 1"} \\ &= nw_{src}^{t}\left(1-\varepsilon_{src}^{t}\right)\beta_{src}^{\mathbb{1}\left[y_{src_i}=\ddot{f}_i^t\right]} \\ &= nw_{src}^{t}\left(1-\varepsilon_{src}^{t}\right)\ \left\{since\ \mathbb{1}\left[y_{src_i}=\ddot{f}_i^t\right]=0\right\} \end{aligned} \tag{3.17}$$

$$\begin{aligned} Q &= \text{Sum of misclassified source weights at iteration "t + 1"} \\ &= nw_{src}^{t}\left(1-\varepsilon_{src}^{t}\right)\beta_{src}^{\mathbb{1}\left[y_{src_i}=\ddot{f}_i^t\right]} \\ &= nw_{src}^{t}\left(\varepsilon_{src}^{t}\right)\beta_{src}^{\mathbb{1}\left[y_{src_i}\neq\ddot{f}_i^t\right]} \\ &= nw_{src}^{t}\left(\varepsilon_{src}^{t}\right)\beta_{src}\ \left\{since\ \mathbb{1}\left[y_{src_i}\neq\ddot{f}_i^t\right]=1\right\} \end{aligned} \tag{3.18}$$

Thus, the sum of source weights at boosting iteration *"t+1"* is $(S = P + Q)$ and is calculated as:

$$\begin{aligned} S &= \text{Sum of source weights at boosting iteration "t + 1"} \\ &= nw^t_{src}\left(1 - \varepsilon^t_{src}\right) + nw^t_{src}\left(\varepsilon^t_{src}\right)\beta_{src} \\ &= nw^t_{src}\left[1 - \left(\frac{\varepsilon^t_{src}}{1+\sqrt{\frac{N}{2\ln(n)}}}\right)\right] \quad \left\{since\, \beta_{src} = \frac{1}{1+\sqrt{\frac{2\ln(n)}{N}}}\right\} \end{aligned} \tag{3.19}$$

As the number of boosting iterations (N) increases, the assumptions in Proposition 1 can be approximated as:

$$\lim_{N\to\infty}\{S\} = \lim_{N\to\infty}\left\{nw^t_{src}\left[1 - \left(\frac{\varepsilon^t_{src}}{1+\sqrt{\frac{N}{2\ln(n)}}}\right)\right]\right\} = nw^t_{src} \tag{3.20}$$

2. Minimize the number of source instances (n)

This is evidently not desired as it negates the knowledge transfer. Examining Equation 3.20 shows that n has negligible effect because it changes logarithmically. For example, with N set to 30 boosting iterations, increasing the number of source instances (n) from 1,000 to 10,000 requires N to change from 30 to 40 and the denominator will stay unchanged.

3. Minimize the error rate $(\varepsilon^t_{src} \to 0)$

This is analogues to Equation 3.14 and can be controlled, to a certain extent, with the choice of learners. After the "Correction Factor" is calculated, it will be demonstrated that the applied correction is inversely proportional to the error rate and reaches unity (No Correction) as the error increases. This is an important property since the effect of the theorems utilizing this proposition is correlated to the proposition's validity.

Theorem 2 will examine the effect of the combined (source + target) normalization in line 4 of Algorithm 2 on transfer learning.

Theorem 2: *If no correction is included in Algorithm 2, source weights will improperly converge even when the instances are correctly classified.*

Proof. In the Weighted Majority Algorithm, the weights are updated as:

$$w_{src}^{t+1} = \begin{cases} \frac{w_{src}^t}{\sum\limits_{\{y_i=f_i\}} w_{src}^t + \sum\limits_{\{y_i\neq f_i\}} \beta_{src} w_{src}^t} & y_{src} = \ddot{f}^t \\ \\ \frac{\beta_{src} w_{src}^t}{\sum\limits_{\{y_i=f_i\}} w_{src}^t + \sum\limits_{\{y_i\neq f_i\}} \beta_{src} w_{src}^t} & y_{src} \neq \ddot{f}^t \end{cases} \tag{3.21}$$

Equation 3.21 shows that the weights for source instances that are correctly classified should not change since:

$$w_{src}^{t+1} = \frac{w_{src}^t}{\sum\limits_{i=1}^{n} w_{src_i}^t} = w_{src}^t \tag{3.22}$$

Without correction, the normalized source weights in Algorithm 2 are updated as:

$$w_{src}^{t+1} = \frac{w_{src}^t}{\sum\limits_{i=1}^{n} w_{src_i}^t + \sum\limits_{j=1}^{m} w_{tar_j}^t \left(\frac{1-\varepsilon_{tar}^t}{\varepsilon_{tar}^t}\right)^{\mathbb{1}\left[y_{tar_j}\neq \ddot{f}_j^t\right]}} \tag{3.23}$$

Equation 3.23 shows that, without correction, correctly classified source weights would still converge in direct correlation to:

$$\sum_{j=1}^{m} w_{tar_j}^t \left(\frac{1-\varepsilon_{tar}^t}{\varepsilon_{tar}^t}\right)^{\mathbb{1}\left[y_{tar_j}\neq \ddot{f}_j^t\right]} \tag{3.24}$$

Since all source weights persistently converge, all target weights would inversely increase since $(nw_{src}^{\mathrm{t}} + mw_{tar}^t) = 1$. This anomaly will be referred to as "Weight Drift" since weight entropy drifts from source to target instances. "Weight Drift" negates transfer since the final classifier is comprised of the cascade of weak learners constructed in boosting iterations $\frac{N}{2} \rightarrow N$ (where the source instances' weights could have already converged).

With converged source weights, Algorithm 2 becomes analogous to standard AdaBoost algorithm with target instances and no transfer learning.

□

Theorem 2 examined the cause of "Weight Drift" and theorem 3 will outline the factors that control it.

Theorem 3: *For n source instances, "Weight Drift" is stochastic and its rate of convergence at iteration t (without correction) is bound by the number of target training samples (m) and the target error rate at that iteration (ε_{tar}^{t}).*

Proof. The fastest rate of convergence is achieved by minimizing the weight for each subsequent boosting iteration (w_{src}^{t+1}) as:

$$\min_{m,n,\varepsilon_{tar}^{t}} \left(w_{src}^{t+1}\right) = \frac{w_{src}^{t}}{\max\limits_{m,n,\varepsilon_{tar}^{t}} \left\{ \sum\limits_{i=1}^{n} w_{src_i}^{t} + \sum\limits_{j=1}^{m} w_{tar_j}^{t} \left(\frac{1-\varepsilon_{tar}^{t}}{\varepsilon_{tar}^{t}} \right)^{\mathbb{I}\left[y_{tar_j} \neq \ddot{f}_j^t\right]} \right\}} \tag{3.25}$$

Equation (3.25) shows that two factors can slow convergence:

1. Maximizing the weak learner's target error rate with $\varepsilon_{tar}^{t} \rightarrow 0.5$ (choosing an extremely weak learner or one that is only slightly better than random). Since the weak learner's input weights cannot be predicted for each iteration, the weak learner's error cannot be controlled and this factor will continue to induce a stochastic effect.

2. Decreasing the number of target samples m, since convergence rate accelerates when $m/n \rightarrow \infty$. Attempting to slow convergence by reducing the number of target instances is counterproductive as knowledge form the removed instances would be lost.

□

Theorem 3 demonstrated that a fixed cost cannot control the convergence rate since the cumulative effect of m, n, and ε_{tar}^{t} is stochastic. A dynamic term has to be calculated to compensate for "Weight Drift" at every iteration.

3.4.3 Correction Factor

In this section, we calculate a dynamic term to compensate for "Weight Drift". We refer to this term as a "Correction Factor" as it corrects for the improper convergence of the source instances' weights. This factor is dynamic and preserves the weights for instances that are consistently correctly classified so these instances can induce positive transfer.

Theorem 4: *A correction factor of* $2\left(1-\varepsilon_{tar}^{t}\right)$ *can be applied to the source weights to prevent their "Weight Drift" and make the weights converge as outlined by the Weighted Majority Algorithm.*

Proof. Un-wrapping the WMA source update mechanism of Equation (3.25), yields:

$$w_{src}^{t+1} = \frac{w_{src}^{t}}{\sum_{i=1}^{n} w_{src_i}^{t} + \sum_{j=1}^{m} w_{tar_j}^{t} \left(\frac{1-\varepsilon_{tar}^{t}}{\varepsilon_{tar}^{t}}\right)^{\mathbb{1}\left[y_{tar_j} \neq \ddot{f}_j^t\right]}} = \frac{w_{src}^{t}}{n w_{src}^{t} + A + B} \tag{3.26}$$

Where A and B are defined as:

$$\begin{aligned} A &= \text{Sum of correctly classified target weights at boosting iteration "t + 1"} \\ &= m w_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)\left(\frac{1-\varepsilon_{tar}^{t}}{\varepsilon_{tar}^{t}}\right)^{\mathbb{1}\left[y_{tar_j} = \ddot{f}_j^t\right]} \\ &= m w_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right) \left\{ since\ \mathbb{1}\left[y_{tar_j} = \ddot{f}_j^t\right] = 0 \right\} \end{aligned} \tag{3.27}$$

$$\begin{aligned} B &= \text{Sum of misclassified target weights at boosting iteration ``t + 1''} \\ &= mw_{tar}^{t}\left(\varepsilon_{tar}^{t}\right)\left(\frac{1-\varepsilon_{tar}^{t}}{\varepsilon_{tar}^{t}}\right)^{\mathbb{I}\left[y_{tar_j}\neq \ddot{f}_j^t\right]} \\ &= mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right) \ \left\{since\ \mathbb{I}\left[y_{tar_j}\neq \ddot{f}_j^t\right]=1\right\} \end{aligned} \tag{3.28}$$

Substituting for A and B, the source update is:

$$w_{src}^{t+1}=\frac{w_{src}^{t}}{nw_{src}^{t}+2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)} \tag{3.29}$$

We will introduce and solve for a correction factor C^t to equate $(w_{src}^{t+1}=w_{src}^{t})$ for correctly classified instances (as per the WMA).

$$w_{src}^{t}=w_{src}^{t+1}=\frac{C^{t}w_{src}^{t}}{C^{t}nw_{src}^{t}+2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)} \tag{3.30}$$

Solving for C^t:

$$C^{t}=\frac{2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)}{\left(1-nw_{src}^{t}\right)}=\frac{2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)}{mw_{tar}^{t}}=2\left(1-\varepsilon_{tar}^{t}\right) \tag{3.31}$$

□

Adding this correction factor to line 10 of Algorithm 2 equates its normalized update mechanism to the Weighted Majority Algorithm and subsequently prevents "Weight Drift".

Theorem 5: *Applying a correction factor of* $2\left(1-\varepsilon_{tar}^{t}\right)$ *to the source weights would cause the target weights to converge as outlined by AdaBoost.*

Proof. In AdaBoost, without any source instances ($n = 0$), target weights for correctly

classified instances would be updated as:

$$\begin{aligned} w_{tar}^{t+1} &= \frac{w_{tar}^{t}}{\sum_{j=1}^{m} w_{tar_j}^{t} \left(\frac{1-\varepsilon_{tar}^{t}}{\varepsilon_{tar}^{t}}\right)^{\mathbb{I}\left[y_{tar_j} \neq \ddot{f}_j^t\right]}} \\ &= \frac{w_{tar}^{t}}{A+B} = \frac{w_{tar}^{t}}{2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)} = \frac{w_{tar}^{t}}{2(1)\left(1-\varepsilon_{tar}^{t}\right)} \end{aligned} \tag{3.32}$$

Applying the "Correction Factor" to the source instances' weight update prevents "Weight Drift" and subsequently equates the target instances' weight update mechanism outlined in Algorithm 2 to that of AdaBoost since:

$$\begin{aligned} w_{tar}^{t+1} &= \frac{w_{tar}^{t}}{nw_{src}^{t}+2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)} = \frac{w_{tar}^{t}}{C^{t}nw_{src}^{t}+2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)} \\ &= \frac{w_{tar}^{t}}{2\left(1-\varepsilon_{tar}^{t}\right)nw_{src}^{t}+2mw_{tar}^{t}\left(1-\varepsilon_{tar}^{t}\right)} \\ &= \frac{w_{tar}^{t}}{2\left(1-\varepsilon_{tar}^{t}\right)(nw_{src}^{t}+mw_{tar}^{t})} = \frac{w_{tar}^{t}}{2\left(1-\varepsilon_{tar}^{t}\right)(1)} \end{aligned} \tag{3.33}$$

□

It was proven that a dynamic cost can be incorporated into Algorithm 2 to correct for weight drifting from source to target instances. This factor would ultimately separate the source instance updates which rely on the WMA and β_{src}, from the target instance updates which rely on AdaBoost and ε_{tar}^{t}. With these two algorithms separated, they can be joined for transfer learning by infusing "best-fit" source instances to each successive weak classifier.

3.5 Empirical Analysis

In this section, we provide empirical validation of our theorems. We demonstrate how Proposition 1 is valid for our analysis. We than demonstrate how a "Correction Factor" fixes the problem of "Weight Drift".

3.5.1 Analysis of Proposition 1

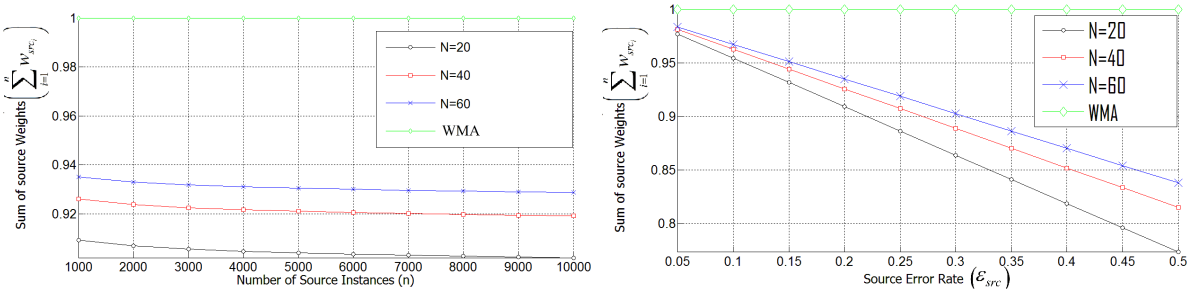
In Proposition 1, we stated and analyzed the three factors (ε^t_{src}, n, N) that minimize $\left(\frac{\varepsilon^t_{src}}{1+\sqrt{\frac{N}{2\ln(n)}}}\right)$ and strengthen Proposition 1. In this section, we empirically validate our analysis with two experiments. The first experiment analyzed the effects of N and n on Proposition 1. The source error rate (ε^t_{src}) was set to 0.2, while the number of source instances (n) varied from 1000 to 10,000 and $N \in \{20, 40, 60\}$. The plot in Figure 3.2-(a) demonstrates that the number of source instances (n) has little impact on the total sum of source weights while N has more significance. The second experiment considered the effects of N and ε^t_{src} on the sum of source weights. The number of source instances (n) was set to 1000 with $\varepsilon^t_{tar} \in \{0.05, \ldots, 0.5\}$ and $N \in \{20, 40, 60\}$. It can be observed in Figure 3.2-(b) that the error rate does have a significant effect on decreasing the total weight for $t+1$. This effect can be only partially offset via increasing N and it would require a large value of N for a reasonable adjustment. Since the source data comprises most of training data, we can generally expect $\varepsilon^t_{src} \approx \varepsilon^t_{tar}$. The effect of source error rates on Proposition 1 will be negated the fact that the correction factor, $C = 2\left(1 - \varepsilon^t_{tar}\right)$, is inversely proportional to ε^t_{tar} and its impact reaches unity (No Correction) as the target error rate increases:

$$\lim_{\varepsilon^t_{tar}\to 0.5}\{C\} = \lim_{\varepsilon^t_{tar}\to 0.5}\left\{2\left(1-\varepsilon^t_{tar}\right)\right\} \approx \lim_{\varepsilon^t_{src}\to 0.5}\left\{2\left(1-\varepsilon^t_{src}\right)\right\} = 1 \tag{3.34}$$

This is an important property because "Weight Drift" is most detrimental to learning at low error rates (where Proposition 1 was set).

3.5.2 "Weight Drift" and "Correction Factor"

The first experiment demonstrates the effect of "Weight Drift" on source and target weights. In Figure 3.3-a, the number of instances was constant ($n = 10000$,$m = 200$), the source error rate was set to zero as per Proposition 1 and the number of boosting



(a) (b)

Figure 3.2: The ratio of a correctly classified source weight for "$t+1$"/"t" (a) For different number of source instances and number of boosting iterations (N) (b) For different source error rate (ε^t_{src}) and number of boosting iterations (N)

iterations was set to $N = 20$. According to the WMA, the weights should not change since $\varepsilon^t_{src} = 0$. The ratio of the weights (with and without correction) to the weights of the WMA are plotted at different boosting iterations and with different target error rates $\varepsilon^t_{tar} \in \{0.1, 0.2, 0.3\}$. The experiment validates the following theorems:

1. With no correction, source weights converge even when correctly classified.

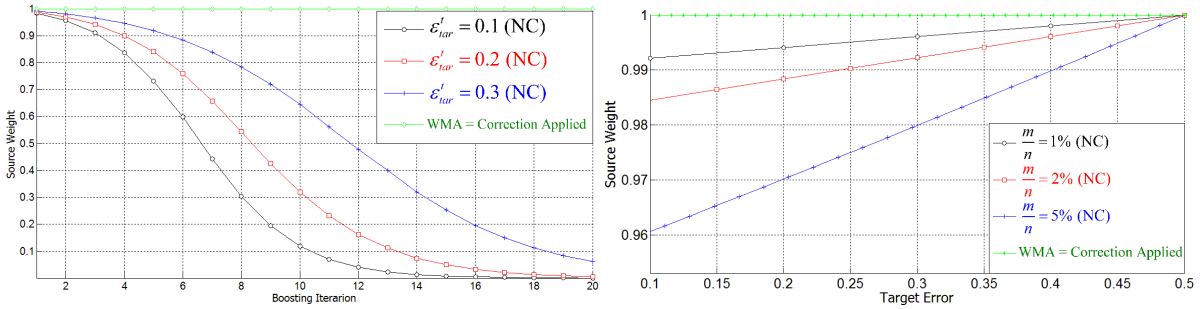
2. Applying our "Correction Factor" equates the weight update of Algorithm 2 to the WMA.

3. If correction is not applied, strong classifiers cause weights to converge at a faster rate than weak ones (Theorem 3).

The figure also demonstrates that for a weak learner with $\varepsilon^t_{tar} \approx 0.1$, if no correction is applied, we would not be able to benefit from all 10,000 source instances **although they were never misclassified**. The final classifier uses boosting iterations $N/2 \rightarrow N$, or $10 \rightarrow 20$, where the weights of ideal source instances would have already lost over 85% of their value. Correction conserved these instances' weights and thus helpful source instances would improve classification.

The second experiment validates the effect of the number of target instances, m, on the

convergence rate (Theorem 3). The number of source instances was set ($n = 1000$), while the number of target instances was varied $\frac{m}{n} \in \{1\%, 2\%, 5\%\}$ and plotted for $\varepsilon^t_{tar} \in \{0.1, \ldots, 0.5\}$. The plot in Figure 3.3-b shows how the source weights converge after a single boosting iteration and it can be observed that the rate of convergence is bound by m/n and the error rate ε_{tar} (which is also bound by m).



(a) (b)

Figure 3.3: The weights (relative to the WMA) for ideal source instances.(a) For 20 iterations with different error.(b) For 1 iteration with different target instances and error.

It should be noted that for both plots in Figure 3.3, the weight lost by the source instances is drifting to the target instances. The plots for the target weights would look inversely proportional to the plots in Figure 3.3 since:

$$\sum_{i=1}^{n} w^t_{src_i} + \sum_{j=1}^{m} w^t_{tar_j} = 1. \tag{3.35}$$

3.6 Experimental Results on Real-World Datasets

3.6.1 Experiment Setup

We tested several popular transfer learning datasets and compared AdaBoost [43] (using target instances), TrAdaBoost [24], TrAdaBoost with fixed costs of (1.1, 1.2, 1.3) and Dynamic-TrAdaBoost. Instances were balanced to have an equal number of positive and negative labels. We ran 30 iterations of boosting.

Base Learner $\left(\ddot{f}\right)$

We did not use decision stumps [52] (one level decision trees) as weak learners since the majority of training data belongs to the source and we need to guarantee an error rate of less than 0.5 on the target to avoid early termination of boosting (as mandated by AdaBoost). For example, applying decision stumps on data with 95% source and 5% target is not guaranteed (and will certainly not work for many boosting iterations) to get an error rate of less than 0.5 on target instances that compromise a small subset of the training data.

A weighted decision tree is a classifier expressed as a recursive partition of weighted instances [15]. The decision tree consists of nodes that form a root node, an internal node or a leaf node [87]. A "root" is one that has no incoming edges while all other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node while all other nodes are called leaves or decision nodes [90]. Usually the tree complexity is measured by one of the following metrics: the total number of nodes, total number of leaves, tree depth and number of attributes used. In our decision tree, the Gini Index [15] is used to split the internal nodes of the instance space into two or more sub-spaces. We used decision trees and applied a top-down approach where we trimmed the tree at the first node that achieved a target error rate that is less than 0.5 as displayed in Figure 3.4.

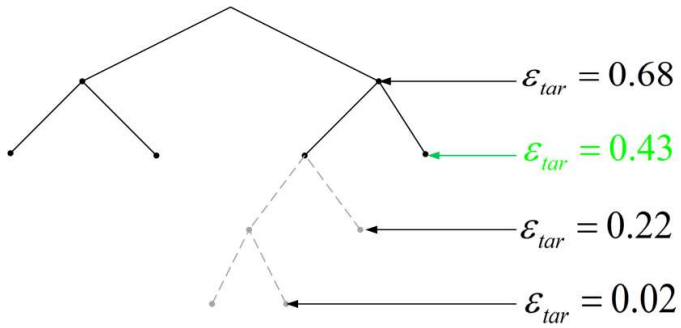


Figure 3.4: Trimmed Classification Trees

Cross Validation

We did not use standard cross validation methods since the target datasets were generally too large and did not need transfer learning to get good classification rates. We generated target datasets by using a small fraction for training and left the remainder for testing. A 2% ratio means that we had two target instances, picked randomly, for each 100 source instances and we used the remaining target instances for validation. We also used all the minority labels and randomly picked an equal number of instances from the majority labels to introduce variation in the datasets whenever possible. We ran each experiment 10 times and reported the average accuracy to reduce bias.

3.6.2 Real-World Datasets

20 Newsgroups

The 20 Newsgroups[3] dataset [67] is a text collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups. We generated 3 cross-domain learning tasks with a two-level hierarchy so that each learning task would involve a top category classification problem where the training and test data are drawn from different sub categories with around 2300 source instances (Rec vs Talk, Rec vs Sci, Sci vs Talk) as outlined in further detail in [23]. We used the "Threshold of Document Frequency" [2] with the value of 188 to maintain around 500 attributes. We used a 0.5% target ratio in our tabulated results and displayed results of up to 10% target ratio in our plots.

Abalone Age

The Abalone[4] dataset's features include the seven physical measurements of male, source, and female, target, abalone sea snails. The goal is to use these physical mea-

[3]http://people.csail.mit.edu/jrennie/20Newsgroups/
[4]http://archive.ics.uci.edu/ml/

surements to determine the age of the abalone instead of enduring the time consuming task of cutting the shell through the cone, staining it, and counting the number of rings through a microscope. We used 160 source instances with 11 target instances for training and 77 for testing.

Wine Quality

The classification task is to determine the quality of white wine samples[4] by using red white samples as source set. The features are the wine's 11 physical and chemical characteristics and the output labels are given by experts' grades of 5 and 6. We used 3655 source instances and 14 target instances for training and 1306 for testing.

3.6.3 Experimental Results

The comparison of classification accuracy is presented in Table 3.2. The results show that Dynamic-TrAdaBoost significantly improved classification on real-world datasets. We performed the following tests to show significance of our results:

1. Tested the null hypothesis that transfer learning is **not** significantly better than standard AdaBoost. We applied the Friedman Test with $p < 0.01$. Only Dynamic-TrAdaBoost was able to reject the hypothesis.

2. We performed paired t-tests with $\alpha = 0.01$ to test the null hypothesis that classification performance was **not** improved over TrAdaBoost. For all datasets, Dynamic-TrAdaBoost rejected the hypothesis while "Fixed-Cost TrAdaBoost" did not.

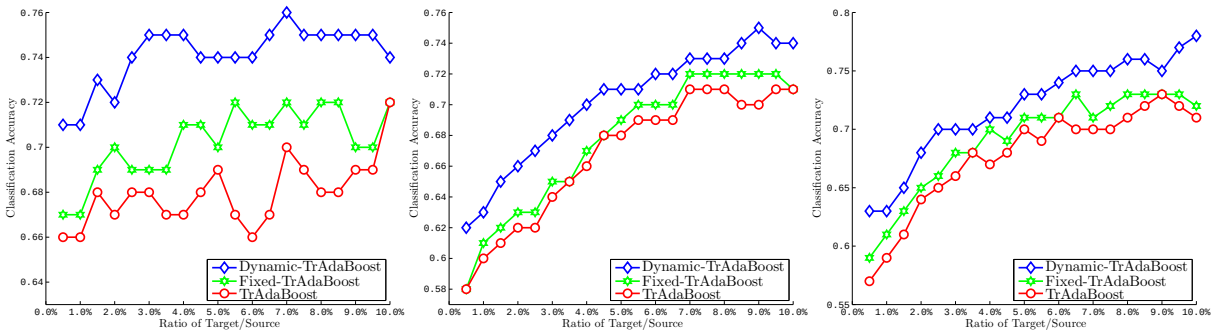
3. Paired t-tests with $\alpha = 0.01$ also rejected the null hypothesis that Dynamic-TrAdaBoost did **not** improve classification over "Fixed-Cost TrAdaBoost" for all datasets.

In Figure 3.5, the accuracy of the "20Newsgroups" dataset is plotted at different target/source ratios. The plots demonstrate that incorporating a dynamic cost into Dynamic-

Table 3.2: Classification accuracy of AdaBoost (Target), TrAdaBoost, Fixed-Cost (best result reported for TrAdaBoost with costs fixed at (1.1, 1.2, 1.3), Dynamic (Dynamic-TrAdaBoost)

Dataset	AdaBoost	TrAdaBoost	Fixed-Cost (1.1,1.2,1.3)	Dynamic
Sci vs Talk	0.552	0.577	0.581	**0.618**
Rec vs Sci	0.546	0.572	0.588	**0.631**
Rec vs Talk	0.585	0.660	0.670	**0.709**
Wine Quality	0.586	0.604	0.605	**0.638**
Abalone Age	0.649	0.689	0.682	**0.740**

TrAdaBoost improved classification at different ratios as compared to TrAdaBoost or a fixed correction cost.



(a) REC vs TALK (b) SCI vs TALK (c) REC vs SCI

Figure 3.5: Accuracy of TrAdaBoost, Best of Fixed-Cost-TrAdaBoost (1.1,1.2,1.3) and Dynamic-TrAdaBoost on the "20 Newsgroup" dataset at different target/source ratios.

3.7 Conclusion

We investigated instance transfer learning methods and analyzed their main weaknesses. We proposed an algorithm with an integrated dynamic cost to resolve a major issue in the most popular boosting-based instance transfer algorithm, TrAdaBoost. This issue causes source instances to converge before they can be used for transfer learning. We theoretically and empirically demonstrated the cause and effect of this rapid convergence and validated that the addition of our dynamic cost improved classification on several datasets.

Table 3.3 presents a summary of how the addition of a Dynamic Correction improved the performance of TrAdaBoost. The improved classification results were theoretically and empirically demonstrated and real-world experiments validated the improved classification performance.

No Correction	Dynamic Correction
❖ Source weights converge even when correctly classified	❖ Source weight convergence matches the "Weighted Majority Algorithm"
❖ Target Weights increase even when correctly classified	❖ Target weight convergence matches AdaBoost
❖ Convergence rate is correlated with choice of weak classifier	❖ Choice of weak classifier has no effect on convergence
❖ Convergence rate is bounded by ratio of source/target	❖ Ratio of source/target has no effect on convergence
❖ Convergence rate is correlated with number of boosting iterations	❖ Number of boosting iterations has no effect on convergence
❖ Convergence rate is correlated with number of source instances	❖ Number of instances has no effect on convergence

Table 3.3: The difference between TrAdaBoost and Dynamic-TrAdaBoost

CHAPTER 4

TRANSFER LEARNING FOR RARE CLASS ANALYSIS

4.1 Label-Space Optimization

Transfer learning has the potential to address the shortage of sample within "Absolute Rarity", but intuitive results require balanced optimization to address the class imbalance. Figure 4.1 demonstrates how optimizing with balanced accuracy measures improve classification with imbalanced datasets. The two classifiers in Figure 4.1 are optimized with different types of accuracy measures (Arithmetic vs. Geometric [63]/ Balanced [47]/Harmonic [89]). This example shows that given a constrained classifier

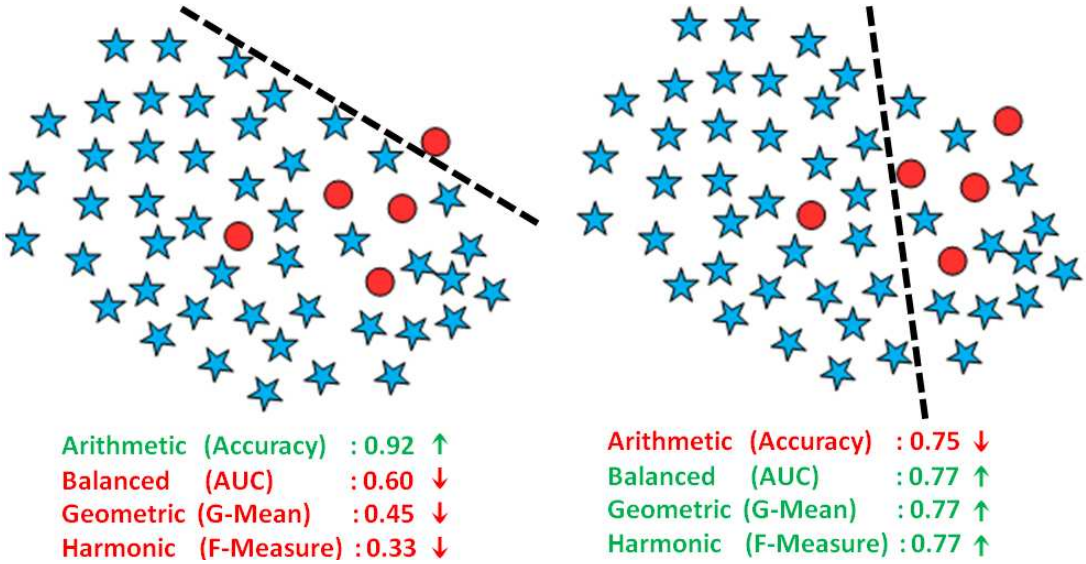


Figure 4.1: (a) Classifier minimizing Arithmetic error. (b) Classifier minimizing (Geometric, Balanced, Harmonic) error.

(linear classifier in this example), more intuitive results can be obtained with a degraded Arithmetic Accuracy and an improved Balanced (Geometric/Balanced/Harmonic) Accu-

racy. The example in Figure 4.1 also demonstrates that a standard classifier (optimized for Arithmetic Accuracy) is biased for correct classification of the majority class. This is the standard premise for "Imbalanced Learning" methods.

Proposition 2: For a non-trivial problem, a standard classifier yields lower error rate for the majority label as compared to that of the minority since it optimizes:

$$\min_{\varepsilon}(n\varepsilon) = \min_{\varepsilon^l}\left(\sum_{\forall l \in Y} n^l \varepsilon^l\right) \tag{4.1}$$

In a label-imbalanced problem where ($n^{l=\text{majority}} \gg n^{l=\text{minority}}$), a traditional classifier optimizing Equation (4.1) can achieve high accuracy if it classifies all instances as majority instances. This proposition serves as a foundation for all imbalanced learning methods [83, 50, 38].

4.2 Label-Dependent Error

In this section, we present Theorems 6, 7, and 8 to prove that minimizing a single label-dependent error is equivalent ti minimizing all the "Balanced Accuracy" measures and it will be demonstrated that this label-dependent optimization can improve the balanced statistics for "Absolute Rarity".

Theorem 6: *Maximizing the Balanced Accuracy (BAC) is equivalent to minimizing the sum of label-dependent errors independent of the number of samples within each class:*

$$\max(BAC) = \arg\min_{\varepsilon^l} \sum_{l \in Y} \varepsilon^l$$

Proof. To prove theorem 6, we will start with a binary labeled example and extend to general form. With no optimization of the prediction threshold of a binary classifier, classifier threshold at a pre-set level, the Area under the ROC Curve (AUC) is equivalent

to Balanced Accuracy (BAC) [47][1]. This balanced accuracy is the average accuracy of each class and in turn equates to the average of sensitivity and specificity. It is calculated as:

$$
\begin{aligned}
AUC = BAC &= \tfrac{1}{2}\left(Sensitivity + Specificity\right) \\
&= \tfrac{1}{2}\left[\left(\frac{TruePositive}{TruePositive+FalseNegative}\right) + \left(\frac{TrueNegative}{TrueNegative+FalsePositive}\right)\right] \\
&= \frac{1}{2}\left[\left(\frac{\sum_{i=1}^{n}\left(y_i=+1, f_i=+1\right)}{\sum_{i=1}^{n}\left(y_i=+1, f_i=\pm 1\right)}\right) + \left(\frac{\sum_{i=1}^{n}\left(y_i=-1, f_i=-1\right)}{\sum_{i=1}^{n}\left(y_i=-1, f_i=\pm 1\right)}\right)\right] \\
&= \sum_{l\in Y}\frac{0.5\left(\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)\right)}{\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)+\sum_{i=1}^{n}\left(y_i^l\neq f_i^l\right)} = \sum_{l\in Y}\frac{0.5\left(n^l\left(1-\varepsilon^l\right)\right)}{n^l\left(1-\varepsilon^l\right)+n^l\left(\varepsilon^l\right)} = 0.5\left(\sum_{l\in Y}\left(1-\varepsilon^l\right)\right)
\end{aligned}
\tag{4.2}
$$

Equation (4.2) can be maximized as:

$$
\max\left(BAC\right) = \arg\max_{\varepsilon^l}\left(0.5\left(\sum_{l\in Y}\left(1-\varepsilon^l\right)\right)\right) = \arg\min_{\varepsilon^l}\left(\sum_{l\in Y}\varepsilon^l\right) \tag{4.3}
$$

□

The optimization problem in Equation (4.3) is a constrained optimization problem bounded mainly by the classifier's VC dimension [106] and the number of samples. It is minimized as:

$$
\begin{aligned}
&\min_{\varepsilon^l}\sum_{l\in Y}\varepsilon^l \\
&\text{s.t.}\sum\nolimits_{\forall l\in Y} n^l\varepsilon^l = \varepsilon
\end{aligned}
\tag{4.4}
$$

Theorem 7: *Maximizing the Geometric Mean (G-Mean) is equivalent to minimizing the product of label-dependent errors and is independent of the number of samples within each*

[1]http://www.causality.inf.ethz.ch/challenge.php?page=evaluation

class:

$$\max\left(G-Mean\right)=\arg\min_{\varepsilon^l}\prod_{l\in Y}\varepsilon^l$$

Proof. Similar to Theorem 6, we start with a binary labeled example and extend to general form:

$$\begin{aligned}
&G-Mean=\sqrt{\left(Sensitivity\right)\left(Specificity\right)}\\
&=\sqrt{\left(\frac{TruePositive}{TruePositive+FalseNegative}\right)\left(\frac{TrueNegative}{TrueNegative+FalsePositive}\right)}\\
&=\sqrt{\left(\frac{\sum_{i=1}^{n}\left(y_i=+1,f_i=+1\right)}{\sum_{i=1}^{n}\left(y_i=+1,f_i=\pm1\right)}\right)\left(\frac{\sum_{i=1}^{n}\left(y_i=-1,f_i=-1\right)}{\sum_{i=1}^{n}\left(y_i=-1,f_i=\pm1\right)}\right)}\\
&=\sqrt{\prod_{l\in Y}\frac{\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)}{\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)+\sum_{i=1}^{n}\left(y_i^l\neq f_i^l\right)}}=\sqrt{\prod_{l\in Y}\frac{n^l\left(1-\varepsilon^l\right)}{n^l\left(1-\varepsilon^l\right)+n^l\left(\varepsilon^l\right)}}=\sqrt{\prod_{l\in Y}\left(1-\varepsilon^l\right)}
\end{aligned}\tag{4.5}$$

Maximizing the statistic in Equation (4.5):

$$\max\left(G-Mean\right)=\arg\max_{\varepsilon^l}\left(\sqrt{\prod_{l\in Y}\left(1-\varepsilon^l\right)}\right)=\arg\min_{\varepsilon^l}\left(\prod_{l\in Y}\varepsilon^l\right)\tag{4.6}$$

□

Similar to Equation (4.4), the optimization problem in Equation (4.6) is a constrained optimization problem and is minimized as:

$$\begin{aligned}
&\min_{\varepsilon^l}\prod_{l\in Y}\varepsilon^l\\
&\text{s.t.}\sum\nolimits_{\forall l\in Y}n^l\varepsilon^l=\varepsilon
\end{aligned}\tag{4.7}$$

Since both Equation (4.4) and Equation (4.7) are constrained by the classifier's error

rate, modifying the weak learner to improve classification on one label can degrade classification on the other label (Figure 4.1).

Theorem 8: *An improved G-Mean coupled with no degradation in the BAC will improve the F-Measure.*

Proof. The harmonic mean of sensitivity and specificity is a particular realization of the F-measure [89] and is maximized as:

$$
\begin{aligned}
f-measure &= \left[\frac{2(Sensitivity)(Specificity)}{Sensitivity+Specificity}\right] = \left[\frac{2\left(\frac{\sum_{i=1}^{n}(y_i=+1,f_i=+1)}{\sum_{i=1}^{n}(y_i=+1,f_i=\pm 1)}\right)\left(\frac{\sum_{i=1}^{n}(y_i=-1,f_i=-1)}{\sum_{i=1}^{n}(y_i=-1,f_i=\pm 1)}\right)}{\left(\frac{\sum_{i=1}^{n}(y_i=+1,f_i=+1)}{\sum_{i=1}^{n}(y_i=+1,f_i=\pm 1)}\right)+\left(\frac{\sum_{i=1}^{n}(y_i=-1,f_i=-1)}{\sum_{i=1}^{n}(y_i=-1,f_i=\pm 1)}\right)}\right] \\
&= \left[\frac{2\prod_{l\in Y}\frac{\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)}{\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)+\sum_{i=1}^{n}\left(y_i^l\neq f_i^l\right)}}{2\sum_{l\in Y}\frac{0.5\left(\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)\right)}{\sum_{i=1}^{n}\left(y_i^l=f_i^l\right)+\sum_{i=1}^{n}\left(y_i^l\neq f_i^l\right)}}\right] = \left[\frac{\prod_{l\in Y}\frac{n^l\left(1-\varepsilon^l\right)}{n^l\left(1-\varepsilon^l\right)+n^l\left(\varepsilon^l\right)}}{\sum_{l\in Y}\frac{0.5\left(n^l\left(1-\varepsilon^l\right)\right)}{n^l\left(1-\varepsilon^l\right)+n^l\left(\varepsilon^l\right)}}\right] \qquad (4.8) \\
&= \left[\frac{\prod_{l\in Y}\left(1-\varepsilon^l\right)}{0.5\left(\sum_{l\in Y}\left(1-\varepsilon^l\right)\right)}\right]
\end{aligned}
$$

Equation can be optimized as:

$$
\begin{aligned}
\max\left(f-measure\right) &= \arg\max_{\varepsilon^l}\left[\frac{\prod_{l\in Y}\left(1-\varepsilon^l\right)}{0.5\left(\sum_{l\in Y}\left(1-\varepsilon^l\right)\right)}\right] \\
&= \arg\max_{\varepsilon^l}\left[\frac{(G-Mean)^2}{BAC}\right] = \arg\max_{\varepsilon^l}\left[\frac{\left(\prod_{l\in Y}\varepsilon^l\right)}{\sum_{l\in Y}\varepsilon^l}\right] \qquad (4.9)
\end{aligned}
$$

□

This section demonstrated that a label-dependent error minimization, where error is

minimized for each label independently, can improve balance and improve the balanced statistics. This is equivalent to minimizing:

$$\arg\min_{\varepsilon^l} \left[\max_{l \in Y} \left(\varepsilon^l \right) \right] \tag{4.10}$$

Equation 4.10 is equivalent to stating that the best balanced results can be achieved by minimizing the error of the worst performing class. In a balanced learning problem, all labels have equal effect on BAC and G-Mean but as the label space gets more imbalanced, $\frac{n^{l=\text{majority}}}{n^{l=\text{minority}}} \to \infty$, the contribution of the minority label's error rate to the classifier's overall accuracy can thus be approximated as:

$$\sum_{\forall l \in Y} n^l \varepsilon^l \approx \sum_{l \in majority} n^l \varepsilon^l \tag{4.11}$$

Equation (4.11) demonstrates that biasing the classifier to favor the minimization of the minority label, in an imbalanced dataset, has minimal effect on the overall accuracy. With no significant change in accuracy, the balanced arithmetic mean will also not be significantly degraded since the increased error of the majority label is negated by the decreased error of the minority label. On the other hand, G-Mean is the balanced geometric mean and is significantly improved if balance is induced.

4.3 "Rare-Transfer" Algorithm

Algorithm 2 in chapter 3 improved the performance of transfer learning and in this chapter, the Weighted Majority Algorithm's (WMA) update mechanism in Equation (3.9) will be modified on line 10 of Algorithm 2 to replace C^t with the label-dependent cost (C^{l^t}) for a class balanced transfer learning. This dynamic cost is calculated on line 8 and it promotes balanced transfer learning. Starting with equal initial weights and using standard weak classifiers, that optimize for accuracy, these classifiers achieve low

error rates for the majority and high error rate for the minority as they are overwhelmed with the majority label. The label dependent cost, C^l, controls the rate of convergence of the source instances so weights converge slower[2] for labels with high initial error rates (minority classes). As minority labels get higher normalized weights with each successive boosting iteration, the weak classifiers would subsequently construct more balanced separating hyperplanes. Since only the $\frac{N}{2} \rightarrow N$ weak classifiers are used for the final output, the expectation is that the most consistent **and** balanced mix of source instances would be used for learning the final classifier. *This is the first transfer learning algorithm to optimize with label information.*

4.3.1 Overview

Figure 4.2 presents an overview of how a label-dependent transfer can improve classification with "Absolute Rarity". While transfer is preserved by incorporating samples that induce positive transfer, modifying this transfer to improve learning and simultaneously compensate for imbalance can address both the impediments that hinder learning with "Absolute Rarity". The figure gives an overview of how a label-dependent correction factor can improve balanced classification on a target dataset with separate controls of minority and majority accuracy to generate the optimal balanced accuracy solution.

4.3.2 Correction for Rare Learning

The "Correction Factor" introduced in the previous chapter allows for strict control of the source weights' rate of convergence and this property will be exploited to induce balance to a "Absolute Rarity". Balanced classifiers can be dynamically promoted by accelerating the rate of weight convergence of the majority label and slowing it for the minority label.

We will first prove that the Weighted Majority Algorithm exacerbate the difficulty of

[2]Slower or decreased convergence rate means that a weight converges to zero with higher number of boosting iterations.

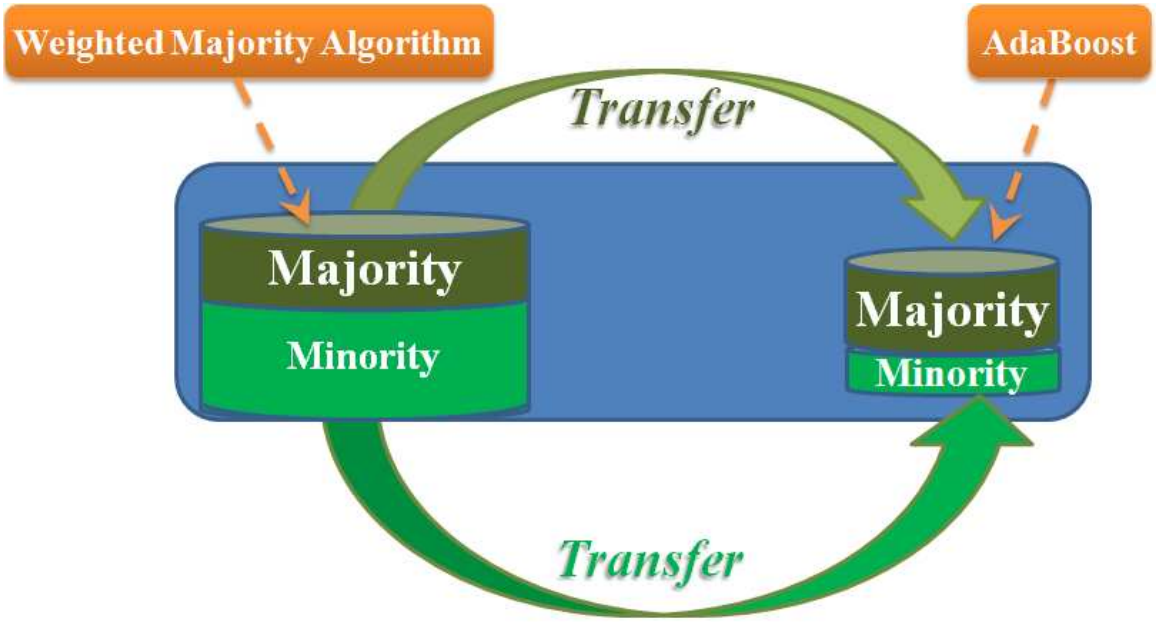


Figure 4.2: Overview of Balanced Transfer

learning when the dataset is imbalanced.

Theorem 9: *In an imbalanced problem, updating the source instances via the Weighted Majority Algorithm, WMA, significantly degrades the classification performance (especially if the final classifier is computed using only the* $\frac{N}{2} \rightarrow N$ *boosting iterations).*

Proof. A misclassified source instance at boosting iteration t is updated via the WMA update mechanism and its $t+1$ weight is adjusted to: $w_{src}^{t+1} = \beta_{src} w_{src}^{t}$. The source update mechanism is set by β_{src} which is set to:

$$0 < \left[\beta_{src} = \frac{1}{1 + \sqrt{\frac{2\ln(n)}{N}}} \right] < 1 \tag{4.12}$$

Since $\beta_{src} < 1$, a misclassified source instance's weight would converge after normalization. Since weak classifiers at initial boosting iterations, with equally initialized weights, yield high error rates for minority labels (Proposition 2), the minority label's weights would subsequently have less influence on the $t+1$ classifier and would accelerate the

rate of convergence as:

$$\begin{aligned} w_{src}^{t+1} &\geq w_{src}^{t} \quad y_{src} = \ddot{f}^{t} \\ w_{src}^{t+1} &< w_{src}^{t} \quad y_{src} \neq \ddot{f}^{t} \end{aligned} \tag{4.13}$$

Ignoring normalization, the minority label's weights decrease exponentially as:

$$\begin{aligned} w_{src}^{t+1} &\approx \beta_{src} w_{src}^{t} \\ w_{src}^{t+2} &\approx \beta_{src} w_{src}^{t+1} \approx \beta_{src}\beta_{src} w_{src}^{t} \\ &\vdots \\ w_{src}^{t+k} &\approx \beta_{src}^{k} w_{src}^{t} \end{aligned} \tag{4.14}$$

Since the final classifier in Algorithm 2 is computed from the cascade of learners constructed in iterations $\frac{N}{2} \rightarrow N$, where the minority source weights could have already converged, the final output would be extremely imbalanced as it will have added only majority weights.

□

This outcome was observed even in generally balanced instance-transfer methods. It was noted by [34] that boosting for transfer learning sometimes yielded a final classifier that always predicted one label for all instances. Dai et al. [24] re-sampled the data at each step to balance the classes since they observed similar behavior.

Conversely, updating the target instances via the AdaBoost update mechanism improves the performance on an imbalanced dataset particularly if the final classifier is computed using only the $\frac{N}{2} \rightarrow N$ boosting iterations. A misclassified target instance at boosting iteration t is updated via the AdaBoost update mechanism and its $t+1$ weight is adjusted to: $w_{tar}^{t+1} = \beta_{tar} w_{tar}^{t}$. The target update for a misclassified instance's weight is dependent on β_{tar} where:

$$1 < \left[\beta_{tar} = \frac{1 - \varepsilon_{tar}^{t}}{\varepsilon_{tar}^{t}} \right] < \infty \tag{4.15}$$

Since $\beta_{tar} > 1$, a misclassified target instance's weight would increase after normalization and the minority label's weights would in turn have more influence on the $t+1$ classifier and bias the classifier to improve learning on the minority as:

$$\begin{aligned} w_{tar}^{t+1} &< w_{tar}^{t} \quad y_{tar} = \ddot{f}^{t} \\ w_{tar}^{t+1} &\geq w_{tar}^{t} \quad y_{tar} \neq \ddot{f}^{t} \end{aligned} \tag{4.16}$$

Since the final classifier is computed from the cascade of learners constructed in iterations $\frac{N}{2} \rightarrow N$, where the minority label's instances have increased weights to compensate for the lack of its samples, the final output would be more balanced.

Optimization for "Absolute Rarity"

Using definition 1, the sum of source instances' weight is monotonically decreasing as:

$$\begin{aligned} n\mathrm{w}_{src}^{t+1} &= n\mathrm{w}_{src}^{t}\left[1 + \varepsilon_{src}^{t}\left(\beta_{src} - 1\right)\right] \\ n\mathrm{w}_{src}^{t+1} &\leq n\mathrm{w}_{src}^{t} \text{ since } \left(\beta_{src} < 1, \varepsilon_{src}^{t} \geq 0\right) \end{aligned} \tag{4.17}$$

Similarly, the target instances' weights are monotonically increasing:

$$\begin{aligned} m\mathrm{w}_{tar}^{t+1} &= m\mathrm{w}_{tar}^{t}\left[1 + \varepsilon_{tar}^{t}\left(\beta_{tar} - 1\right)\right] \\ \mathrm{mw}_{tar}^{t+1} &\leq m\mathrm{w}_{tar}^{t} \text{ since } \left(\beta_{tar} > 1, \varepsilon_{tar}^{t} \geq 0\right) \end{aligned} \tag{4.18}$$

Line 4 of "Rare Transfer" normalizes the sum of all weights and all thus all source weights are monotonically converging. On the other hand, Theorem (9) demonstrated that the minority sources' weights converge faster than the majority sources' weights. To improve balanced classification, we include a "Label-Dependent Correction Factor" to dynamically slow the convergence of the source instances' weights while simultaneously

reducing the differential in error between the minority and majority label. It is set to:

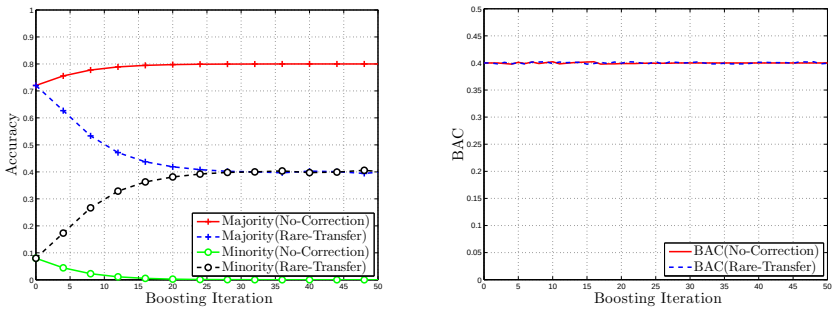
$$C^l = \left(1 - \varepsilon^l_{src}\right) \tag{4.19}$$

This factor dynamically slows convergence for the label with a higher error since the convergence rate is inversely correlated to the error. Biasing each label's weights allows "Rare Transfer" to steer for the construction of a final classifier that includes a best-fit set of auxiliary samples and has an equal error on all labels.

4.4 Empirical Analysis

This section presents an empirical validation of Theorems 6, 7, 8 and 9. A binary labeled classification problem was simulated with 900 majority instances, 100 minority instances and the weak classifier's arithmetic error rate was set to ($\varepsilon = 0.2$). Since this an imbalanced dataset and the weak classifier is weighted, error rate (ε^l) was correlated with the label's relative weight as: $\varepsilon^l = \frac{\varepsilon \sum_{i \in l} w^i}{\sum w}$.

In Figure 4.3-a, we plot the accuracy for both labels and demonstrate that applying a label dependent correction factor to the weight update mechanism induces balance while the un-corrected WMA update mechanism minimizes only the majority label's error and causes extreme imbalance. This behavior is reflected in the statistical measures as Figure 4.3-b shows that inducing balance causes no degradation in BAC while Figure 4.3-c shows that inducing balance improved G-Mean. The improved G-Mean coupled with no degradation in BAC is reflected in the improved F-Measure in Figure 4.3-d.



(a) Label Accuracy (b) BAC

(c) G-Mean (d) F-Measure

Figure 4.3: Effect of "Rare Correction"

4.5 Experimental Results of "Rare Transfer"

4.5.1 Experiment Setup

AdaBoost [43] was used as the reference algorithm. We applied SMOTE [16][3], with 5 k-nearest neighbors ($k = 5$), before boosting to compare with an imbalanced classification method (SMOTE-AdaBoost). Reference algorithms were trained with the target-only set and with the combined target/source set. Thirty boosting iterations were experimentally proven sufficient for training.

Base Learner$\left(\ddot{f}\right)$: We did not use decision stumps as weak learners since most data belongs to the source and it was not possible to keep the target error below 0.5 (as mandated by AdaBoost) for more than a few iterations. We used a strong classifier, classification trees, and applied a top-down approach where we trimmed the tree at the first node that achieved the desirable target error rate $\varepsilon_{tar}^{t} < 0.5$.

[3]Refer to section 5.3 for more details about SMOTE

Cross Validation: Since rare datasets easily over-fit and terminate boosting, we restarted all algorithms with a new cross validation fold when any algorithm terminated before reaching 30 iterations. We used random non-intersecting folds and tabulated each statistic with the macro [124] average of 30 runs using the number of samples outlined in Table 4.1. We also present plots to demonstrate two imbalance rates across a variable size of minority training sets.

4.5.2 Datasets Used

20 Newsgroups[4] is a popular transfer learning text collection that is partitioned across 20 groups with 3 cross-domain tasks and a two-level hierarchy as outlined in [23]. We used the Threshold of Document Frequency [2] to maintain around 500 features and imbalanced the set to generate a high dimensional, small and imbalanced training set. Table 4.1 presents a detailed description of the data used in the experiments. Different within-class imbalance ratio were used to examine the effect of imbalance on the different algorithm's performance. The overall size of the minority class within training set also varied and was maintained at a low sample/feature ratio to demonstrate the effectiveness of transfer learning in compensating for the lack of samples.

4.5.3 Experimental Results

BAC Results

The BAC results presented in Table 4.2 show that Rare-Transfer improved the Balanced Accuracy. The improved performance is consistent even when the addition of auxiliary data seemed to degrade the performance due to the infusion of negative transfer. This is proof that the "transfer learning" objective in our algorithm improved learning with *only* the best set of auxiliary instances. Figure 4.4 demonstrates that the improved performance is consistent across different datasets, imbalance ratios and absolute number of minority samples.

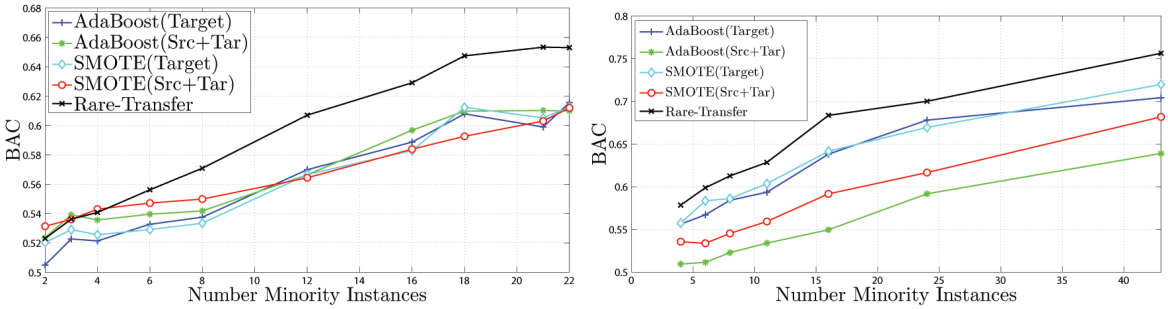
[4] http://people.csail.mit.edu/jrennie/20Newsgroups/

Table 4.1: Detailed data description. Nu:Numeric, No:Nominal.

Dataset	Features	Source Majority	Source Minority	Target Majority	Target Minority
REC vs TALK	Nu : 500 No : 0	rec autos motorcycles 1009 2%,5%,10%	talk politics.guns politics.misc 20,50,101 2%,5%,10%	rec sports.baseball sports.hockey 472,453,393 2%,5%,10%	talk politics.mideast religion.misc 10,23,39 2%,5%,10%
REC vs SCI	Nu : 500 No : 0	rec autos sports.baseball 1187 2%,5%,10%	sci sci.crypt sci.space 24,59,119 2%,5%,10%	rec motorcycles sports.hockey 486,518,543 2%,5%,10%	sci sci.electronics sci.med 10,26,55 2%,5%,10%
SCI vs TALK	Nu : 500 No : 0	sci sci.med sci.electronics 840 2%,5%,10%	talk politics.misc religion.misc 17,42,84 2%,5%,10%	sci sci.crypt sci.space 513,529,507 2%,5%,10%	talk politics.guns politics.mideast 10,26,51 2%,5%,10%

Table 4.2: Comparison of Balanced Accuracy values on real-world datasets

Dataset	AdaBoost (Target)	AdaBoost (Src+Tar)	SMOTE (Target)	SMOTE (Src+Tar)	Rare Transfer
Rec-Sci (2%)	0.564	0.571	0.566	0.569	0.594
Sci-Talk (2%)	0.544	0.541	0.544	0.540	0.571
Rec-Talk (2%)	0.569	0.534	0.577	0.547	0.610
Rec-Sci (5%)	0.635	0.622	0.635	0.645	0.664
Sci-Talk (5%)	0.602	0.591	0.607	0.596	0.632
Rec-Talk (5%)	0.635	0.569	0.642	0.602	0.672
Rec-Sci (10%)	0.696	0.680	0.699	0.706	0.706
Sci-Talk (10%)	0.662	0.639	0.672	0.647	0.679
Rec-Talk (10%)	0.714	0.628	0.722	0.673	0.736



(a) REC-VS-SCI (2%) (b) REC-VS-TALK(10%)

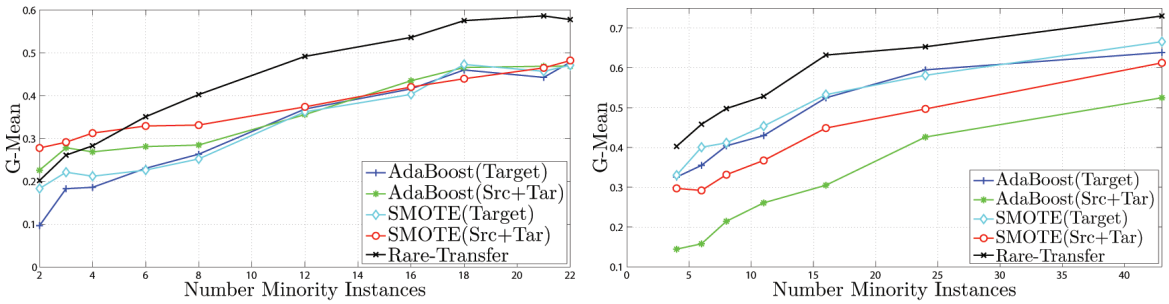
Figure 4.4: BAC at different minority samples

G-Mean Results

The results in Table 4.3 confirm that Rare-Transfer *significantly* improved the Geometric Mean. The results on the 20-News(2%) dataset demonstrate improved performance with severe label imbalance and an extremely high features/samples ratio (10 minority samples, ≈500 majority samples, 500 features). Figure 4.5 shows that Rare-Transfer consistently yield superior results even after conditions improve and the other algorithms can construct representative hypotheses.

Table 4.3: Comparison of G-Mean values on real-world datasets

	AdaBoost (Target)	AdaBoost (Src+Tar)	SMOTE (Target)	SMOTE (Src+Tar)	Rare Transfer
Rec-Sci (2%)	0.324	0.362	0.338	0.379	0.430
Sci-Talk (2%)	0.270	0.271	0.277	0.292	0.380
Rec-Talk (2%)	0.343	0.221	0.378	0.293	0.460
Rec-Sci (5%)	0.500	0.492	0.501	0.561	0.592
Sci-Talk (5%)	0.433	0.423	0.446	0.464	0.541
Rec-Talk (5%)	0.502	0.340	0.516	0.450	0.591
Rec-Sci (10%)	0.615	0.605	0.623	0.641	0.674
Sci-Talk (10%)	0.563	0.531	0.584	0.575	0.637
Rec-Talk (10%)	0.647	0.483	0.661	0.597	0.702



(a) REC-VS-SCI (2%) (b) REC-VS-TALK(10%)

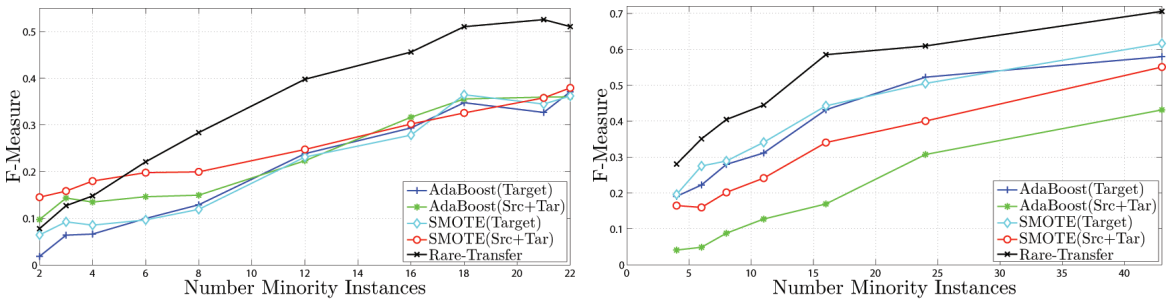
Figure 4.5: G-Mean at different minority samples

F-Measure Results

The F-Measure [89] results are presented in Table 4.4 and demonstrate that Rare-Transfer constructs a more balanced classifier. The improvements are consistent at different imbalance ratios and sample sizes as shown in Figure 4.6. The results also demonstrate the effect of "Absolute Rarity" on imbalance as the classifiers generate more balanced classification results with the addition of samples to the training set. This is similar to the results observed in previously conducted research on the effect of "Absolute Rarity" on imbalanced classification [117].

Table 4.4: Comparison of F-Measure values on real-world datasets

	AdaBoost (Target)	AdaBoost (Src+Tar)	SMOTE (Target)	SMOTE (Src+Tar)	Rare Transfer
Rec-Sci (2%)	0.208	0.240	0.218	0.258	0.331
Sci-Talk (2%)	0.225	0.115	0.256	0.174	0.363
Rec-Talk (2%)	0.149	0.144	0.152	0.164	0.275
Rec-Sci (5%)	0.405	0.393	0.408	0.490	0.534
Sci-Talk (5%)	0.407	0.228	0.424	0.349	0.527
Rec-Talk (5%)	0.324	0.309	0.343	0.365	0.473
Rec-Sci (10%)	0.550	0.541	0.560	0.638	0.645
Sci-Talk (10%)	0.590	0.389	0.609	0.536	0.671
Rec-Talk (10%)	0.484	0.445	0.514	0.513	0.600



(a) REC-VS-SCI (2%) (b) REC-VS-TALK(10%)

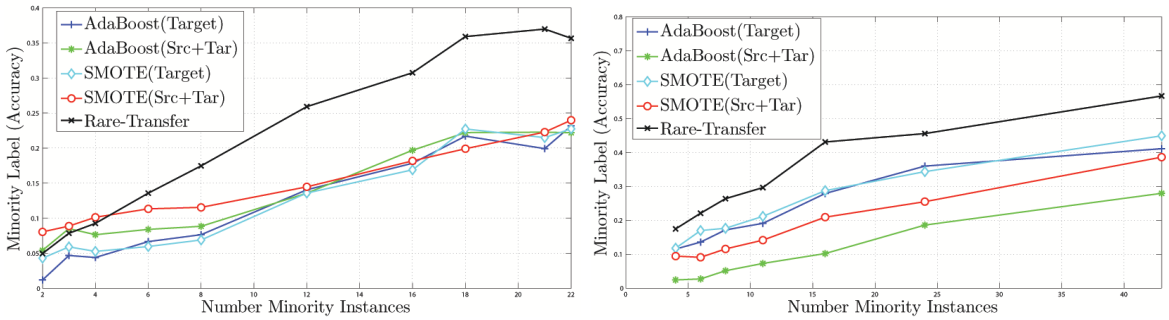
Figure 4.6: F-Measure at different minority samples

Minority Label Accuracy

The F-Measure results of the previous section only show that the harmonic mean was improved and thus we present the average classification accuracy for the minority class in Table 4.5. The improvement is consistent even when the number of samples is very small as evident in Figure 4.7. The results demonstrate that the "Absolute Rarity" of minority training examples is a challenging problem and although "Rare-Transfer" improved classification, the error rate for the minority label is still generally high.

Table 4.5: Minority Label Accuracy

	AdaBoost (Target)	AdaBoost (Src+Tar)	SMOTE (Target)	SMOTE (Src+Tar)	Rare Transfer
Rec-Sci (2%)	0.129	0.146	0.135	0.155	0.223
Sci-Talk (2%)	0.089	0.084	0.090	0.093	0.179
Rec-Talk (2%)	0.139	0.069	0.157	0.102	0.247
Rec-Sci (5%)	0.273	0.256	0.277	0.339	0.401
Sci-Talk (5%)	0.208	0.190	0.224	0.233	0.347
Rec-Talk (5%)	0.273	0.142	0.289	0.228	0.392
Rec-Sci (10%)	0.400	0.386	0.415	0.498	0.533
Sci-Talk (10%)	0.334	0.295	0.368	0.364	0.481
Rec-Talk (10%)	0.436	0.264	0.459	0.396	0.555



(a) REC-VS-SCI (2%) (b) REC-VS-TALK(10%)

Figure 4.7: Minority label's accuracy at different minority samples

4.6 Conclusion

We discussed the impediments to instance transfer learning with "Absolute Rarity" and proposed the first classification method optimized specifically for "Absolute Rarity". Our framework simultaneously compensated for the lack of data and the label-imbalance using a transfer learning paradigm with a balanced statistics objective. We theoretically analyzed and empirically verified our work and demonstrated its effectiveness with several imbalance ratios and different sizes of training datasets.

CHAPTER 5

DEMOGRAPHICS EXPERIMENTS AND EXTENSIONS

5.1 Introduction

In this chapter, we first compare our algorithms on real-world demographics problems. The results demonstrate how our work improves classification performance when addressing a real-world problem with "Absolute Rarity". In the second section of this chapter, we extend our work by modifying our "Rare-Transfer" algorithm to address imbalanced learning as an over-sampling approach where the generated samples come from an auxiliary domain instead of artificially creating new instances as is done in current imbalanced learning approaches.

5.2 Demographics Experiments

In this section, we applied our algorithm to several demographics datasets where "Absolute Rarity" presents an obstacle to learning.

5.2.1 Data Description

Heart Failure: We collected Heart Failure (HF) patient data from the Henry Ford Health System (HFHS) in Detroit. This dataset contains records for 8913 unique patients who had their first hospitalization with primary HF diagnosis. The goal is to predict if a patient will be re-admitted within 30 days after being discharged from the hospital and to apply the model to rural hospitals or to demographics with less data. This is an important healthcare problem since re-hospitalization for heart failure (HF) occurs in around one-in-five patients within 30 days of discharge. HF is disproportionately distributed across the US population with significant disparities based on gender, age, ethnicity, geographic

area, and socioeconomic status [21]. The physiology is also different; for example, renal function features are calculated as [96]:

$$\text{eGFR} = \frac{186.3\,(gender)\,(race)}{(\text{age})^{0.203}(\text{Creatinine Levels})^{1.154}}$$

Where *gender* is set to 0.742 for women and *race* is set 1.21 for African-Americans and both are set to 1 otherwise. Other non-demographic features included length of hospital stay, ICU stay and dichotomous variables for whether a patient was diagnosed with diabetes, hypertension, peripheral vascular disease, transient ischemic attack, heart failure, chronic kidney disease, coronary artery disease, hemodialysis treatment, cardiac catheterization, right heart catheterization, coronary angiography, balloon pump, mechanical ventilation or general intervention. The average results with 50 minority samples (patient was re-hospitalized) is reported.

Employment: This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey [40]. We used the dataset[1] to predict if a non-Muslim woman is employed based on her demographic and socio-economic characteristics. In the training set, only 22 of the 1275 were not Muslim and only 7 of them were employed.

Parkinson: This dataset [69] is composed of a range of biomedical voice measurements from people with early-stage Parkinson's disease[1]. The goal is to predict if a female patient's score on the Unified Parkinson's Disease Rating Scale [39] is high (UPDRS$\geq$10) or low (UPDRS$<$10). In the training set, only 125 of the 3732 participants were female and only 13 of them had a low UPDRS score.

5.2.2 Experiment Setup

We compared our algorithms, "Dynamic-TrAdaBoost" [3] and "Rare Transfer" with one imbalanced and one balanced classifier. AdaBoost [43] with target instances was used and we also applied SMOTE [16] to the target data before boosting to compare

[1] http://archive.ics.uci.edu/ml/

Table 5.1: HF:Heart Failure, Nu:Numeric, No:Nominal. AA:African American, CA:Caucasian American, NReH:Not Re-Hospitalized, ReH:Re-Hospitalized

Dataset	Features	Source Majority	Source Minority	Target Majority	Target Minority
HF (Race)	Nu : 2 No : 20	AA NReH 4468 (78.0%)	AA ReH 1026 (17.9%)	CA NReH ≈183 (3.2%)	CA ReH ≈50 (0.9%)
HF (Age)	Nu : 1 No : 21	Over 50 NReH 4513 (75.4%)	Over 50 ReH 1182 (19.8%)	Under 50 NReH ≈241 (4.0%)	Under 50 ReH ≈50 (0.8%)
HF (Gender)	Nu : 2 No : 20	Male NReH 3366 (75.7%)	Male ReH 818 (18.4%)	Female NReH ≈211 (4.8%)	Female ReH ≈50 (1.1%)
Employment (Religion)	Nu : 5 No : 3	Muslim Un-employed 955 (74%)	Muslim Employed 298 (23%)	Non-Muslim Un-employed 15 (0.02%)	Non-Muslim Employed 7 (0.006%)
Parkinson (Gender)	Nu : 19 No : 0	Male UPDRS≥10 3732 (89%)	Male UPDRS<10 276 (8%)	Female UPDRS≥10 112 (0.03%)	Female UPDRS<10 13(0.003%)

with an imbalanced method (SMOTE-AdaBoost). We followed the setup in Chapter 4.

5.2.3 Experimental Results

This section presents the balanced classification statistics including G-Mean, BAC, F-measure and the minority label's accuracy on the demographics dataset outlined in Table 5.1. Additionally, we present classification plots for the Heart Failure prediction dataset in Figures 5.1, 5.2, 5.3 and 5.4. We plot each accuracy metric using different demographics and with different number of target samples. The results demonstrate that our "Rare Transfer" algorithm yields superior results and compensates for "Absolute Rarity". The plots also illustrate that "Dynamic-TrAdaBoost" performs well once the minority samples' training size reaches a significant number and more target examples are available to compensate for the within-class imbalance without the need for balanced transfer.

G-Mean

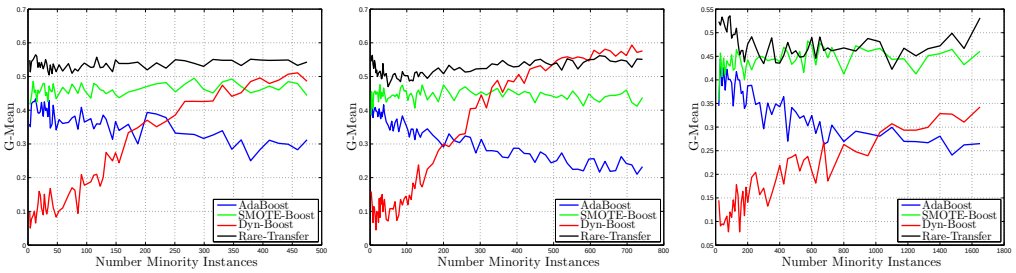
The G-Mean results are presented in Table 5.2. We performed the following significance tests:

- Tested the null hypothesis that the G-Mean performance of "Rare Transfer" is **not** significantly better than AdaBoost. We applied the Friedman Test with $p < 0.01$. SMOTE-AdaBoost and Rare-Transfer were able to reject the hypothesis for all datasets.

- We performed paired t-tests with $\alpha = 0.01$ to test the null hypothesis that G-Mean performance was **not** improved over SMOTE-AdaBoost. For all datasets, "Rare-Transfer" rejected the hypothesis.

Table 5.2: G-Mean on demographics data

	Target AdaBoost	SMOTE AdaBoost	Dynamic TrAdaBoost	Rare Transfer
Heart Failure (Race)	0.382	0.456	0.145	**0.533**
Heart Failure (Age)	0.355	0.440	0.164	**0.478**
Heart Failure (Gender)	0.374	0.444	0.117	**0.510**
Employment (Religion)	0.422	0.467	0.331	**0.488**
Parkinson (Gender)	0.518	0.715	0.841	**0.874**



(a) G-Mean (Race) (b) G-Mean (Gender) (c) G-Mean (Age)

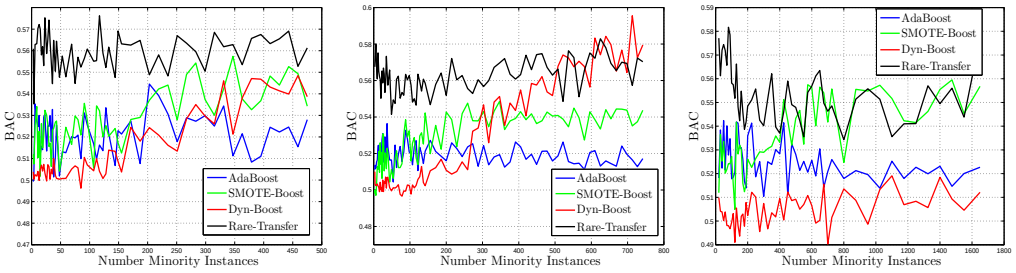
Figure 5.1: G-mean on different demographics and at different minority samples.

BAC Analysis

We present the BAC results in Table 5.3 as evidence that biasing the classifier to reduce the classification error of the minority label did not degrade the overall performance of the classifier. The results confirm the analysis in chapter 4 where the classifier improved learning on the minority label without an overall degradation in balanced accuracy. This is desired outcome for a classification algorithm that is limited to a small number of training examples combined with a within-class imbalance. This outcome also matches the theoretical and empirical analysis in chapter 4.

Table 5.3: BAC on demographics data

	Target AdaBoost	SMOTE AdaBoost	Dynamic TrAdaBoost	Rare Transfer
Heart Failure (Race)	0.519	0.521	0.504	**0.559**
Heart Failure (Age)	0.526	0.532	0.503	**0.555**
Heart Failure (Gender)	0.520	0.517	0.502	**0.560**
Employment (Religion)	0.506	0.510	**0.524**	0.513
Parkinson (Gender)	0.649	0.761	0.862	**0.885**



(a) BAC (Race) (b) BAC (Gender) (c) BAC (Age)

Figure 5.2: BAC on different demographics and at different minority samples.

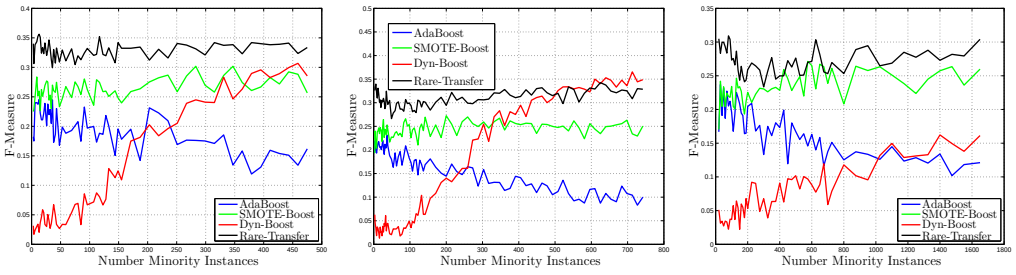
F-Measure Analysis

The F-measure results are presented in Table 5.4. We performed the following significance tests:

- Tested the null hypothesis that the F-measure performance is **not** significantly better than AdaBoost. We applied the Friedman Test with $p < 0.01$. SMOTE-AdaBoost and Rare-Transfer were able to reject the hypothesis for all datasets.
- We performed paired t-tests with $\alpha = 0.01$ to test the null hypothesis that F-measure performance was **not** improved over SMOTE-AdaBoost. For all datasets, Rare-Transfer rejected the hypothesis.

Table 5.4: F-Measure on demographics data

	Target AdaBoost	SMOTE AdaBoost	Dynamic TrAdaBoost	Rare Transfer
Heart Failure (Race)	0.205	0.257	0.055	**0.328**
Heart Failure (Age)	0.180	0.229	0.063	**0.269**
Heart Failure (Gender)	0.194	0.236	0.037	**0.301**
Employment (Religion)	0.276	0.325	0.188	**0.378**
Parkinson (Gender)	0.404	0.552	0.702	**0.749**



(a) F-Measure (Race) (b) F-Measure (Gender) (c) F-Measure (Age)

Figure 5.3: F-measure on different demographics and at different minority samples.

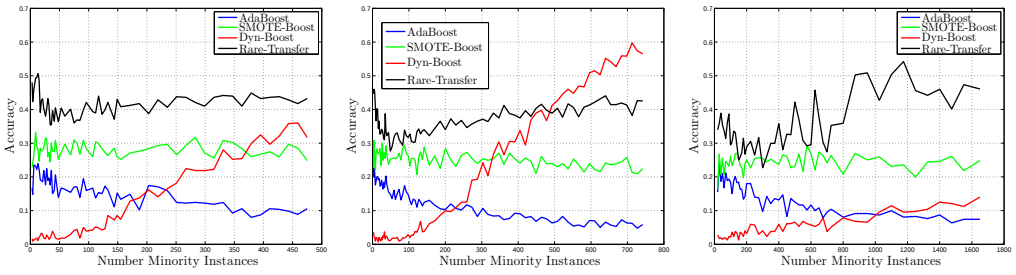
Analysis of Minority Label's Accuracy

The results for the minority label's accuracy are presented in Table 5.5. We performed the following significance tests:

- Tested the null hypothesis that the minority label's accuracy is **not** significantly higher than AdaBoost. We applied the Friedman Test with $p < 0.01$. SMOTE-AdaBoost and Rare-Transfer were able to reject the hypothesis for all datasets.
- We performed paired t-tests with $\alpha = 0.01$ to test the null hypothesis that the minority label's accuracy was **not** improved over SMOTE-AdaBoost. For all datasets, Rare-Transfer rejected the hypothesis.

Table 5.5: Minority Label's Accuracy on demographics data

	Target AdaBoost	SMOTE AdaBoost	Dynamic TrAdaBoost	Rare Transfer
Heart Failure (Race)	0.178	0.279	0.033	**0.411**
Heart Failure (Age)	0.150	0.244	0.039	**0.321**
Heart Failure (Gender)	0.167	0.262	0.021	**0.341**
Employment (Religion)	0.249	0.321	0.120	**0.471**
Parkinson (Gender)	0.306	0.550	0.748	**0.792**



(a) Accuracy (Race) (b) Accuracy (Gender) (c) Accuracy (Age)

Figure 5.4: Minority label's accuracy on different demographics and minority sample sizes.

5.3 "Auxiliary Domain Over Sampling"

In this section, we transform "Rare Transfer" to a sampling technique that samples an auxiliary (source) domain for new instances instead of synthetically generating new instances from the existing (target) domain. This algorithm is different from transfer learning methods, and more like imbalanced methods, since we make no attempt at classification. We simply exploit the concepts within transfer learning to search for instances from the auxiliary data that best fit the hypothesis of our training set under the assumption that these instances are likely to improve learning by discovering the real underlying distribution of the target domain. Once the training set in that target domain is augmented with a best-fit set of auxiliary instances, any machine learning algorithm can be applied.

Sampling for Imbalanced learning

To bias a classifier for improved minority classification, several sampling methods have been proposed to synthetically subtract or create new training instances and modify the training dataset so it has a relatively more balanced class distribution. The simplest method is to apply random under-sampling or random over-sampling. In random under-sampling methods [64], instances from the majority class are randomly removed until there is a balanced mix of majority and minority samples. While removing instances does induce balance, this is not helpful for learning when there is limited number of minority samples since the total size of the dataset would be around twice that of the minority and useful information from the majority can be lost.

There are two main set of methods for oversampling. The simplest method is "Random over-sampling" [17] which create duplicates of the minority instances to bias the classifier with an increased number of minority instances. The second type of oversampling uses an advanced sampling strategy that creates new synthetic instances based on exist-

ing instances' distribution. This intelligent oversampling strategy is superior to random over-sampling since random oversampling generally overfits [33] as the duplicated data generates a training model that is too close to the over-duplicated training set. Synthetic Minority Over-sampling Technique (SMOTE) [16] is the most popular intelligent over sampling method and it creates synthetic instances instead of merely duplicating existing ones. To create new synthetic minority instances, SMOTE randomly selects minority samples and generates new ones along the line segment connecting neighboring samples. This is performed in the feature space and each synthetic sample is the average of its k neighbors. In one of the many variation of SMOTE [56], the randomness in minority instance selection is replaced with a minority sample selection that is based on k-means [72]. The minority instances are clustered to an equal number of clusters and oversampling is applied to each cluster independently to generate a broader minority space. Another method, "Focused Resampling" [53], applies over-sampling only to minority instances that lie along the classification boundary.

A newly proposed algorithm [112] "Transfer Ensemble Model for Imbalanced Data" or TEMID used an "imbalance than transfer" concept to learn with imbalance and transfer where the two problems were cascaded. Over-sampling of the source and target was the first step and was followed by under-sampling of the merged set of source and target instances. The first two steps comprised the balancing stage and were followed by an ensemble transfer phase.

Sampling From an Auxiliary Domain

The weakness of sampling methods is that they cannot discover the true underlying distribution if the number of minority samples is sparse. For example, Figure 5.5 demonstrates how a classifier might construct the separation boundary if no sampling strategy is applied. There are two important observations: First, the learner discovers

only a subsection of the minority label's space since there is not enough data to discover the true underlying distribution. Second, the learner rejects outliers since the minority label's outlier is overwhelmed with majority label's instances.

In Figure 5.6, a SMOTE strategy oversamples the minority label by creating synthetic

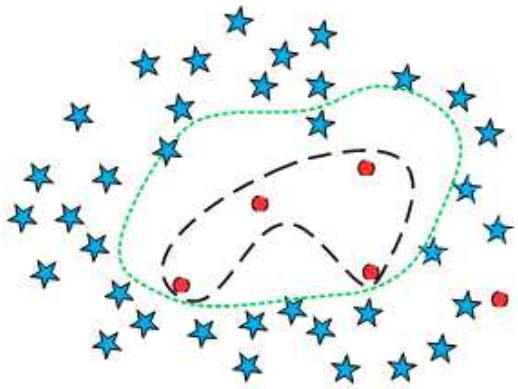


Figure 5.5: Classification without Oversampling.

minority instances. This oversampling strategy discovers a subsection of the minority label's distribution but is only limited to the subsection where minority samples already exist. While the SMOTE over-generalization improves training with the denser minority space, this methods also overgeneralizes to outliers and biases for a noise sensitive classifier.

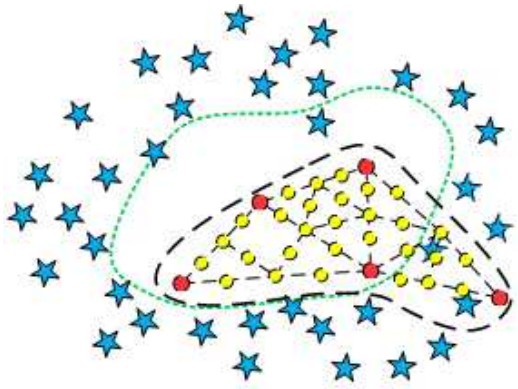


Figure 5.6: Classification with SMOTE.

To find the true minority space, a similar auxiliary dataset can be used to find the direction where the minority label's true distribution exists. The example in Figure 5.7 shows

how synthetic data can be found and not created by using the instances that are found in the auxiliary dataset. The instances from the auxiliary dataset might have a shifted distribution that is different from that of the minority samples but finding instances that fit within the sparse underlying distribution of the target dataset allows for the discovery of that underlying distribution. In Figure 5.7, the minority instances come from an auxiliary domain where the distribution is shifted to the left-upper corner. The shifted distribution still fills the sparse hyperplane where the minority instances reside. This strategy is analogous to observing a similar dataset to get a general idea as to where the underlying true distribution resides. Subsequently, this method of over-sampling from an auxiliary domain has a better chance of finding the minority's instances true distribution in a datasets with "Absolute Rarity". This method is also better fit to reject outliers as these outliers are less likely to have many instances within close proximity.

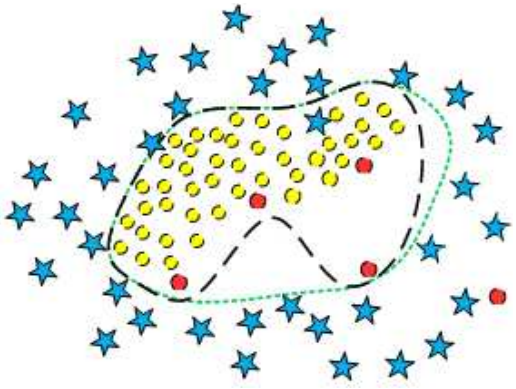


Figure 5.7: Classification from an auxiliary domain.

Figure 5.8 gives a single view of how the three classification boundaries differ. Sampling from an auxiliary domain discovered the classification boundary with the largest proportion of the underlying distribution while rejecting the outlier.

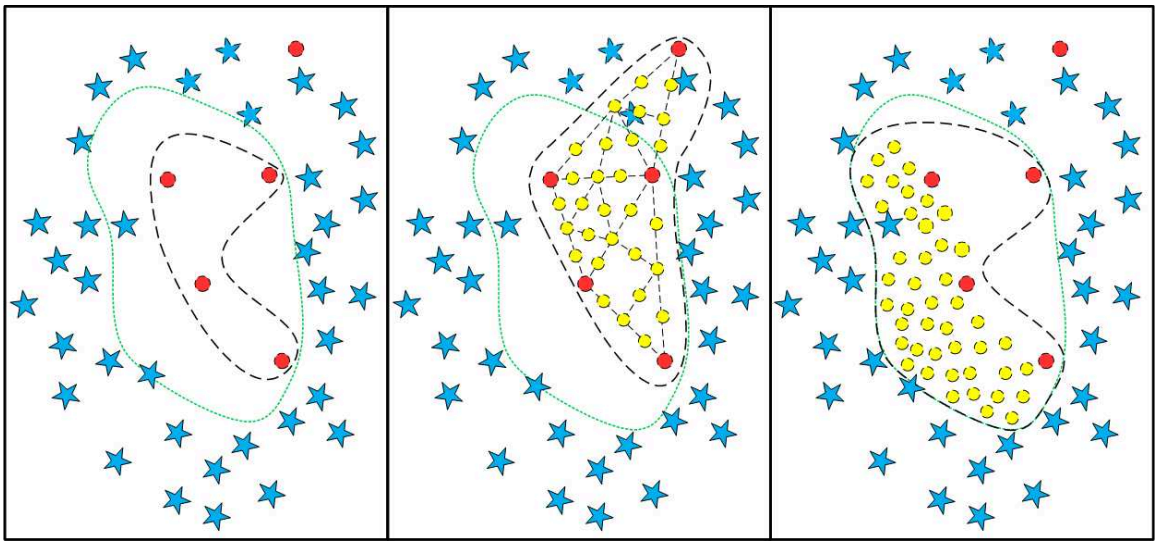


Figure 5.8: a) Left: Classification without Oversampling. b) Center: Classification with SMOTE. c) Right: Classification from an auxiliary domain.

5.4 Proposed Algorithm

5.4.1 Auxiliary Domain Over Sampling (ADOS)

The pseudo code of "Auxiliary Domain Over Sampling", ADOS, is presented in Algorithm 3. The weak classifiers on line 8 are trained with the weighted target instances to discover the hypothesis space for the target data. In AdaBoost's intended design, weighted weak classifiers are constructed to discover the hyperplane that forms the decision boundaries for classification.

In Algorithm 3, the decision boundaries are utilized to narrow the candidate set of auxiliary instances that are best fit for oversampling. We will use target instances to find the optimal separating hyperplane and subsequently use that hyperplane inversely to search for the optimal set of source instances that are aligned with that hyperplane. This concept allows for the discovery of the true underlying distribution in a dataset with "Absolute Rarity".

Sampling is performed on line 3 to generate balanced classifiers to make it easier to discover the proper separating hyperplane in an imbalanced dataset. Sampling also allows

for multiple hypotheses to be generated as set by the constant R and can reduce sensitivity to outliers. The weights of the best-fit auxiliary samples would maintain higher values after N boosting iterations, while the weights of weak performing instances (unaligned instances) would converge. After boosting is complete, the auxiliary instances with the highest weights can augment the minority target samples to improve learning. Algorithm 3 can also augment the majority instances as the overall size of the dataset might be too small and inadequate for training and adding additional instances, even for the majority class, could improve classification. Algorithm 3 can be followed by any classification algorithm for training.

5.5 Experimental Results on Real-World Datasets

5.5.1 Experiment Setup

We follow the setup in Chapter 4 with two additional datasets:

- ***Abalone***[2]***:*** The goal is to use seven physical measurements of an abalone sea snail to determine its age. This dataset is mostly from male abalone (source set, 1448 majority, 80 minority) and thus there is only a small number of female abalone samples (target set, 99 majority, 22 minority).

- ***Solar Flare:*** The goal is to predict if a C-class flare will occur within the next 24 hours using 10 attributes of solar activity measurements. Data collected in 1969 (source set, 884 majority, 182 minority) was used to augment data collected in 1978 (target set, 71 majority, 9 minority) with the latter having much more error correction and is consequently more reliable.

[2]http://archive.ics.uci.edu/ml/

Algorithm 3 Auxiliary Domain Over Sampling (ADOS)

Require:

$\triangleright$ Source Majority $D_{src-maj} = \{x_{src-maj_i}, y_{src-maj_i}\}$
$\triangleright$ Source Minority $D_{src-min} = \{x_{src-min_i}, y_{src-min_i}\}$
$\triangleright$ Target Majority $D_{tar-maj} = \{x_{tar-maj_i}, y_{tar-maj_i}\}$
$\triangleright$ Target Minority $D_{tar-min} = \{x_{tar-min_i}, y_{tar-min_i}\}$
$\triangleright$ Max iterations : N, Base learner : $\ddot{f}$

Output: Weighted auxiliary instances as they are fit for oversampling: $w = \frac{\sum_{k=1}^{R} w^k}{R}$

Procedure:

Set $\beta_{src} = \frac{1}{1+\sqrt{\frac{2\ln(n)}{N}}}$

for $k = 1$ to R **do**

Sample for Training Majority:

$$size\,(x_{train-maj}) = size\,(x_{src-min}) = m$$
$$x_{train-maj} \in x_{src-maj}$$

Initialize the training weight vector: $w_{train} = \{w_{train-maj} \cup w_{src-min}\}$

Initialize the source weight vector: $w_{src} = \{w_{src-maj} \cup w_{src-min}\}$

for $t = 1$ to N **do**

Normalize Training Weights: $w = \frac{\mathrm{w}}{\sum_{j}^{m} w_{train_j}}$

Find the candidate weak learner $\ddot{f}^t : X \to Y$ that minimizes error for D_{train} weighted according to w

Calculate the error of $\ddot{f}^t$ on D_{train}: $\varepsilon^t_{train} = \sum_{j=1}^{m} \frac{\left[w^j_{train}\right] \mathbb{1}\left[y_{train_j} \neq \ddot{f}^t_j\right]}{\sum_{i=1}^{m}\left[w^i_{train}\right]}$

Set $\beta = \frac{1-\varepsilon^t_{train}}{\varepsilon^t_{train}}$

$w^{t+1}_{src_i} = w^t_{src_i} \beta_{src}^{\mathbb{1}\left[y_{src_i} \neq \ddot{f}^t_i\right]}$ where $i \in D_{src}$

$w^{t+1}_{train_i} = w^t_{train_i} \beta^{t\,\mathbb{1}\left[y_{train_i} \neq \ddot{f}^t_i\right]}$ where $i \in D_{train}$

end for

Store Normalized Source Weights: $w^k = \frac{\mathrm{w^t_{src}}}{\sum_{i}^{n} w^t_{src_i}}$

end for

5.5.2 Experimental Results

A summary of different performance metrics is presented in Table 5.6. The results demonstrate the effectiveness of our method and the failure of SMOTE when the ratio of features/samples is high (as is the case in the 20 Newsgroups dataset that has ≈ 500 features). The "Abalone" and "Solar Flare" datasets also fit the description of "Absolute Rarity" but occupy a fairly small 7 and 10 dimensional feature spaces and reside within a limited hypothesis space that is feasible to shatter [4] with standard models.

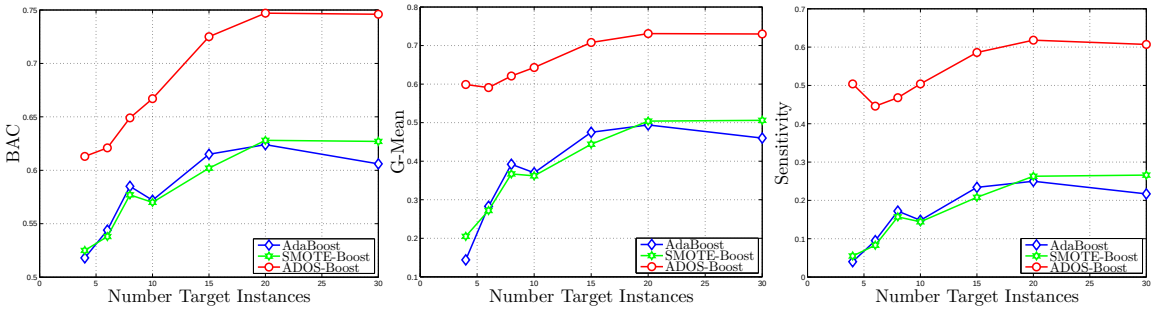
Table 5.6: Algorithm comparison with different performance metrics. (Left) AdaBoost.(Center) SMOTE-AdaBoost. (Right) ADOS-AdaBoost

Dataset	Fea	G-Mean			BAC			Minority Accuracy		
Abalone	7	0.60	0.69	**0.75**	0.67	0.71	**0.76**	0.40	0.55	**0.70**
Sun Flare	10	0.33	0.38	**0.42**	0.53	0.53	0.53	0.13	0.18	**0.22**
REC_SCI	≈500	0.34	0.34	**0.64**	0.57	0.57	**0.66**	0.14	0.14	**0.51**
REC_TALK	≈500	0.36	0.38	**0.60**	0.58	0.58	**0.64**	0.16	0.16	**0.44**
SCI_TALK	≈500	0.29	0.28	**0.63**	0.55	0.55	**0.66**	0.11	0.10	**0.50**

Comparison at different absolute and relative rates

In Figure 5.9 we vary the size of the training set for the "20-Newsgroups" dataset with a 5% imbalance ratio. The results demonstrate that SMOTE failed because the absolute number of samples is insufficient. On the other hand, ADOS compensated for the lack of minority data and for the overall lack of samples. ADOS augmented the training dataset with the best set of source instances that represent the true underlying target's distribution.

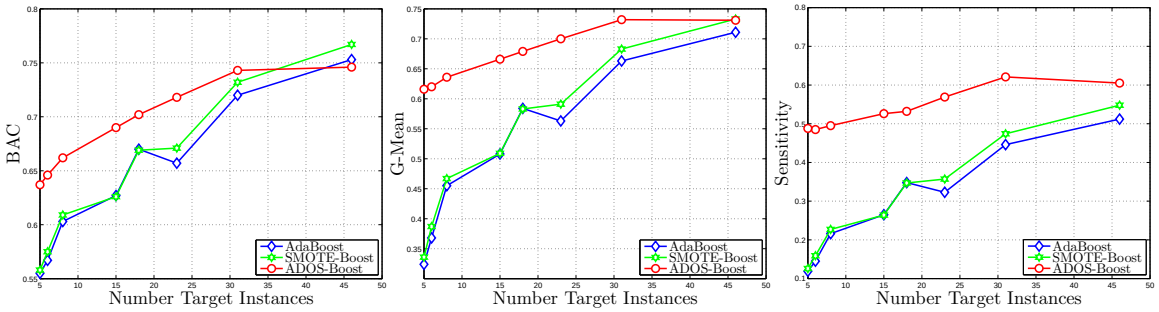
Figure 5.10 presents a comparison of our algorithm at 10% imbalance rate to validate how our algorithm performs at a rate that can include a sufficient number of training examples. The results demonstrate that competing algorithms perform well only when there is a sufficient number of samples for generalization while ADOS was able to find the true distribution faster uses the auxiliary domain. The results demonstrate that ADOS



(a) BAC (b) G-Mean (c) Minority Accuracy

Figure 5.9: REC-VS-SCI (5% minority, 3-30 samples).

can replace SMOTE particularly when the number of training examples in an imbalanced dataset is small (as in "Absolute rarity").



(a) BAC (b) G-Mean (c) Minority Accuracy

Figure 5.10: REC-VS-TALK (10% minority, 5-46 samples).

5.6 Conclusion

We tested our algorithms "Dynamic-TrAdaBoost" and "Rare-Transfer" on several real world problems and demonstrated that our algorithms improved classification for "Absolute Rarity". The results fit the theoretical and empirical analysis that we did in chapters 3 and 4. We also proposed an algorithm that extracts samples from an auxiliary domain to augment a high-dimensional, label-skewed and sample-deficient dataset. Our algorithm augmented the minority samples in an imbalanced problem with samples from an auxiliary domain and improved over existing approaches by finding the true underlying distribution of a dataset with "Absolute Rarity".

CHAPTER 6

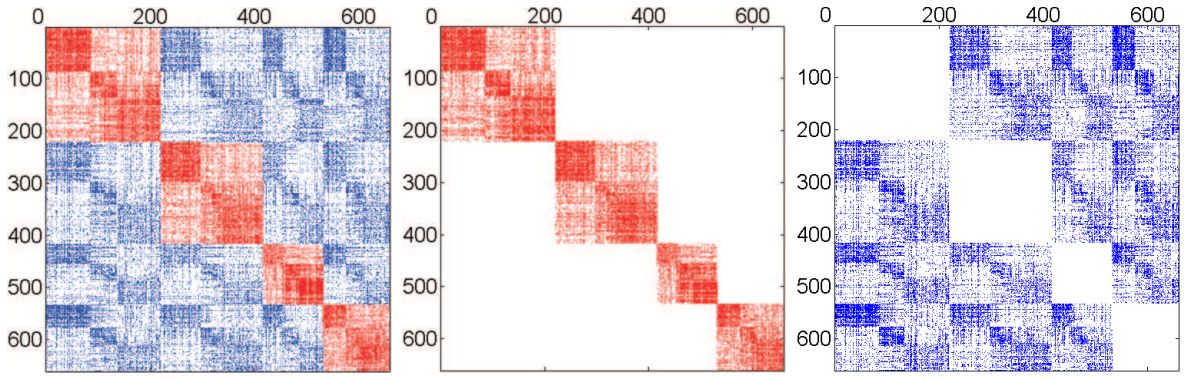
MULTI-TASK CLUSTERING USING CONSTRAINED SYMMETRIC NON-NEGATIVE MATRIX FACTORIZATION

Multi-Task clustering is a promising research direction that can leverage knowledge from related tasks to improve clustering quality. In this chapter, we present a novel formulation where we simultaneously cluster multiple tasks with an optimized Intra-Task (within-task) and Inter-Task (between-task) knowledge sharing. We present an efficient and flexible geometric affine transformation (contraction or expansion) of the distances between Inter-Task and Intra-Task instances. This transformation allows for improved Intra-Task clustering without overwhelming the individual tasks with the bias accumulated from other tasks. Inter-Task contraction compresses the distance between different tasks to obtain a global solution while Inter-Task expansion stretches the manifold and dilutes the connections between the tasks so they exhibit less influence on one-another for a single task solution. This multi-task affinity transformation requires a constrained low-rank decomposition so that simultaneous clustering can be performed while maintaining the class distribution within each individual task. We impose an Intra-Task soft orthogonality constraint to a Symmetric Non-Negative Matrix Factorization (NMF) problem to generate basis vectors that are near orthogonal within each task. Inducing orthogonal basis vectors in each task is analogous to imposing the prior knowledge that a task should have two orthogonal (different) clusters. We validate that our transformation is efficient, flexible and generates superior clustering results with several real-world experiments.

6.1 Introduction

Researchers aim to improve the performance of clustering algorithms by improving the quality of clustering or algorithm run-time [1]. One prominent method to improve clustering quality is to simultaneously cluster a set of related datasets (tasks) in what is called *multi-task clustering*. The aim of multi-task clustering is to improve clustering in each individual task by sharing knowledge between the different datasets (Inter-Task knowledge sharing). Clustering quality can be improved with some bias from the global task (combination of all tasks) with the trade-off that the Inter-Task knowledge can overwhelm each individual task and alter its distribution.

An "Affinity Matrix" is a positive symmetric similarity matrix that describes the distance (weight) between a set of instances. Figure 6.1 gives a simple example to demonstrate the decomposition of an affinity matrix, with multiple tasks, into Intra-Task and Inter-Task components. We plot the affinity matrix for a four-task clustering problem where each task comes from a different university and contains two classes of websites (personal vs. project). Figure 6.1(a) shows the full affinity matrix where all of the connections are treated equally and all tasks are combined into a single affinity matrix. The matrix in Figure 6.1(b) presents Intra-Task components while Figure 6.1(c) presents the Inter-Task components.



(a) Full Task (b) Intra-Task (c) Inter-Task

Figure 6.1: Decomposition of a Four-Task Multi-Task Affinity Matrix.

Combining all tasks into one single clustering problem generally yields inferior results since multiple tasks, with different distributions, bias and distort each individual task's distribution. Multi-Task clustering methods aim to control the effect of the Inter-Task knowledge via regularization of the clustering objective or by finding a mapping (or view) where the multiple distributions share a common distribution in the mapped space. The multi-task clustering method in [129] used Bregman divergence [13] for task regularization to require the learned local mixture densities for all tasks to be similar. In [85, 46], different tasks are mapped to a shared distribution in a Reproducing Kernel Hilbert Space (RKHS) [10] where standard clustering can be performed in the common RKHS. Information theoretic clustering methods minimize the difference in mutual information between the original data matrix and that of the clustered random variables [29]. Self-Taught clustering [25] used the information theoretic approach for unsupervised transfer learning while the loss in mutual information was added for Inter-Task regularization for multi-task co-clustering in [121].

Existing multi-task methods do not directly control the effect of the Inter-Task knowledge bias on individual tasks. Combining multiple tasks can overwhelm the Intra-Task distribution. For example, in the four-task problem, each task will be influenced by three other tasks and if all connections are treated equally, the Intra-Task clustering will be distorted (and overwhelmed) by the compounded effect of the three other tasks.

In this chapter, we control the Intra-Task vs. Inter-Task contribution to the affinity matrix for an optimal clustering outcome. Controlling the bias induced by the Inter-Task knowledge can improve the clustering quality without overwhelming the individual tasks. To incorporate controlled bias into multi-task clustering, an Affinity Matrix is transformed to a Multi-Task Affinity Matrix where the weight, w, between two instances i and k can be biased (compressed or stretched) for an optimal clustering solution. Con-

trolling the bias is performed using general multi-task coefficients (λ):

$$w_{ik} = \begin{cases} \lambda_{\text{intra}}\left(w_{ik}\right) & \text{if } \langle \text{Intra} - \text{Task} \rangle \\ \lambda_{\text{inter}}\left(w_{ik}\right) & \text{if } \langle \text{Inter} - \text{Task} \rangle \end{cases} \tag{6.1}$$

where $\lambda \in \{0, \ldots, 1\}$ is a multi-task coefficient (or maybe a matrix of coefficients for exact tailoring) that can be modified for different clustering solutions.

Diminishing the Inter-Task connections can diminish the bias induced by other tasks but as the tasks become loosely connected, standard clustering methods will cut the weakly connected tasks into different clusters. For example, in the four-task example in Figure 6.1(b), standard clustering might group the first two tasks as one cluster and the last two tasks as another (or some other combination where each task belongs to only one class). To prevent this phenomenon, we force a solution where a low-rank decomposition contains near orthogonal basis vectors within each task. Forcing orthogonal basis vectors within each task is analogous to forcing the basis vectors to contain two different clusters within each task and thus the cut is not between tasks but rather within individual tasks.

The rest of the chapter is organized as follows: Section 6.2 proposes a flexible and efficient construction method for a "Multi-Task Affinity Matrix". Section 6.3 presents an algorithm for generating relevant multi-task clustering solutions. Section 6.4 demonstrates our experimental results on several real-world datasets while section 6.5 concludes this chapter.

6.2 Multi-Task Affinity Matrix

6.2.1 Multi-Task Transformation

Figure 6.2 illustrates how a multi-task affinity transformation translates to different clustering solutions. For clarity, we present the simplest variations of λ_{intra} and λ_{inter} as we set ($\lambda_{\text{intra}} = 1$) and ($\lambda_{\text{inter}} = \lambda$). The goal is to cluster documents into either sports

or science documents where each individual task has documents that belong to a branch of science (Chemistry, Biology) or sports (Basketball, Football). Intra-Task connections can link documents via task-dependent (NBA, Avogadro) features and task-independent (Score, Celsius) features. On the other hand, Inter-Task instances can only connect via task independent features. Following the multi-task definition in Equation (6.1), different λ values generate different clustering solutions as:

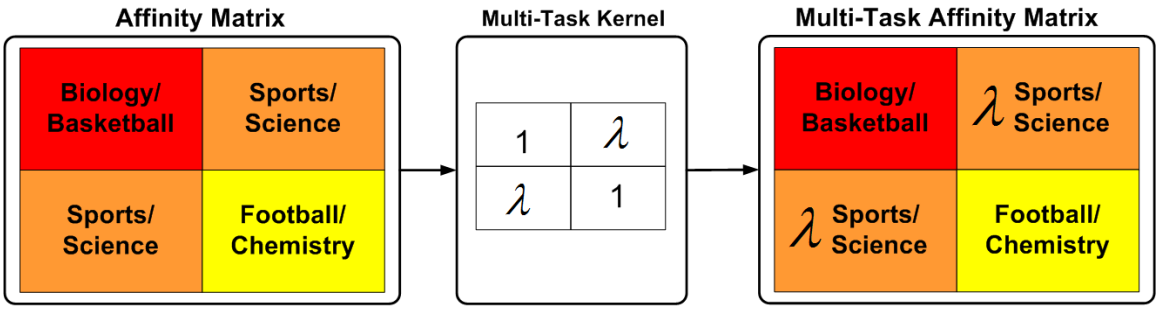


Figure 6.2: Multi-Task Affinity Transformation.

- Intra-Task Clustering ($\lambda = 0$): This coefficient removes all Inter-Task connections and thus a task cannot share or get any knowledge or bias from the other task. Because there is no connection between the two tasks, clustering is independent and so is the solution.

- Global-Task Clustering ($\lambda = 1$): All weights are biased equally as the multi-task coefficients are equal and the two tasks will combine into one global clustering solution.

- Multi-Task Clustering ($0 < \lambda < 1$): This is the general definition of multi-task clustering where the clustering is an Intra-Task solution with an Inter-Task bias (or knowledge sharing).

6.2.2 Multi-Task Graph

In section 6.2.1, we proposed some different λ variations that can produce different feature sets and different clustering solutions. For an efficient and flexible transformation, we create a star structured network that is constructed with tasks $t = \{1, \ldots, T\}$. Table 6.1 presents a summary of notations.

Table 6.1: Summary of notations

Notation	Description
D	Input Data Matrix: $D^{n \times n^z}$
I	Instance Tasks: $I = \{X_t\}_{t=1}^{T}$
Z	Feature Task (Zero$^{\text{th}}$ Task): $Z = \{X_t\}_{t=0}$
M	Multi-Task Graph: $M = \{I \cup Z\}$
T	Total Number of Tasks
t	Task Index: $t = \{1, \ldots, T\}$
n^t	Number of Instances in the t^{th} Instance Task
n^f	Number of Features in the Feature Task
f^i	i^{th} feature
e_{AB}	Binary Relation between any two nodes A and B: $e \in \{0, 1\}$
w_{AB}	Weighted Relation between any two nodes A and B: $w \in \{\mathbb{R}^+\}$
E	Binary Relation Set
W	Weighted Multi-Task Affinity Matrix
x_t^i	Node i in Task t (Instance or Feature)
N	Total Number of Instances
l	Number of Labels
$V^{N \times 2}$	NMF Basis Vectors
$H^{T \times N}$	Task-Indicator Function
$K^{2 \times T}$	Class-Indicator Function

An Input Data Matrix $\left(D^{n \times n^z}\right)$ is split as:

1. Instances (samples) form nodes in an "Instances Task": $I^{n \times 1} = \{X_t\}_{t=1}^{T}$ where each task (t) has n^t samples for a total of $n = \sum_{t=1}^{T} n^t$ nodes.

2. Features form nodes in the "Feature Task" or the Zero$^{\text{th}}$ task: $Z^{1 \times n^z} = \{X_t\}_{t=0}$ for a total of n^z feature nodes.

With all instances and features mapped as nodes, an information graph $G = \langle M, E, W \rangle$ is constructed where the union of the "Instance Tasks" and the "Feature Task" construct

the Multi-Task Graph, M:

$$M = \{I \cup Z\} = \{X_t\}_{t=0}^{T} \tag{6.2}$$

The binary relation between all nodes within this network is:

$$e \in E \in \{0, 1\} \tag{6.3}$$

This network is weighted with the non-negative weights mapping feature nodes to instance nodes with $w \in \Re^{+}$ such that:

$$\forall e = \left\langle x_t^j, x_t^i \right\rangle, \left\{ x_t^j \in X_{t=0} \wedge x_t^i \in X_{t\neq 0} \right\} \tag{6.4}$$

Equation (6.4) states that instance nodes only connect to feature nodes to form a bipartite graph. This graph is considered a bipartite graph since instance nodes and feature nodes can be divided into two disjoint sets ($t = 0$ and $t \neq 0$) such that every edge connects a vertex in ($t = 0$) to one in ($t \neq 0$). This bipartite graph will be transformed into a Weighted Multi-Task Affinity Matrix (W).

6.2.3 Sub-Graph Matrices

For an efficient and flexible mapping of the bipartite graph for the multi-task setting, we will construct two types of sub-graphs:

1. Intra-Task sub-graphs: Each sub-graph is a weighted graph connecting the Intra-Task instance nodes. For T tasks, a total of T sub-graphs are constructed.

2. Inter-Task sub-graphs: Each sub-graph is a weighted graph connecting instance nodes from two different tasks using the feature nodes that are common to both tasks. A total of $T_{C_2} = \frac{T(T-1)}{2}$ sub-graphs are constructed.

A sub-graph is defined as G_{tt^*}, where t is the index of the first task and t^* is the index of the second task. $t \neq t^*$ for an Inter-Task graph while $t = t^*$ for an Intra-Task sub-graph.

Let us define $e_{x_t^i z^j}$ as the binary relation between the j^{th} feature node (z^j) and the i^{th} instance node (x^i) of task t. Let $e_{x_t^i z^j}$ indicate if the j^{th} feature node (z^j) is connected to the i^{th} instance node (x^i) of task t as:

$$e_{x_t^i z^j} \equiv \left(z^j \in x_t^i\right) = \begin{cases} 1 & w_{x_t^i z^j} \neq 0 \\ 0 & otherwise \end{cases} \tag{6.5}$$

To check if a feature node (z^j) belongs to any instance node (x_t) in task t, it has to connect to at least one of the task's instance nodes and thus we define the binary indicator $f_{z^j}^t$ to designate if the j^{th} feature node z^j belongs to task t as follows:

$$f_{z^j}^t \equiv \left(z^j \in t\right) = \begin{cases} 1 & \sum_{i=1}^{n^t} e_{x_t^i z^j} \neq 0 \\ 0 & otherwise \end{cases} \tag{6.6}$$

Now that the preliminaries have been defined, a sub-graph $G_{tt^*}\left[x_t^i x_{t^*}^k\right]$ is constructed as:

$$\sum_{j=1}^{n^z} \left(s_{x_t^i x_{t^*}^k}\right) \left(f_{z^j}^t f_{z^j}^{t^*}\right) \left(e_{x_t^i z^j} e_{x_{t^*}^k z^j}\right) \left(w_{x_t^i z^j} + w_{x_{t^*}^k z^j}\right) \tag{6.7}$$

Equation (6.7) is divided into 4 components where the first 3 components are Kronecker's Delta "checks" to generate weighted connections between instance nodes:

1. $\left(s_{x_t^i x_{t^*}^k}\right)$ is optional to prevent self-loops and is defined as:

$$\left[s_{x_t^i x_{t^*}^k}\right] = \begin{cases} 1 & x_t^i \neq x_{t^*}^k \\ 0 & otherwise \end{cases} \tag{6.8}$$

2. $\left(f_{z^j}^t f_{z^j}^{t^*}\right)$: This section is an Inter-Task check and by definition of $f_{z^j}^t$, we will get a value of one if a feature node belongs to both tasks. It is a redundant check and is included for efficiency as it eliminates all feature nodes that are not shared by

the two tasks. For example: if two tasks only share 10% of the feature nodes, this check can eliminate 90% of the feature nodes for a reduced sub-graph. It is defined as:

$$\left[f^t_{z^j} f^{t^*}_{z^j}\right] = \begin{cases} 1 & (z^j \in t) \wedge (z^j \in t^*) \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

3. $(e_{x_t z_i} e_{x_{t^*} z_i})$: This section checks if a feature node (z_i) belongs to both instance nodes (x_{t^*}) and (x_t). The outcome is a value of One if that feature node belongs to both instance nodes as:

$$\left[e_{x^i_t z^j} e_{x^k_{t^*} z^j}\right] = \begin{cases} 1 & \left(z^j \in x^i_t\right) \wedge (z^j \in x^i_{t^*}) \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

4. $\left(w_{x^i_t z^j} + w_{x^k_{t^*} z^j}\right)$: If a feature node belongs to both tasks and is connected to instance nodes (x_t and x_{t^*}), the weight of the path connecting these two instance nodes through the feature node (z_i) should be combined in the sub-graph.

6.2.4 Multi-Task Weighted Affinity Matrix

For T tasks, a total of $T_{C_2} + T$ weighted sub-graphs are constructed and combine to construct a single Multi-Task Weighted Affinity Matrix with the weights defined as:

$$w_{x^i_t x^k_{t^*}} = \left[(\delta_{tt^*})\, \lambda_{\text{intra}} + (\delta'_{tt^*})\, \lambda_{\text{inter}}\right] G_{tt^*} \left[x^i_t x^k_{t^*}\right] \tag{6.11}$$

where (δ'_{tt^*}) is the inverse of the Kronecker's delta (δ_{tt^*}) which is defined as:

$$\delta_{tt^*} = \delta\left[tt^*\right] = \begin{cases} 1 & t = t^* \\ 0 & t \neq t^* \end{cases} \tag{6.12}$$

Equation (6.11) can be broken down to:

$$w_{x_t^i x_{t^*}^k} = \begin{cases} \lambda_{\text{intra}} G_{tt^*} \left[x_t^i x_{t^*}^k \right] & t = t^* \\ \\ \lambda_{\text{inter}} G_{tt^*} \left[x_t^i x_{t^*}^k \right] & t \neq t^* \end{cases} \tag{6.13}$$

This is the original definition of multi-task clustering in Equation (6.1).

The sub-graph construction method is:

- **Efficient:** Several steps are added to the sub-graph construction method to scale to large graphs. For example: It leverages the symmetry of the affinity matrix, sub-graphs are constructed independently (parallel construction), sub-graphs can be constructed and re-used, and no multiplications are required.

- **Flexible:** Different multi-task coefficients (or a matrix of coefficients) can be incorporated for different multi-task formulations. A star-structured network is also very flexible to accommodate a diverse set of problems such as heterogeneous networks.

6.3 Symmetric Multi-Task NMF

A Multi-Task Affinity Matrix can stretch the distance between tasks weakening the connection between them. This is beneficial as Inter-Task knowledge should only contribute complementary (auxiliary) knowledge and not overwhelm the Intra-Task knowledge. The drawback of diminishing the connection between tasks is that standard clustering does not distinguish between different tasks and different clusters and thus will assign different tasks to different clusters (labels). To prevent this, a Non-negative Matrix Factorization (NMF) [68] method is proposed where orthogonality in each task's basis vectors is promoted. Enforcing orthogonal basis vectors, within each task, is equivalent to forcing a solution with two different clusters within each task and thus the clustering "cut" is within each task and not between different tasks.

6.3.1 Non-negative Matrix Factorization

Non-negative Matrix Factorization(NMF) [68] is a matrix factorization technique that focuses on the analysis of data matrices whose elements are non-negative. The non-negativity is a useful constraint for matrix factorization since it allows for a parts representation of the data where the basis vectors are distributed and also form sparse combinations that can generate expressiveness in the reconstructions[75].

Given a non-negative data matrix X, non-negative matrix factorization is a linear, non-negative approximate data representation that aims to find two non-negative matrices U and V whose product can approximate the original matrix: $X \approx UV^T$. Various objective functions have been proposed [94] and the most widely used is the sum of squared error, Euclidean distance, function:

$$\min_{U,V \geqslant 0} \left\| X - UV^T \right\|^2 \tag{6.14}$$

Symmetric NMF is a special case of NMF decomposition where the basis U is replaced with V and the NMF optimization approximates a symmetric matrix W as: $W \approx VV^T$. Symmetric NMF can improve over standard NMF as it can discover clusters with a nonlinear underlying structure [61]. Symmetric NMF is also useful for clustering as it can be constrained to morph into several popular clustering methods [111]. For example, for a square symmetric affinity matrix, W, Symmetric NMF can be equivalent to kernel k-means clustering with the additional constraints of symmetry as follows:

$$\arg\min_{V \geq 0} \left\| W - VV^T \right\|^2, s.t. \left(V^T V = I \right) \tag{6.15}$$

NMF can also be transformed to Normalized-Cut spectral clustering by normalizing the adjacent matrix W in Equation (6.15) as:

$$\tilde{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, D = diag\left(d_1, \ldots, d_m\right), d_i = \sum_j w_j \tag{6.16}$$

6.3.2 Symmetric Multi-Task Non-Negative Matrix Factorization

In this section, we modify the Symmetric NMF objective function for multi-task clustering. Formally, given a Symmetric Multi-Task Affinity Matrix W, we want to find the basis vectors V such that:

$$\underset{V\geq 0}{\arg\min}\left[J\left(V\right)\right]=\underset{V\geq 0}{\arg\min}\left[\frac{1}{2}\left\|W-VV^{T}\right\|^{2}+\alpha Tr\left(\phi\phi^{T}\right)\right] \tag{6.17}$$

The constraint ϕ is a multi-task sparsity/orthogonality constraint and is defined as:

$$\phi=HVK \tag{6.18}$$

where $H^{T\times N}\in\{0,1\}$ is the task-indicator function. Within the trace penalty constraint, this matrix limits the orthogonality constraint to Intra-Task basis while excluding Inter-Task basis. It is defined as:

$$H\left(t,i\right)=\begin{cases}1 & i\in t\\ 0 & i\notin t\end{cases} \tag{6.19}$$

$K^{2\times T}\in\{-1,+1\}$ is the class-indicator function and sums the basis (if normalized) to zero when an Intra-Task solution is orthogonal. For a binary problem, it is defined as:

$$K\left(i,t\right)=\begin{cases}+1 & i=1\\ -1 & i=2\end{cases} \tag{6.20}$$

The first part of Equation (6.17) is a global Symmetric NMF clustering solution minimizing the reconstruction error where the generated basis vectors are near-orthogonal [111]. The second part of equation (6.17) constrains the objective function to include Intra-

Task orthogonality thus optimizing for a solution where each task has orthogonal basis vectors and thus instances that belong to different classes. The multi-task penalty added to Equation (6.17) is a soft sparsity or orthogonality constraint and is equivalent to:

$$\sum_{t \in Tasks} \Big(\sum_{i \in task_t} V_{i,1} - \sum_{i \in task_t} V_{i,2} \Big)^2 \tag{6.21}$$

Equation (6.21) can be minimized as a:

1. Sparsity Constraint: The basis vectors are sparse (have small values). This is the trivial solution and it directly increases the reconstruction error.

2. Orthogonality Constraint: The basis vectors are orthogonal within each task. This has less effect on the reconstruction error since a task with two clusters naturally has two different distributions (one for each cluster) and thus the basis vectors can be near orthogonal (after normalization) without drastic increase in re-construction error. For example, for a a simple multi-task problem with 2 independent clusters within each task, the penalty $Tr\left(\phi\phi^T\right)$ can be minimized to zero without affecting the re-construction error. Symmetric NMF's near-orthogonal basis is an essential property in equation (6.21) as orthogonal basis do not need to be normalized.

This penalty thus encourages Intra-Task orthogonality which is analogous to enforcing the prior knowledge that each task should contain two clusters. The penalty $Tr\left(\phi\phi^T\right)$ equals zero for a fully orthogonal within-task solution and is strictly increasing otherwise.

6.3.3 Multiplicative Update Rule

To derive the updating rule for Equation (6.17) with non-negative constraints on v_{ij}, we introduce the Lagrangian multipliers λ to minimize the Lagrangian function: $L = J + \sum_{ij} \lambda_{ij} V_{ij}$.

The first order KKT condition for local minima is:

$$\frac{\partial L}{\partial V_{ij}} = 0 \text{ and } \lambda_{ij} V_{ij} = 0, \forall i, j \tag{6.22}$$

Expanding the Lagrangian function L:

$$\begin{aligned} L &= \tfrac{1}{2}\left\|W - VV^T\right\|^2 + \alpha Tr\left(\phi\phi^T\right) + Tr\left(\lambda V^T\right) \\ &= Tr\left(\tfrac{1}{2}\left(W^T W - 2WVV^T + VV^T VV^T\right)\right) \\ &\quad + Tr\left(\alpha HVKK^T V^T H^T + \lambda V^T\right) \end{aligned} \tag{6.23}$$

The gradient of Equation (6.23) is:

$$\frac{\partial L}{\partial V} = -2WV + 2VV^T V + 2\alpha H^T HVKK^T + \lambda \tag{6.24}$$

The KKT complementarity condition for the non-negativity of V_{ik} gives:

$$\left(-2WV + 2VV^T V + 2\alpha H^T HVKK^T\right)_{ik} V_{ik} = 0 \tag{6.25}$$

This is the fixed point relation that the local minima for V must satisfy.

To minimize Equation(6.17), we use the gradient descent method:

$$V_{ij} \leftarrow V_{ij} - \varepsilon_{ij} \frac{\partial J}{\partial V_{ij}} \tag{6.26}$$

Setting $\varepsilon_{ij} = \frac{V_{ij}}{4VV^T V}$, we derive the proposed updating rules of Equation (6.27).

$$V_{ij} = \frac{1}{2}\left[V_{ij}\left(1 + \frac{\left(WV - \alpha H^T HVKK^T\right)_{ij}}{\left(VV^T V\right)_{ij}}\right)\right] \tag{6.27}$$

At $\alpha = 0$, this update mechanism is the same as the standard update mechanism for Symmetric Non-Negative Matrix Factorization. A value of α has to be set such that

non-negativity is enforced $\forall_{ij}$:

$$\left[WV - \alpha\left(H^T HVKK^T\right)\right]_{ij} \geq 0 \tag{6.28}$$

Since $KK^T \in \{\mathbb{R}_{<0}, \mathbb{R}_{>0}\}$, Equation (6.28) can be decomposed into its negative and positive components as:

$$\left[WV - \alpha\big(H^T HVKK^T\big)^+ - \alpha\big(H^T HVKK^T\big)^-\right]_{ij} \geq 0 \tag{6.29}$$

The term $\alpha\big(H^T HVKK^T\big)^-$ can be dropped from Equation (6.29) since $\forall_{ij}$:

$$\left[-\alpha\big(H^T HVKK^T\big)^-\right]_{ij} \geq 0 \tag{6.30}$$

Thus α has to be set to any value such that $\forall_{ij}$:

$$\alpha \leq \left(\frac{WV}{(H^T HVKK^T)^+}\right)_{ij} \tag{6.31}$$

Simply stated, non-negativity is preserved if α is set to any positive value less than the minimum of the matrix calculated in Equation (6.31).

$$\alpha \in \left\{0, \ldots, \min\left(\frac{WV}{(H^T HVKK^T)^+}\right)\right\} \tag{6.32}$$

In our implementation, we preserved non-negativity and minimized $Tr\left(\phi\phi^T\right)$ by setting α to:

$$\alpha = \min\left(\frac{WV}{(H^T HVKK^T)^+}\right) \tag{6.33}$$

6.3.4 Symmetric Multi-Task NMF Clustering Algorithm

In this section, we present our Multi-Task clustering algorithm "Symmetric Multi-Task NMF". The first two steps generate a Multi-Task Affinity Matrix where the Inter-Task connection have their weights reduced by the Multi-Task coefficients λ[1]. We iterate to get the basis vectors and set the class membership to the basis vector with highest value.

Algorithm 4 Symmetric Multi-Task NMF (SMT-NMF)

Require: Input Data Matrix $\left(D^{n\times n^z}\right)$. Multi-Task Coefficients λ

1: Construct Sub-Graph Matrices using equation (6.7).
2: Construct Weighted Multi-Task Affinity Matrix W using equation (6.11).
3: Set H using equation (6.19) and K using equation (6.20).
4: Initialize V with random non-negative values
5: **repeat**
6: $\quad \alpha = \min\left(\frac{WV}{(H^THVKK^T)^+}\right)$
7: $\quad V_{ij} = \frac{1}{2}\left[V_{ij}\left(1 + \frac{\left(WV - \alpha H^THVKK^T\right)_{ij}}{\left(VV^TV\right)_{ij}}\right)\right]$
8: **until** Convergence

Ensure: Assign clusters to: $\max(V)$.

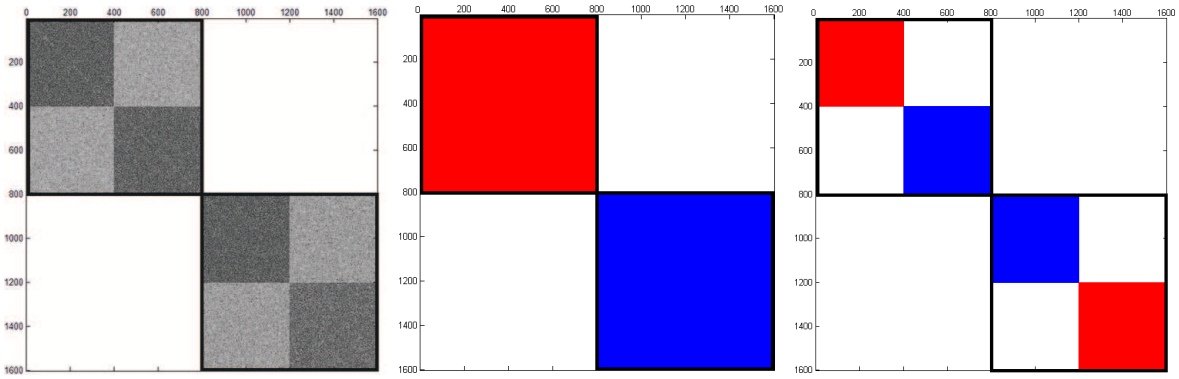
6.3.5 Synthetic Example and Relationship to Orthogonality

A synthetic 2-task example was generated with 1,600 samples where each task includes 400 samples for each label. The multi-task affinity matrix is sorted by label (for clarity) and plotted in Figure 6.3(a) with no Inter-Task connections ($\lambda_{\text{inter}} = 0$). The clustering result using standard Symmetric NMF ($\alpha = 0$) is plotted in Figure 6.3(b) and it demonstrates that standard Symmetric NMF assigns all instances in each task to one class. This is the expected behavior since the two tasks form two disjoint clusters. In Figure 6.3(c), Symmetric Multi-Task NMF formulation forces a clustering solution where instances within each task are clustered into two different classes and this is accomplished by reducing the trace orthogonality penalty, $Tr\left(\phi\phi^T\right)$, by 20 times as:

[1]For simplicity we set: $\lambda_{\text{intra}} = 1$.

$$\frac{\underset{\alpha\neq 0}{Tr}\left(\phi\phi^T\right)}{\underset{\alpha=0}{Tr}\left(\phi\phi^T\right)} = 0.05$$

The outcome of the standard Symmetric NMF formulation matches the expected outcome



(a) Two-Task Affinity (b) Standard NMF:$\alpha = 0$ (c) Multi-Task NMF:$\alpha \neq 0$

Figure 6.3: NMF Results with different λ values. (a) Affinity Matrix. (b) Clustering with Symmetric NMF. (c) Clustering with Symmetric Multi-Task NMF

in [111] and generates an orthogonal solution for the global affinity matrix as:

$$\left(VV^T\right) : \text{All} = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix}, \text{Task1} = \begin{bmatrix} 2.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix}, \text{Task2} = \begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 2.00 \end{bmatrix}$$

This solution is orthogonal for the global affinity matrix but is not "within-task orthogonal". On the other hand, our Symmetric Multi-Task NMF formulation generates a near orthogonal solution but orthogonality is task-dependent and the basis vectors are orthogonal in each individual task as:

$$\left(VV^T\right) : \text{All} = \begin{bmatrix} 0.99 & 0.16 \\ 0.16 & 1.00 \end{bmatrix}, \text{Task1} = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix}, \text{Task2} = \begin{bmatrix} 1.00 & 0.34 \\ 0.34 & 1.00 \end{bmatrix}$$

6.4 Experimental Results on Real-World Datasets

6.4.1 Dataset Description

The detailed constitution of the datasets is summarized in Table 6.2.

- 20 Newsgroups: The 20 Newsgroups[2] dataset [67] is a collection of newsgroup documents. We generated six multi-task learning problems where each task is drawn from different sub-categories as outlined in [23]. For example, if the classes are from the two top categories: "Rec vs. Talk", the first task is from sub-categories \rec.sport.hockey and \talk.religions.misc whereas the second task is from \rec.sport.baseball and \talk.politics.mideast and so on.
- Reuters-21758: The Reuters-21758[3] corpus contains Reuters news articles from 1987. Three multi-task problems with 2 tasks per problem were generated where the subcategory splits are analogous to the 20newsgroup dataset [23].
- WebKB4: The WebKB4[4] dataset contains web pages from four universities (Cornell, Texas, Washington, Wisconsin) and thus 4 tasks were generated. Web-sites belong to either Personal (student/faculty) or Project (course/project).

Table 6.2: Description of the datasets.

Dataset	Tasks	#Tasks	#Samples	#Features
20 Newsgroups	Rec vs Talk	4	40,80,120,160	636-1963
	Rec vs Sci	4	40,80,120,160	448-1876
	Rec vs Comp	4	40,80,120,160	405-1468
	Talk vs Sci	4	40,80,120,160	631-2388
	Talk vs Comp	4	40,80,120,160	504-2066
	Sci vs Comp	4	40,80,120,160	634-1939
WebKB4	Project vs Personal	4	80,160,240,320	141-760
Reuters	Orgs vs People	2	40,80,120,160	1514-2552
	Orgs vs Places	2	40,80,120,160	1501-2583
	People vs Places	2	40,80,120,160	1281-2610

[2]http://people.csail.mit.edu/jrennie/20Newsgroups/
[3]http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
[4]http://archive.ics.uci.edu/ml/

6.4.2 Experiment Setup

We compare the proposed Symmetric Multi-Task NMF (SMT-NMF) clustering algorithm with single-task and combined-task clustering methods including K-means, Normalized Cut (N-Cut) and standard Symmetric NMF ($\alpha = 0$). Additionally, we compare with the recently proposed multi-task clustering algorithm "LNKMTC" [85]. For N-Cut, we search for the best distance kernel and for LNKMTC we follow the setup in [85] where the neighborhood size for the LNKMTC's lambda was uniform for all labels, the k-NN graph is set to $k = 10$, the regularization parameter C is set by searching the grid $\{0.1, 1, 10, 100, 500, 1000\}$ and b is set to 30. For SMT-NMF, we set $\lambda_{\text{inter}} = 1$ and search the multi-task coefficient $\lambda_{\text{intra}} = \{0, \ldots, 1\}$.

Since single task algorithms performed poorly with a small number of samples, we varied the number of samples and compiled the results at four different sample sizes. Instances were randomly selected as outlined in Table 6.2. As the number of samples increased, so did the number of features (processed into TF-IDF [2] representations). At each sample size, we calculated the average clustering accuracy [122] of 30 runs and tabulated the total average from 120 runs.

6.4.3 Multi-Task Learning between Similar Tasks

The first set of experiments (4-tasks, 2-classes) and (2-tasks, 2-classes) tested the ability of SMT-NMF to improve learning when knowledge was mostly beneficial. We generated six 20newsgroups (4-tasks, 2-classes), one WebKB4 (4-tasks, 2-classes) and three Reuters (2-tasks, 2-classes) experiments. The comparison of clustering accuracy [122] is presented in Tables (6.3-6.5). **SMT-NMF consistently outperformed all other algorithms**. For the (4-tasks, 2-classes) datasets, the second best algorithms were (Single-Task NMF, 80% of experiments) and (All-Task NMF, 20% of experiments). For the (2-tasks,2-classes) Reuters dataset, the second best algorithms were evenly split between Multi-Task NMF, Single-Task NMF, and Single-Task N-cut. "LNKMTC" [85] did not

perform well as the k-NN graph construction method creates sparse affinity matrices where a disjoint (or very weakly connected) set is formed when a set of instances in any one task only connect to instances within their own cluster and do not connect with the remainder of the graph to output a solution where only one task has a good clustering solution.

Table 6.3: Performance comparison with similar tasks (WebKB4, Reuters).

Four-Tasks,Two-Classes(WebKB4). Two-Tasks,Two-Classes(Reuters)										
	WebKB4					Reuters					
DataSet	Project vs Personal					ppl-orgs		plcs-orgs		ppl-plcs	
Method	T1	T2	T3	T24		T1	T2	T1	T12	T1	T2
S-Ncut	77.3	72.5	78.1	65.8		70.9	74.0	74.3	69.1	66.7	69.7
S-NMF	84.6	83.0	84.9	86.6		72.3	65.3	62.1	71.4	69.8	61.1
S-Kmeans	76.2	72.1	80.0	60.5		55.6	57.0	59.2	58.8	57.0	57.5
A-Ncut	69.8	67.9	67.2	70.4		71.6	73.9	72.9	66.5	61.7	65.6
A-NMF	85.9	82.9	81.3	86.5		73.1	75.4	74.0	68.8	63.4	69.3
A-Kmeans	67.8	67.1	67.1	67.0		57.6	57.3	54.6	58.1	59.5	56.5
LNKMTC	61.7	52.9	52.0	63.3		59.4	62.8	62.8	59.1	56.7	60.0
SMT-NMF	**92.1**	**88.3**	**88.0**	**92.6**		**81.0**	**84.6**	**81.3**	**77.6**	**73.7**	**75.1**

Table 6.4: Performance comparison with similar tasks (20Newsgroups(1-3)).

Four-Tasks, Two-Classes. (20 Newsgroups)												
DataSet	Rec vs Talk				Rec vs Sci				Rec vs Comp			
Method	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4
S-Ncut	82.5	91.9	81.0	85.8	81.4	89.1	86.0	82.2	86.0	92.4	79.4	83.8
S-NMF	84.9	93.8	86.1	89.0	86.6	91.6	91.4	82.9	90.9	94.5	86.1	89.3
S-Kmeans	76.4	88.5	79.6	78.9	80.1	86.8	86.1	74.3	83.2	90.1	79.7	83.4
A-Ncut	82.4	86.0	61.3	64.0	81.0	81.3	60.5	60.7	84.8	86.5	72.4	75.1
A-NMF	83.7	91.9	74.9	76.1	84.3	85.2	75.6	70.2	90.7	94.9	80.7	83.3
A-Kmeans	81.8	89.1	64.3	64.7	81.5	87.7	55.4	54.3	87.2	94.0	71.3	71.0
LNKMTC	53.1	80.8	52.0	55.3	60.0	63.7	56.9	56.1	58.8	77.3	51.4	52.7
SMT-NMF	**93.2**	**96.7**	**89.6**	**91.2**	**91.2**	**94.9**	**91.8**	**88.7**	**95.7**	**97.5**	**92.0**	**92.8**

Table 6.5: Performance comparison with similar tasks (20Newsgroups(4-6)).

Four-Tasks, Two-Classes. (20 Newsgroups)												
DataSet	Talk vs Sci				Talk vs Comp				Sci vs Comp			
Method	T1	T2	T23	T4	T1	T2	T3	T4	T1	T2	T3	T4
S-Ncut	75.7	84.5	83.9	77.2	84.3	81.8	94.0	89.3	70.1	84.4	84.9	82.8
S-NMF	80.6	88.4	87.5	81.0	85.4	90.1	92.2	92.0	75.8	88.4	89.6	82.0
S-Kmeans	71.6	80.4	78.2	70.6	79.3	81.7	88.6	85.9	68.7	80.8	82.8	73.4
A-Ncut	71.9	68.6	72.3	66.1	86.3	87.2	91.4	89.2	66.1	80.2	79.0	72.4
A-NMF	76.3	76.7	80.7	71.7	91.6	92.5	95.9	94.1	65.9	82.7	84.7	78.7
A-Kmeans	70.3	73.1	78.2	66.9	91.5	91.2	96.0	93.6	65.0	79.5	81.1	74.7
LNKMTC	54.5	54.1	68.1	56.8	51.8	54.5	80.1	55.0	52.4	62.2	60.5	61.8
SMT-NMF	**89.3**	**88.5**	**91.9**	**84.7**	**97.0**	**97.2**	**97.9**	**97.4**	**78.2**	**91.8**	**93.6**	**91.0**

6.4.4 Multi-Task Learning with Similar and Different Tasks

To test the performance of SMT-NMF when there was an overwhelming bias from other tasks, we generated six 20newsgroups experiments where each experiment had four tasks with two classes per task and a random permutation of four classes (4-tasks, 4-classes). For each class, one Inter-Task distribution was helpful while the remaining five distributions were different. The results in Tables (6.6, 6.7) demonstrate that SMT-NMF had the best performance while K-means was the second best algorithm in around 2/3 of the experiments and standard NMF was the second best algorithm in around 1/3 of the experiments.

Table 6.6: Comparison with similar/different tasks (20Newsgroups(1-3)).

	Talk-Sci-Rec-Comp				Talk-Sci-Rec-Comp				Talk-Sci-Rec-Comp			
Method	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4
A-Ncut	70.6	63.7	57.6	58.9	77.9	62.2	65.6	60.0	68.7	60.1	72.3	60.6
A-NMF	62.3	60.1	66.2	69.6	66.7	70.5	65.0	67.0	63.1	63.8	65.5	69.6
A-Kmeans	71.0	64.2	65.6	65.5	**84.4**	74.8	68.5	55.6	70.5	59.0	81.8	64.4
LNKMTC	57.2	53.5	58.4	57.4	64.5	55.9	55.7	52.6	54.2	53.9	67.1	54.1
SMT-NMF	**76.7**	**72.2**	**80.4**	**81.7**	82.3	**82.9**	**76.0**	**79.9**	**75.6**	**79.2**	**82.3**	**80.5**

Table 6.7: Comparison with similar/different tasks (20Newsgroups(4-6)).

	Talk-Sci-Rec-Comp				Talk-Sci-Rec-Comp				Talk-Sci-Rec-Comp			
Method	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4
A-Ncut	66.4	61.6	65.1	61.5	60.7	59.6	80.3	66.2	63.8	74.6	60.8	70.8
A-NMF	64.0	70.0	65.2	65.5	68.4	61.8	71.0	63.2	62.1	65.7	66.5	62.6
A-Kmeans	62.6	76.4	71.4	64.7	70.0	62.4	74.7	76.0	66.7	75.8	64.1	72.6
LNKMTC	55.1	65.7	55.8	53.3	56.8	50.9	72.8	51.6	52.3	73.0	51.4	54.2
SMT-NMF	**79.9**	**89.6**	**85.5**	**79.6**	**87.1**	**76.3**	**92.2**	**85.7**	**78.7**	**83.0**	**83.2**	**76.6**

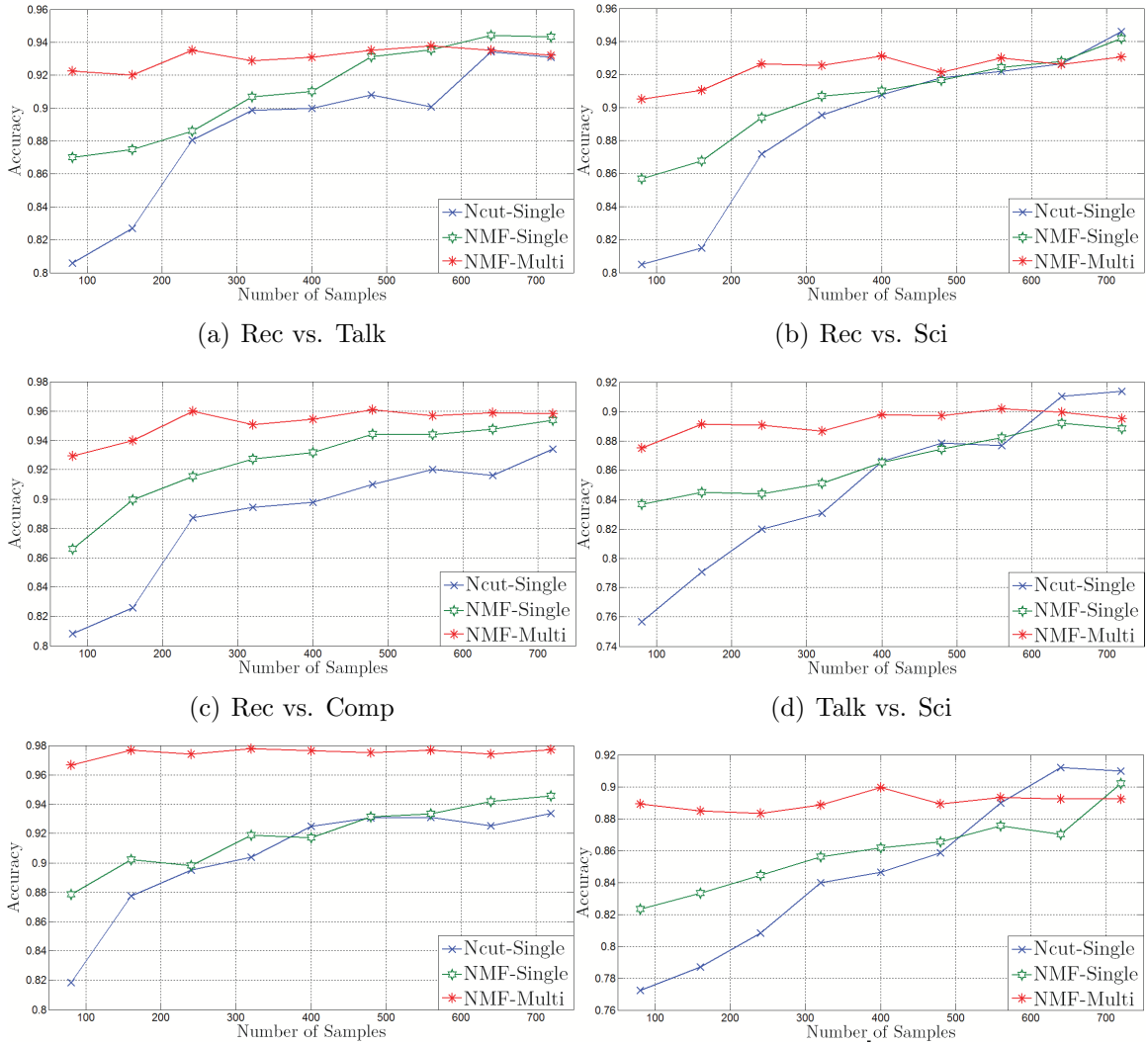
6.4.5 Clustering with Different Number of Samples

In Figure 6.4 we demonstrate the clustering performance with a variable number of instances. We did not plot the WebKB4 and Reuters datasets because SMT-NMF performed significantly better with all sample sizes. For the 20newsgroups dataset, the performance of Single-Task algorithms improved with increased availability of data. For clarity we only compare the two most competitive algorithms (Single-Task NMF and Single-Task N-Cut).

The sub-figures demonstrate that SMT-NMF effectively and consistently improved the clustering accuracy. The results also demonstrate that increasing the number of examples improved single-task clustering performance since the increased sample size expands the feature set (with more non-zero features), improves generalization (with more samples) and diminishes sensitivity to outliers.

6.5 Conclusion

A multi-task clustering framework is proposed where distances within and between tasks can be stretched or compressed to increase or diminish the knowledge-sharing between tasks. The formulation is efficient, flexible and extends to a variety of multi-task problems. A Symmetric Multi-Task Non-Negative Matrix Factorization method is presented where the NMF basis vectors are orthogonal within each task thus producing a clustering solution where knowledge-sharing does not overwhelm or bias individual



(a) Rec vs. Talk

(b) Rec vs. Sci

(c) Rec vs. Comp

(d) Talk vs. Sci

(e) Talk vs. Comp

(f) Sci vs. Comp

Figure 6.4: Performance on "20newsgroups" with varying number of training instances.

tasks. The effectiveness of the framework was demonstrated and it was illustrated that it can address several multi-task clustering problems. The superiority of the multi-task formulation was verified with an extensive of real-world multi-task clustering problems.

CHAPTER 7

FUTURE WORK AND CONCLUSION

In this dissertation, we presented several set of algorithms that can serve as a foundation for works in applied machine learning research. All the presented algorithms were validated with several type of learning problems to demonstrate that they can be extended to multiple learning domains. Transfer and Multi-Task learning methods are gaining popularity as these methods present effective solutions to many world problems and the extension of these methods to extremely small and label-skewed datasets was not an option and our hope is that this work presents methods that can be easily coded and readily available for addressing such datasets. Improvements, even minor, from methods optimized specifically for "Absolute Rarity" can have significant financial and social impact within domains, such as healthcare, where only human expertise are currently applicable. For example, rare diseases are a substantial public health burden as $6-8\%$ of people have a rare disease at some point and currently no global registry or classification codes exist [41]. Rare methods can improve learning and also encourage data collection and warehousing. Future methods can leverage information in distributed environments with multiple source sets for greater impact.

We addressed several classification ideas and our work for integrating Transfer Learning with Imbalanced Learning was the first to simultaneously combine these domains for "Absolute Rarity". Traditional imbalanced modifications including SMOTEBoost [19], over or under sampling [9] followed by transfer [112] or cost sensitive learning [98] are a straight-forward extension and can further improve classification. We made no modifications to demonstrate, without ambiguity, that our algorithmd improved balanced learning strictly using an auxiliary domain. The affinity transformation we presented is flexible where the bipartite graph can extend to heterogeneous networks while parallel

construction of sub-graphs equates the processing time of a multi-task affinity transformation to that of the largest single task to extend to large networks.

Several variation of λ_{intra} and λ_{inter} can generate different transformations. For example, λ in SMT-NMF was set to $(\lambda_{\text{intra}} = 1, \lambda_{\text{inter}} < 1)$ where Inter-Task distance was stretched to give more relevance to Intra-Task weights and generate a Multi-Task solution. Setting $(\lambda_{\text{intra}} < 1, \lambda_{\text{inter}} = 1)$ would compress Inter-Task distance for consensus clustering.

The class-indicator function K in Equation (6.20) is set to $\{-1, +1\}$ to generate an orthogonal solution when the clustering is balanced. This is beneficial for the multi-task setting but can be also applied for the single or multi task setting with imbalanced clustering. For example, the constraint ϕ in Equation (6.18) can be set to encourage a clustering solution with prior probabilities as:

$$K = \left\{ \frac{-1}{P\left(Y = -1|X\right)}, \frac{+1}{P\left(Y = +1|X\right)} \right\} \tag{7.1}$$

Since the Multi-Task constraint is minimized with an equal number of active $(V_{ij} = 1)$ basis, a **multi-class** constraint can be enforced with the concatenation of binary orthogonality constraints. For example, an $(l + 1)$-class problem changes the optimization in Equation (6.17) to:

$$\underset{V \geq 0}{\arg\min} \left[J\left(V\right)\right] = \underset{V \geq 0}{\arg\min} \left[\frac{1}{2} \left\| W - VV^T \right\|^2 + \alpha_l Tr\left(\phi_l \phi_l^T\right) \right] \tag{7.2}$$

The orthogonality constraints would be set to:

$$\phi_1 = HVK_l \tag{7.3}$$

For example, a 3-class problem would minimize the penalty in Equation (7.2) with a class-indicator function set to $K_1 = \{-1, +1, 0\}, K_2 = \{-1, 0, +1\}$.

Learning from multiple sources of data is a promising research direction as researchers leverage ever more diverse sources of information. When data is not readily available and knowledge has to be transferred from other sources, new methods (both supervised and un-supervised) have to be developed to selectively share and transfer knowledge. As machine learning methods extend to more complex and diverse set of problems, situations arise where the complexity and availability of data presents a situation where the information source is not "adequate" to generate a representative hypothesis.

In this dissertation, we presented both supervised and un-supervised techniques to tackle a problem where learning algorithms can not generalize and require an extension to leverage knowledge from different sources of data. Knowledge transfer is a difficult problem as diverse sources of data can overwhelm each individual dataset's distribution and a careful set of transformations has to be applied to increase the relevant knowledge at the risk of biasing a dataset's distribution and inducing negative transfer that can degrade a learner's performance.

We gave an overview of the issues encountered when the learning dataset does not have a sufficient supply of training examples. We categorized the structure of small datasets and highlighted the need for further research. We presented an instance-transfer supervised classification algorithm to improve classification performance in a target dataset via knowledge transfer from an auxiliary dataset. The improved classification performance of our algorithm was demonstrated with several real-world experiments. We extended the instance-transfer paradigm to supervised classification with "Absolute Rarity", where a dataset has an insufficient supply of training examples and a skewed class distribution. We demonstrated a solution with a transfer learning approach and another with an imbalanced learning approach and demonstrated the effectiveness of our algorithms with real world text and demographics classification problems. We also presented an unsupervised multi-task clustering algorithm where several datasets were simultaneously clustered and

knowledge was transferred between the datasets to improve clustering performance on each individual dataset and we demonstrated the improved clustering performance with an extensive set of experiments.

BIBLIOGRAPHY

[1] Aggarwal, C. C., and Reddy, C. K., Eds. *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC Press, 2013.

[2] Aizawa, A. An information-theoretic perspective of tf–idf measures. *Information Processing & Management 39*, 1 (2003), 45–65.

[3] Al-Stouhi, S., and Reddy, C. K. Adaptive boosting for transfer learning using dynamic updates. In *ECML/PKDD (1)* (2011), pp. 60–75.

[4] Alpaydin, E. *Introduction to machine learning*. MIT press, 2004.

[5] Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine Learning 73*, 3 (2008), 243–272.

[6] Bakker, B., and Heskes, T. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research 4* (2003), 83–99.

[7] Banko, M., and Brill, E. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (2001), Association for Computational Linguistics, pp. 26–33.

[8] Bartle, R. G., and Bartle, R. G. *The elements of integration and Lebesgue measure*. Wiley Online Library, 1995.

[9] Batista, G., Prati, R., and Monard, M. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter 6*, 1 (2004), 20–29.

[10] Berlinet, A., and Thomas-Agnan, C. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic, 2004.

[11] Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schlkopf, B., and Smola, A. J. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics 22*, 14 (2006), e49–e57.

[12] Bradley, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition 30*, 7 (1997), 1145–1159.

[13] Bregman, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics 7*, 3 (1967), 200–217.

[14] Breiman, L. Bagging predictors. *Machine learning 24*, 2 (1996), 123–140.

[15] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

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

[17] Chawla, N. V. C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *proceedings of the international conference on Machine learning* (2003), vol. 3.

[18] Chawla, N. V., Japkowicz, N., and Kotcz, A. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter 6*, 1 (2004), 1–6.

[19] Chawla, N. V., Lazarevic, A., Hall, L. O., and Bowyer, K. W. Smoteboost: improving prediction of the minority class in boosting. In *In Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003* (2003), pp. 107–119.

[20] Cieslak, D., Hoens, T., Chawla, N., and Kegelmeyer, W. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery 24* (2012), 136–158.

[21] Clancy, C., Munier, W., Crosson, K., Moy, E., Ho, K., Freeman, W., and Bonnett, D. 2010 national healthcare quality & disparities reports. Tech. rep., Agency for Healthcare Research and Quality (AHRQ), 2011.

[22] Crammer, K., Kearns, M., and Wortman, J. Learning from data of variable quality. *Advances in Neural Information Processing Systems 18* (2006), 219.

[23] Dai, W., Xue, G.-R., Yang, Q., and Yu, Y. Co-clustering based classification for out-of-domain documents. In *proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (2007), pp. 210–219.

[24] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. Boosting for transfer learning. In *proceedings of the international conference on Machine learning* (2007), pp. 193–200.

[25] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. Self-taught clustering. In *Proceedings of the 25th international conference on Machine learning* (2008), ACM, pp. 200–207.

[26] Davis, J., and Goadrich, M. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning* (2006), ACM, pp. 233–240.

[27] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 209–216.

[28] Dean, J., and Ghemawat, S. Mapreduce: simplified data processing on large clusters. *Commun. ACM 51* (January 2008), 107–113.

[29] Dhillon, I. S., Mallela, S., and Modha, D. S. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), KDD, pp. 89–98.

[30] Dietterich, T. G. Ensemble methods in machine learning. In *Multiple classifier systems*. Springer, 2000, pp. 1–15.

[31] Dietterich, T. G., and Bakiri, G. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101* (1995).

[32] Drucker, H. Improving regressors using boosting techniques. In *Proceedings of the 14th International Conferences on Machine Learning* (1997), pp. 107–115.

[33] Drummond, C., Holte, R. C., et al. C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II* (2003).

[34] Eaton, E. *Selective Knowledge Transfer for Machine Learning*. PhD thesis, University of Maryland Baltimore County, 2009.

[35] Eaton, E., and desJardins, M. Set-based boosting for instance-level transfer. In *proceedings of the 2009 IEEE International Conference on Data Mining Workshops* (2009), pp. 422 –428.

[36] Eaton, E., and desJardins, M. Selective transfer between learning tasks using task-based boosting. In *AAAI* (2011).

[37] Ertekin, S., Huang, J., Bottou, L., and Giles, L. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth*

ACM conference on Conference on information and knowledge management (2007), pp. 127–136.

[38] Ertekin, S., Huang, J., Bottou, L., and Giles, L. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (2007), pp. 127–136.

[39] Fahn, S., Elton, R., Committee, U. D., et al. Unified parkinson's disease rating scale. *Recent developments in Parkinson's disease 2* (1987), 153–163.

[40] for Child Survival, U. I. C., Nasional, I. B. K. K. B., and Institute, E.-W. P. *Secondary Analysis the the 1987 National Indonesia Contraceptive Prevalence Survey*. National Family Planning Coordinating Board, 1991.

[41] Forrest, C., Bartek, R., Rubinstein, Y., and Groft, S. The case for a global rare-diseases registry. *The Lancet 377*, 9771 (2011), 1057–1059.

[42] Frank, A., and Asuncion, A. Uci machine learning repository, 2010.

[43] Freund, Y., and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In *proceedings of the Second European Conference on Computational Learning Theory* (1995), pp. 23–37.

[44] Friedman, J. H. On bias, variance, 0/1loss, and the curse-of-dimensionality. *Data mining and knowledge discovery 1*, 1 (1997), 55–77.

[45] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. A kernel method for the two-sample problem. *CoRR abs/0805.2368* (2008).

[46] Gu, Q., and Zhou, J. Learning the shared subspace for multi-task clustering and transductive transfer classification. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining* (2009), ICDM '09, pp. 159–168.

[47] Guyon, I., Aliferis, C. F., Cooper, G. F., Elisseeff, A., Pellet, J.-P., Spirtes, P., and Statnikov, A. R. Design and analysis of the causation and prediction challenge. *Journal of Machine Learning Research - Proceedings Track 3* (2008), 1–33.

[48] Halliday, J. Email spam level bounces back after record low. *guardian.co.uk* (January 2011).

[49] Han, H., Wang, W.-Y., and Mao, B.-H. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing* (2005), vol. 3644, pp. 878–887.

[50] He, H., and Garcia, E. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on 21*, 9 (2009), 1263–1284.

[51] He, J. *Rare Category Analysis*. PhD thesis, Carnegie Mellon University, 2010.

[52] Iba, W., and Langley, P. Induction of one-level decision trees. In *ML* (1992), Citeseer, pp. 233–240.

[53] Japkowicz, N., et al. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets* (2000), vol. 68.

[54] Japkowicz, N., Myers, C., Gluck, M., et al. A novelty detection approach to classification. In *International Joint Conference on Artificial Intelligence* (1995), vol. 14, pp. 518–523.

[55] Japkowicz, N., and Stephen, S. The class imbalance problem: A systematic study. *Intell. Data Anal. 6*, 5 (Oct. 2002), 429–449.

[56] Jo, T., and Japkowicz, N. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter 6*, 1 (2004), 40–49.

[57] Joshi, M. V. *Learning classifier models for predicting rare phenomena.* PhD thesis, University of Minnesota, 2002.

[58] Kearns, M. J., and Vazirani, U. V. *An introduction to computational learning theory.* MIT Press, 1994.

[59] Khalilia, M., Chakraborty, S., and Popescu, M. Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making 11*, 1 (2011), 51.

[60] Kohavi, R., and Provost, F. Glossary of terms. *Machine Learning 30*, 2-3 (1998), 271–274.

[61] Kuang, D., Ding, C., and Park, H. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of 2012 SIAM International Conference on Data Mining* (2012), pp. 106–117.

[62] Kuang, D., Ling, C. X., and Du, J. Foundation of mining class-imbalanced data. In *Proceedings of the 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part I* (2012), PAKDD'12, pp. 219–230.

[63] Kubat, M., Holte, R. C., and Matwin, S. Machine learning for the detection of oil spills in satellite radar images. *Mach. Learn. 30*, 2-3 (1998), 195–215.

[64] Kubat, M., Matwin, S., et al. Addressing the curse of imbalanced training sets: one-sided selection. In *proceedings of the international conference on Machine learning* (1997), pp. 179–186.

[65] Kulis, B., Saenko, K., and Darrell, T. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 1785–1792.

[66] Kullback, S., and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), 79–86.

[67] Lang, K. Newsweeder: Learning to filter netnews. In *proceedings of the 12th International Machine Learning Conference* (1995), pp. 331–339.

[68] Lee, D. D., and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature 401*, 6755 (Oct 1999), 788–791.

[69] Little, M. A., McSharry, P. E., Hunter, E. J., Spielman, J., and Ramig, L. O. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *Biomedical Engineering, IEEE Transactions on 56*, 4 (2009), 1015–1022.

[70] Littlestone, N., and Warmuth, M. K. The weighted majority algorithm. In *proceedings of the 30th Annual Symposium on Foundations of Computer Science* (1989), pp. 256–261.

[71] Luo, W., Li, X., Li, W., and Hu, W. Robust visual tracking via transfer learning. In *ICIP* (2011), pp. 485–488.

[72] MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), vol. 1, pp. 281–297.

[73] Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. In *Advances in neural information processing systems* (2008), pp. 1041–1048.

[74] Mitchell, T. *Machine Learning*. McGraw-Hill, 1997.

[75] Olshausen, B. A., and Field, D. J. Sparse coding of sensory inputs. *Current opinion in neurobiology 14*, 4 (Aug. 2004), 481–487.

[76] Orriols, A., and Bernadó-Mansilla, E. The class imbalance problem in learning classifier systems: a preliminary study. In *Proceedings of the 2005 workshops on Genetic and evolutionary computation* (2005), pp. 74–78.

[77] Pan, S. J., Kwok, J. T., and Yang, Q. Transfer learning via dimensionality reduction. In *proceedings of the national conference on Artificial intelligence* (2008), pp. 677–682.

[78] Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. Domain adaptation via transfer component analysis. In *proceedings of the 21st international jont conference on Artifical intelligence* (2009), pp. 1187–1192.

[79] Pan, S. J., and Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering 22*, 10 (2010), 1345–1359.

[80] Pardo, L. *Statistical inference based on divergence measures*, vol. 185. Chapman & Hall/CRC, 2005.

[81] Pardoe, D., and Stone, P. Boosting for regression transfer. In *proceedings of the 27th International Conference on Machine Learning* (2010), pp. 863–870.

[82] Perkins, D. N., and Salomon, G. Transfer of learning. *International encyclopedia of education 2* (1992).

[83] Provost, F. Machine learning from imbalanced data sets 101. In *of the Am. Assoc. for Artificial Intelligence Workshop* (2000).

[84] Qi, G.-J., Aggarwal, C., Rui, Y., Tian, Q., Chang, S., and Huang, T. Towards cross-category knowledge propagation for learning visual concepts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 897–904.

[85] Quanquan, G., Zhenhui, L., and Han, J. Learning a kernel for multi-task clustering. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI)* (2011), AAAI '11, pp. 000–000.

[86] Quinlan, J. R. Induction of decision trees. *Machine learning 1*, 1 (1986), 81–106.

[87] Quinlan, J. R. *C4. 5: programs for machine learning*, vol. 1. Morgan kaufmann, 1993.

[88] Rettinger, A., Zinkevich, M., and Bowling, M. H. Boosting expert ensembles for rapid concept recall. In *AAAI* (2006), pp. 464–469.

[89] Rijsbergen, C. J. V. *Information Retrieval*, 2nd ed. Butterworth-Heinemann, Newton, MA, USA, 1979.

[90] Rokach, L., and Maimon, O. Decision trees. *Data Mining and Knowledge Discovery Handbook* (2005), 165–192.

[91] Rosenstein, M. T., Marx, Z., Kaelbling, L. P., and Dietterich, T. G. To transfer or not to transfer. In *proceedings of the NIPS 2005 Workshop* (2005).

[92] Roumani, Y., May, J., Strum, D., and Vargas, L. Classifying highly imbalanced icu data. *Health Care Management Science* (2012), 1–10.

[93] Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. *Computer Vision-ECCV 2010* (2010), 213–226.

[94] Seung, D., and Lee, L. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems 13* (2001), 556–562.

[95] Shi, X., Fan, W., and Ren, J. Actively transfer domain knowledge. In *proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II* (2008), pp. 342–357.

[96] Stevens, L., Coresh, J., Greene, T., and Levey, A. S. Assessing kidney function–measured and estimated glomerular filtration rate. *New England Journal of Medicine 354*, 23 (June 2006), 2473–83.

[97] Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics 60*, 4 (2008), 699–746.

[98] Sun, Y., Kamel, M. S., Wong, A. K., and Wang, Y. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition 40*, 12 (2007), 3358 – 3378.

[99] Tabar, L., Fagerberg, C., Gad, A., Baldetorp, L., Holmberg, L., Gröntoft, O., Ljungquist, U., Lundström, B., Månson, J., Eklund, G., et al. Reduction in mortality from breast cancer after mass screening with

mammography. randomised trial from the breast cancer screening working group of the swedish national board of health and welfare. *Lancet 1*, 8433 (1985), 829.

[100] Thorndike, E., and Woodworth, R. The influence of improvement in one mental function upon the efficiency of other functions. ii. the estimation of magnitudes. *Psychological Review 8*, 4 (1901), 384.

[101] Thrun, S., and Pratt, L. *Learning To Learn*. Kluwer Academic Publishers, November 1997.

[102] Trowbridge, M. H., and Cason, H. An experimental study of thorndike's theory of learning. *The Journal of General Psychology 7*, 2 (1932), 245–260.

[103] Tsuboi, Y., Kashima, H., Hido, S., Bickel, S., and Sugiyama, M. Direct density ratio estimation for large-scale covariate shift adaptation. *Information and Media Technologies 4*, 2 (2009), 529–546.

[104] Van der Vaart, A. W. *Asymptotic statistics*, vol. 3. Cambridge university press, 2000.

[105] Van Rijsbergen, C. J. *Information retrieval / C. J. van Rijsbergen*. Butterworths, London ; Boston :, 1975.

[106] Vapnik, V. N., and Ya. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and its Applications 16*, 2 (1971), 264–280.

[107] Venkatesan, A., Krishnan, N., and Panchanathan, S. Cost-sensitive boosting for concept drift. In *proceedings of the 2010 International Workshop on Handling Concept Drift in Adaptive Information Systems* (2010).

[108] Veronesi, U., Paganelli, G., Galimberti, V., Viale, G., Zurrida, S., Bedoni, M., Costa, A., de Cicco, C., Geraghty, J. G., Luini, A., et al. Sentinel-node biopsy to avoid axillary dissection in breast cancer with clinically negative lymph-nodes. *Lancet 349*, 9069 (1997), 1864.

[109] Vieriu, R.-L., Rajagopal, A., Subramanian, R., Lanz, O., Ricci, E., Sebe, N., and Ramakrishnan, K. Boosting-based transfer learning for multi-view head-pose classification from surveillance videos. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO),* (2012), pp. 649–653.

[110] Wall, P. Non-invasive optical spectroscopy and imaging of human brain function. *Trends Neurosci 20* (1997), 324–325.

[111] Wang, D., Li, T., Zhu, S., and Ding, C. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), ACM, pp. 307–314.

[112] Wang, Y., and Xiao, J. Transfer ensemble model for customer churn prediction with imbalanced class distribution. In *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on* (2011), vol. 3, IEEE, pp. 177–181.

[113] Waters, D. Spam overwhelms e-mail messages. *BBC News* (April 2009).

[114] Weinreich, U. *Languages in contact: Findings and problems*. De Gruyter Mouton, 1979.

[115] Weiss, G. M. Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl. 6*, 1 (2004), 7–19.

[116] Weiss, G. M. Mining with rare cases. In *Data Mining and Knowledge Discovery Handbook*. Springer, 2010, pp. 747–757.

[117] Weiss, G. M., and Provost, F. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research 19*, 1 (Oct 2003), 315–354.

[118] Wiberley, S. E., Colthup, N., and Daly, L. Introduction to infrared and raman spectroscopy. *ed. NB Colthup and LH Daly, Academic Press, Inc., San Diego* (1990).

[119] Wu, P., and Dietterich, T. G. Improving svm accuracy by training on auxiliary data sources. In *proceedings of the twenty-first international conference on Machine learning* (2004), pp. 871–878.

[120] Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., et al. Top 10 algorithms in data mining. *Knowledge and Information Systems 14*, 1 (2008), 1–37.

[121] Xie, S., Lu, H., and He, Y. Multi-task co-clustering via nonnegative matrix factorization. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (2012), IEEE, pp. 2954–2958.

[122] Xu, W., Liu, X., and Gong, Y. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (2003), ACM, pp. 267–273.

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

[124] Yang, Y. An evaluation of statistical approaches to text categorization. *Information retrieval 1*, 1 (1999), 69–90.

[125] Yang, Y., and Barron, A. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics 27*, 5 (1999), 1564–1599.

[126] Yao, Y., and Doretto, G. Boosting for transfer learning with multiple sources. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 1855–1862.

[127] Yao, Y., and Doretto, G. Boosting for transfer learning with multiple sources. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (2010), IEEE, pp. 1855–1862.

[128] Zha, Z.-J., Mei, T., Wang, M., Wang, Z., and Hua, X.-S. Robust distance metric learning with auxiliary knowledge. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (2009), pp. 1327–1332.

[129] Zhang, J., and Zhang, C. Multitask bregman clustering. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (2010), pp. 655–660.

[130] Zhang, Y., and Yeung, D.-Y. Transfer metric learning by learning task relationships. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), ACM, pp. 1199–1208.

ABSTRACT

LEARNING WITH AN INSUFFICIENT SUPPLY OF DATA VIA KNOWLEDGE TRANSFER AND SHARING

by

SAMIR AL-STOUHI

December 2013

Advisor: Abhilash Pandya & Chandan K. Reddy

Major: Computer Engineering

Degree: Doctor of Philosophy

As machine learning methods extend to more complex and diverse set of problems, situations arise where the complexity and availability of data presents a situation where the information source is not "adequate" to generate a representative hypothesis. Learning from multiple sources of data is a promising research direction as researchers leverage ever more diverse sources of information. Since data is not readily available, knowledge has to be transferred from other sources and new methods (both supervised and un-supervised) have to be developed to selectively share and transfer knowledge. In this dissertation, we present both supervised and un-supervised techniques to tackle a problem where learning algorithms cannot generalize and require an extension to leverage knowledge from different sources of data. Knowledge transfer is a difficult problem as diverse sources of data can overwhelm each individual dataset's distribution and a careful set of transformations has to be applied to increase the relevant knowledge at the risk of biasing a dataset's distribution and inducing negative transfer that can degrade a learner's performance.
We give an overview of the issues encountered when the learning dataset does not have a sufficient supply of training examples. We categorize the structure of small datasets and highlight the need for further research. We present an instance-transfer supervised classification algorithm to improve classification performance in a target dataset via knowl-

edge transfer from an auxiliary dataset. The improved classification performance of our algorithm is demonstrated with several real-world experiments. We extend the instance-transfer paradigm to supervised classification with "Absolute Rarity", where a dataset has an insufficient supply of training examples and a skewed class distribution. We demonstrate one solution with a transfer learning approach and another with an imbalanced learning approach and demonstrate the effectiveness of our algorithms with several real world text and demographics classification problems (among others). We present an unsupervised multi-task clustering algorithm where several small datasets are simultaneously clustered and knowledge is transferred between the datasets to improve clustering performance on each individual dataset and we demonstrate the improved clustering performance with an extensive set of experiments.

AUTOBIOGRAPHICAL STATEMENT

SAMIR AL-STOUHI

Samir Al-Stouhi a doctoral student at Wayne State University where he is a member of the Data Mining and Knowledge Discovery Lab. His research interests include transfer learning, multi-task learning, imbalanced learning, self-taught learning and alternative clustering. His research specifically addresses learning when data is not readily available and is only possible via knowledge transfer and sharing. Prior to joining the lab, he was in the Smart Sensors and Integrated Micro-Systems Program at Wayne State University. Samir has over 12 years of experience in embedded software and signal processing.