Mining a Shared Concept Space for Domain Adaptation in Text Mining

CHEN, Bo

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in

Systems Engineering and Engineering Management

The Chinese University of Hong Kong

March 2011

UMI Number: 3492009

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI
Dissertation Publishing

UMI 3492009

Copyright 2011 by ProQuest LLC.

All rights reserved. This edition of the work is protected against
unauthorized copying under Title 17, United States Code.

ProQuest

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

論文評審委員會

于旭 教授 （主席）
林偉 教授 （論文導師）
蘇文藻 教授（委員）
張國威 教授 （委員）

Thesis /Assessment Committee

Professor Yu Xu, Jeffrey (Chair)
Professor Wai Lam (Thesis Supervisor)
Professor So Man-Cho Anthony (Committee Member)
Professor William Kwok-Wai Cheung (External Examiner)

ABSTRACT

In many text mining applications involving high-dimensional feature space, it is difficult to collect sufficient training data for different domains. One strategy to tackle this problem is to intelligently adapt the trained model from one domain with labeled data to another domain with only unlabeled data. This strategy is known as domain adaptation. However, there are two major limitations of the existing domain adaptation approaches. The first limitation is that they all separate the domain adaptation framework into two separate steps. The first step attempts to minimize the domain gap, and then the second step is to train the predictive model based on the reweighted instances or transformed feature representation. However, such a transformed representation may encode less information affecting the predictive performance. The second limitation is that they are restricted to using the first-order statistics in a Reproducing Kernel Hilbert Space (RKHS)to measure the distribution difference between the source domain and the target domain. In this thesis, we focus on developing solutions for those two limitations hindering the progress of domain adaptation techniques.

we develop a novel model to learn a low-rank shared concept space with respect to two criteria simultaneously: the empirical loss in the source domain, and the embedded distribution gap between the source domain and the target domain. Besides, we can transfer the predictive power from the extracted common features to the characteristic features in the target domain by the feature graph Laplacian. Moreover, we can kernelize our proposed method in the Reproducing Kernel Hilbert Space (RKHS) so as to generalize our model by making use of the

powerful kernel functions. We theoretically analyze the expected error evaluated by common convex loss functions in the target domain under the empirical risk minimization framework, showing that the error bound can be controlled by the expected loss in the source domain, and the embedded distribution gap.

Then we propose an improved symmetric Stein's loss (SSL) function which combines the mean and covariance discrepancy into a unified Bregman matrix divergence of which Jensen-Shannon divergence between normal distributions is a particular case. Based on our proposed distribution gap measure based on second-order statistics, we present another new domain adaptation method called Location and Scatter Matching. The target is to find a good feature representation which can reduce the embedded distribution gap measured by SSL between the source domain and the target domain, at the same time, ensure the new derived representation can encode sufficient discriminants with respect to the label information. Then a standard machine learning algorithm, such as Support Vector Machine (SVM), can be adapted to train classifiers in the new feature subspace across domains.

We conduct a series of experiments on real-world datasets to demonstrate the performance of our proposed approaches comparing with other competitive methods. The results show significant improvement over existing domain adaptation approaches.

摘要

在许多包含高维特征空间的文本挖掘应用中，有的时候为不同的领域采集足够多的标注数据会特别困难。对付这种问题，有一种策略那就是智能地 改变在有标注数据领域（源领域）上学习的模型，从而使其适应到其他没有标注数据的领域（目标领域）上去。这种策略就是机器学习中一种新 型的学习模式，领域适应。在现有的领域适应方法中，普遍存在着两种不足。第一个不足之处就是基本上所有的方法都会分成两个步骤，第一 步就是学习一种变换，包括样本加权和特征加权，使得两个领域的内在分布保持一致；第二步就在这个保持分布一致的空间上，训练出预测模型，从而用到另外一个没有标注 数据的领域上去。很显然的，为了使两个领域保持一致的变换势必会人为地丢失掉很多有用的信息，因为第一步根本就没有用到标注的信息，从而 使得训练出来的模型在目标领域上面表现不佳。第二个不足之处就是在衡量两个领域的分布差异时，已有的方法都只是在生成的希尔伯特核空间里 考虑一阶统计量的差异，然而这个统计量还不足已去衡量两个分布的差异。在这篇论文里，我们主要专注于怎么样消灭这两个阻碍领域适应算法发展 的拦路虎。

首先我们研究了一个新颖的方法去学习一个低秩的共有概念空间，学习的时候主要考虑两个指标。第一个就是源领域和目标领域嵌入到这个共有概念空间 之后的隐含分布差异会最小，第二个就是在这个空间里面，标注的信息和学习出来的信息误差会达到最小。这样的话最终学习出来的模型就可以无缝隙的 推广到目标领域上去。同时，我们可以从挖掘的共有特征上传递很多预测能量到目标领域里特有的特征上去，这是通过特征的拉普拉斯图进行实现的。除此之外， 我们可以核函数化我们提出的方法到生成的希尔伯特核空间上去，从而使得我们的模型有更好的推广性。我们还在经验风险最小的框架上

面理论地分析了我们这个 凸函数 模型在目标领域上的误差上限。这个上限表明它可以被源领域标注误差，以及嵌入空间里源领域和目标领域之间的差异很好地控制住。

然后针对第二个不足之处，我们提出了一个对称的Stein损失函数，可以同时考虑均值差异和协方差矩阵差异，并且可以将他们映射到一个统一的Bregman矩阵散度上去。 它非常的宽泛，经典的Jensen-Shannon散度在两个正态分布上的值是我们提出的衡量方法的特殊情况。基于我们提出的衡量领域差异的指标，我们提出了方位和散度同时匹配的 领域适应算法。学习过程基于的指标跟第一个算法一样，挖掘出一个共同的概念空间，使得源领域和目标领域在新的衡量标准下分布差异最小，同时兼顾源领域的标注信息。 最后用经典的机器学习方法，例如支持向量机，去训练最终的模型。

我们在现实的数据集上面实施了一系列的实验，通过比较其他的方法去展示我们提出的方法的优势。结果表明了我们的方法相对于其他领域适应算法有着非常明显的提高。

Acknowledgements

I would like to thank all people who have helped and inspired me during my doctoral study. First of all, I am profoundly grateful to my advisor, professor Wai Lam, for his guidance during my research and study at the Chinese University of Hong Kong. His tireless pursuit of excellence in research, teaching, advising, and every other aspect of his academic work is truly inspirational. I am indebted to Wai Lam for priceless and copious advice about selecting interesting problems, making progress on difficult ones, pushing ideas to their full development, writing and presenting results in an engaging manner. I would like to thank my thesis committee members, for their excellent suggestions and thought-provoking questions. I have learned a great deal from their work and their influence on this thesis is immense. I further express my gratitude to professors in the Department of Systems Engineering and Engineering Management for enriching my knowledge.

I would like to express my appreciation and gratitude to all my collaborators for their excellent ideas, hard work and dedication. Ivor Tsang who inspired me with new ideas whenever I discussed with him. Tak-Lam Wong who taught me many research and project experience. I would like to thank my friends for their great spiritual support, encourage, advices. I remember all the shared time with you in CUHK, in SEEM.

Most of all, I am grateful to my parents who have been giving me love, warmth, unbending support and constant encouragement in my life. Many thanks to all of them!

Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1. Motivation

In many text mining applications involving high-dimensional feature space, it is difficult to collect sufficient training data for different domains. For example, suppose we wish to build an employment analysis system. There is a major component that conducts text information extraction from recruitment Web sites. To handle a massive amount of information, we can develop a learning model whose aim is to learn information extraction patterns to extract precise job information related to a particular field such as job title, duty, requirement, etc, in different industries. A more effective training strategy is to prepare different training data for different industries so that tailor-made information extraction patterns can be learned for each industry. However, typically we may just have few experts who can accurately annotate the information in one specific industry like accounting. The learned model deployed obviously cannot perform well in other industries such as logistic or health care due to the distribution of the terms in each industry is different. One strategy to tackle this problem is to intelligently adapt the trained extraction model from one industry to another industry. Another real-world example comes from the email spamming system, where we know that the general spam filters can be trained on some public collection of spam emails. But when the trained spam filter is applied to an individual person or newsgroup's

inbox, the filtering performance may degenerate seriously. Therefore, we need to personalize the spam filter. Specifically, we should adapt the spam system to fit the person or newsgroup's own email distribution by discovering the shared pattern's and discriminative patterns.

1.2. Domain Adaptation

Adapting text mining models can be treated as a kind of domain adaptation problems. Different from the traditional statistical learning settings, which assume that the training data and the operational (testing) data are drawn from the same underlying distribution, the operational data in domain adaptation setting, sampled from one domain, has different underlying distribution with the training data sampled from another domain.

Definition 1. *(Domain Adaptation). Given a source domain D_S and a corresponding learning task T_S, a target domain D_T and a corresponding learning task T_T, domain adaptation aims to improve the learning of the target predictive function $f_T(.,.)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ and $T_S = T_T$. In addition, sufficient unlabeled samples in D_T must be available when training.*

The above definition indicates that domain adaptation can be regarded as a transductive learning algorithm which tries to make the best use of the unlabeled data in the target domain. Besides, the source and target tasks are the same so that one can adapt the learnt prediction function in the source domain to the target domain through the distribution matching on the data samples. Here we state that $D_S \neq D_T$ without giving the feature space definition. Obviously, the setting can be divided into two cases, (1) the feature spaces in D_S and D_T are different; (2) the feature spaces in D_S and D_T are same, but the corresponding marginal distributions are different. Generally speaking, most of the domain adaptation techniques are related to case (2), including our proposed methods

in this thesis. Arnold *et al.* investigated the difference between traditional transductive learning setting and domain adaptation in detail in [2].

1.3. Drawbacks of Existing Approaches

Distinction between training and testing distributions in a learning problem has been referred to as *sample selection bias* [37] or *covariate shift* [56, 57]. Sample selection bias actually refers to the fact that the training instances are originally drawn from the testing distribution, but sampled as training data with probability. Covariate shift is a particular sample selection bias which allows different distributions of the instances between the training and testing set, but it assumes that the conditional probabilities of the label variables given an instance remain unchanged. There are two main approaches to removing the bias, namely, instance-level approach and feature-level approach.

Instance-level approaches [37, 56, 57] infer the re-sampling weight of training samples by matching the distributions between training and testing sets in the original feature space. Huang *et al.* [37] proposed a Kernel Mean Matching (KMM) to learn the re-sampling weights directly by matching the means between the source domain data and the target domain data in a Reproducing Kernel Hilbert Space (RKHS). Sugiyama *et al.*. [57] proposed a Kullback-Leibler Importance Estimation Procedure (KLIEP) to estimate the re-sampling weights directly by minimizing the Kullback-Leibler divergence between the two domains.

Feature-level approaches [9, 48, 49] try to learn an optimal feature representation where the marginal distributions between the data in different domains are closely matched. Then with the new feature representation, the performance of the target task is expected to improve significantly. Blitzer *et al.* [9] proposed a heuristic method to select some domain independent pivot features to learn an embedded space where the data coming from both domains can share the same feature structure. Pan *et al.* [48] exploited the Maximum Mean Discrepancy Embedding (MMDE) method to learn a low-dimensional space to reduce the

distribution difference between domains for domain adaptation. Their following work on Transfer Component Analysis (TCA) tries to learn a set of common transfer components across domains for reducing the distribution gap [49].

Both the instance-level and feature-level approaches in domain adaptaion try to reduce the distribution gap between the training and testing set so as to propagate the label information and they have been demonstrated to be effective in various applications. However, generally, for both kinds of approaches, they all separate the domain adaptation framework into two separate steps. The first step attempts to minimize the domain gap, and then the second step is to train the predictive model based on the reweighted instances or transformed feature representation. They do not consider these two steps in a unified framework. However, such a transformed representation may encode less information leading to an increase of the empirical loss on the labeled data. While the use of labels in linear discriminant analysis usually helps extract more discriminative features, the label information from the source domains may be also useful to learn kernels or extract features for better domain adaptation.

1.4. Contributions

1.4.1. Discovering Low-Rank Shared Concept Space

In Chapter 3, we develop a novel model to learn a low-rank shared concept space with respect to two criteria simultaneously: the empirical loss in the source domain, and the embedded distribution gap between the source domain and the target domain. We call our method Low-Rank Shared Concept (LRSC) domain adaptation method. Consider again the job information extraction example. For the task of extracting the job requirement information in the domain of accounting, the most representative terms are "qualified", "year", "experience", "CPA", "CA", "ACCA", etc. For the domain of health care, the representative terms shift to "qualified", "degree", "year", "CCP", "Physiology", "experience",

etc. If we can extract the shared domain independent features such as "qualified", "year", "experience", then the learned model for extraction can be effectively adapted to the domain of health care.

Our proposed framework discovers the low-rank shared concept space, where the empirical loss on the labeled data, as well as the distribution gap between the source domain and the target domain, are jointly minimized. Besides, we can transfer the predictive power from the extracted common features to the characteristic features in the target domain by the feature graph Laplacian. Moreover, we can kernelize our proposed method in the Reproducing Kernel Hilbert Space (RKHS) so as to generalize our model by making use of the powerful kernel functions [54]. Therefore, it can be applied to fit some special kinds of data, like graphs, strings, where linear space usually fails to model them. The proposed alternating optimization strategy can solve our model efficiently. Moreover, our model is very general in selecting the loss functions like Hinge loss and sparse loss, instead of the least square loss function, as the computation of the empirical label loss for the labeled data.

We theoretically analyze the expected error evaluated by common convex loss functions in the target domain under the empirical risk minimization framework, showing that the error bound can be controlled by the expected loss in the source domain, and the embedded distribution gap. We prove that minimizing the objective function is very reasonable for domain adaptation and it can lead to good prediction performance in the target domain. Our model is also capable of considering multiple classes and their interactions simultaneously. We have conducted extensive experiments on two common text mining problems, namely, information extraction and document classification to demonstrate the effectiveness of our proposed method [16, 62, 18, 17, 69].

1.4.2. Location and Scatter Matching by High-Order Statistics

Even though these empirical mean based domain adaptation methods have achieved encouraging performance in various applications, there still exists some technical difficulties hindering even better results. First it is extremely hard to estimate the density function especially when the feature space is of high dimensional. Another major difficulty is how to incorporate an effective statistical criterion measuring distribution discrepancy into a tractable framework. Currently, most existing instance-level and feature-level approaches are restricted to the first-order statistics matching and enforce the empirical means of the training and testing instances be closer in a Reproducing Kernel Hilbert Space (RKHS). However, intuitively, they may have a considerable limitation in matching two probability distributions where only the first-order statistics are exactly the same. Moreover, for many text mining applications, it is not appropriate to ignore the feature dependency which can be explored by considering the document covariance. Specifically, we can observe that the sample covariance matrix on text data with zero mean is exactly the same as the feature similarity matrix. This motivates us to utilize the covariance information to evaluate the distribution discrepancy. First it can strengthen the distribution matching criterion instead of only considering the mean. The second advantage is that we can utilize the feature dependency to distinguish domain specific features and common features, and then filter such features whose similarity with other features varies greatly from the training data to the testing data by investigating the sample covariance matrices.

In order to overcome the limitations mentioned above, we develop a new non-parametric distance metric call symmetric Stein's loss (SSL) to empirically measure the distribution gap between two domains with finite samples [19]. It jointly considers the empirical mean (Location) and sample covariance (Scatter) difference, and it can map the location and scatter information to one matrix

smoothly which can avoid treating them separately. More specifically, we propose an improved symmetric Stein's loss (SSL) function which combines the mean and covariance discrepancy into a unified Bregman matrix divergence of which Jensen-Shannon divergence between normal distributions is a particular case. In Chapter 4, we state the detailed description of our proposed model free distribution gap measure. Moreover, we test the properties of SSL on both synthetic dataset and real-world dataset from different aspects, like convergence, sensitivity, and generalization ability. We also conduct two-sample tests on a real-world microarray dataset comparing with other competitive methods, such as t-test, maximum mean discrepancy, and so on.

Based on our proposed distribution gap measure, the symmetric Stein's loss (SSL), we present another new domain adaptation method called Location and Scatter Matching (LSM) in Chapter 5. The target is to find a good feature representation which can reduce the embedded distribution gap measured by SSL between the source domain and the target domain, at the same time, ensure the new derived representation can encode sufficient discriminants with respect to the label information. Then a standard machine learning algorithm, like Support Vector Machine (SVM), can be adapted to train classifiers in the new feature subspace across domains. We also conduct a group of experiments on real-world datasets to demonstrate the performance comparing with other competitive methods.

1.5. Thesis Outline

The rest of the chapters in the thesis is organized as follows:

Chapter 2. Related Work: In this chapter, we review some related methods for domain adaptation.

Chapter 3. Discovering Low-Rank Shared Concept Space for Adaptation: In this chapter, we present our proposed domain adaptation method which directly minimizes both the distribution gap between the source

domain and the target domain, as well as the empirical loss on the labeled data in the source domain by extracting the low-rank concept subspace. Furthermore, we apply the graph Laplacian [3] to exploit the predictive power for some domain dependent representative features in the target domain based on the co-occurrence with the shared features.

Chapter 4. Modeling Domain Difference Using High-order Statistics: In this chapter, we develop a new non-parametric distance metric called symmetric Stein's loss (SSL) to empirically measure the distribution gap between two domains with finite samples. It jointly considers the empirical mean (Location) and sample covariance (Scatter) difference, and it can map the location and scatter information to a unified framework. A diverse set of statistical tests are conducted to demonstrate the properties of our proposed distribution gap measure.

Chapter 5. Location and Scatter Matching for Domain Adaptation. In this chapter, we present another new domain adaptation method called Location and Scatter Matching (LSM), which targets at finding a good feature representation that can reduce the embedded distribution gap measured by our proposed criteria SSL, at the same time, ensure the new derived representation can encode sufficient discriminants with respect to the label information in the target domain.

Chapter 6. Conclusion and Future Works. In this chapter, we review the main contributions of the thesis and summarize their significance. We discuss some potential extensions and future research directions.

CHAPTER 2

Related Works

In this chapter, we first review some traditional semi-supervised learning methods which didn't consider the domain difference. Then we present some existing methods for measuring the domain difference. Finally we describe some existing state-of-the-art domain adaptation techniques.

2.1. Semi-Supervised Learning

If we ignore the domain difference, and treat the labeled source domain instances as labeled data and the unlabeled target domain instances as unlabeled data, then it is reduced to the standard semi-supervised learning (SSL) problem. We can then apply any SSL algorithms [73, 13] to the domain adaptation problem. The difference between SSL and domain adaptation is that (1) the amount of labeled data in SSL is small but large in domain adaptation, and (2) the labeled data may be noisy in domain adaptation whereas in SSL the labeled data is all reliable. There has been some work extending semi-supervised learning methods for domain adaptation. Dai *et al.* [20] proposed an EM-based algorithm for domain adaptation, which can be shown to be equivalent to a semi-supervised EM algorithm [47] except that they proposed to estimate the trade-off parameter between the labeled and the unlabeled data using the KL-divergence between the two domains. Jiang and Zhai [40] proposed to not only include weighted

source domain instances but also weighted unlabeled target domain instances in training, which essentially combines instance weighting with bootstrapping. Xing *et al.* [63] proposed a bridged refinement method for domain adaptation using label propagation on a nearest neighbor graph, which has resemblance to graph-based semi-supervised learning algorithms.

2.2. Measuring Domain Difference

Recall that, in domain adaptation, the fundamental question is how to evaluate the difference in distribution between two domains given finite observations of each domain. There exists many criteria that can be used to measure their distance, such as the Kullback-Leibler (KL) divergence:

$$\mathrm{KL}(\mathcal{P}(x)||\mathcal{Q}(x)) = \int \mathcal{P}(x)\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}dx \tag{2.1}$$

where $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ denote the marginal distribution functions of the source domain and the target domain respectively for data samples x. It can be observed that KL divergence is not symmetric. Then its symmetric form, the Jensen-Shannon divergence was proposed to measure the distribution gap. There are many methods attempting to approximate KL divergence by finite samples empirically. For example, in [59] authors proposed to estimate the densities based on data-dependent histograms with a fixed number of samples from the marginal distributions in each bin. In [22], the authors computed relative frequencies on data-driven partitions achieving local independence for estimating mutual information. However, many of these estimators are parametric and require an intermediate density estimate, which will be very tough or even impossible when the dimensionality is very high, such as in large scale text mining and bioinformatics applications. There are some methods trying to approximate the divergence without estimating the density functions. They directly used the empirical cumulative distribution functions or *k-nearest-neighbour* to approximately represent the density function. Even though the process is much easier than those

methods requiring density estimation, they are usually parametric and need to pre-set some important parameters, such as the nearest neighbour number.

To avoid conducting this non-trivial task, a non-parametric distance estimate between distributions is more desirable. Recently, Gretton *et al.* [32] introduced the Maximum Mean Discrepancy (MMD) for comparing distributions based on the Reproducing Kernel Hilbert Space (RKHS) distance. The MMD between the source domain and the target domain is defined as the superb value in the set of the expected value difference between the source domain and the target domain, measure by a functional set. Usually we restrict the functional set in a unit ball in the RKHS to simplify the discussion. For real-world applications, we can only obtain finite samples in each domain. Hence we need to use the empirically form to calculate the MMD value, which can be formulated as the empirical mean value difference measured by a unit ball bounded function set in RKHS. MMD is very general in many applications, and easy to calculate. it has been proved to be powerful in two-sample test problems and can lead to good performance in real-world applications such as bioinformatics.

2.3. Supervised Domain Adaptation

Many researchers try to enforce the learning processes among multiple domains which all have sufficient labeled data. Those supervised domain adaptation methods can be grouped into two major categories: feature-based approaches [24, 61] and parameter-based approaches [11, 12, 28, 65, 26, 68, 23]. Daumé III proposed the Feature Augmentation method to augment features into the high-dimensional space by kernel-mapping functions for domain adaptation [24]. Then standard discriminative learning method will be employed in the kernel induced space. However the kernel-mapping function is domain dependent and not easy to generalize. Hash kernels was introduced by Weinberger *et al.* to extend the feature augmentation method to large scale problems [61], where the interference between independently hashed subspaces is negligible with high probability, which

allows large-scale statistical learning in a very compressed space. Instead of transferring shared feature information among different domains, others suggested to transferring shared structure prior as parameter information. Finkel and Manning proposed a model to set both domain-specific parameters and global parameters for each feature via hierarchical Bayesian prior [28], which have been shown to be equivalent with the domain adaptation work of [24], but it is more flexible by setting different hyperparameters leading to significant improvement on performance. Li and Bilmes [43] proposed a general Bayesian divergence prior framework for domain adaptation. They then showed how this general prior can be instantiated for generative classifiers and discriminative classifiers. Chelba and Acero [14] applied this kind of Bayesian priors for the task of adapting maximum entropy and maximum entropy Markov models of capitalizer across domains under a maximum "a posteriori" (MAP) framework.

2.4. Unsupervised Domain Adaptation

By making use of the labeled data in the target domain, supervised domain adaptation approaches can obtain encouraging performance as expected. However, in most situations, it is hard to prepare the labeled data for the target domain. Therefore, unsupervised domain adaptation approaches are more applicable to the real-world applications. Generally speaking, the existing unsupervised domain adaptation techniques can be grouped to two directions: feature level approaches which focus on changing the feature representation to reduce the distribution gap, and instance level approaches which focus on weighing the instances considering the distribution difference.

2.4.1. Feature level unsupervised domain adaptation

Many works on domain adaptation try to learn a new feature representation which can bridge the source domain and the target domain. Blitzer *et al.* [9] proposed a structural correspondence learning (SCL) method to select some domain independent pivot features to learn an embedded space where the data coming from both domains can share the same feature structure. If the pivot features are well designed, then the learned embedded space can encode the correspondence between the features from the different domains. Ben-David *et al.* analyzed the error bound of domain adaptation and showed experimentally that SCL can reduce the domain gap [5]. In [9], Blitzer *et al.* used a heuristic method to select pivot features for natural language processing (NLP) problems, such as sentences tagging. In their following works [8], the researchers proposed to use mutual information criteria, instead of the heuristic collection, to choose the high dependence pivot features. Raina *et al.* [51] learned the sparse basis from the unlabeled data which is not necessary in the same domain as the labeled data. Then it represents the labeled data by those learned high-level basis for further classification.

Dai *et al.* [20] proposed a coclustering-based algorithm to propagate the label information across different domains. Ling *et al.* [44] proposed a cross domain spectral clustering (CDSC) method which tries to seek the spectral consistence between the in-domain and out-of-domain structures. Xue *et al.* [64] extended the traditional probabilistic latent semantic analysis (PLSA) [36] algorithm for cross-domain text classification. The extended model is named Topic-bridged PLSA, which can integrate label and unlabeled data in different but related domains, into a unified probabilistic framework. Yang *et al.* [66] proposed Adaptive SVM (A-SVM) to enhance the prediction performance of video concept detection, in which the new SVM classifier is adapted from an existing classifier trained from the auxiliary domain.

A special and simple kind of feature transformation is feature subset se-

lection without any label information. Satpal and Sarawagi [53] proposed a feature subset selection method for domain adaptation, where the criterion for selecting features is to minimize an approximated distance function between the conditional distributions in two domains.

2.4.1.1. MMD Embedding and Transfer Component Analysys

The simple feature re-weighting schemes may have a limited improvement in the target domain when the dimensionality of the data is high. In particular, some features may cause the data distribution between domains to be different, while others may not. Some features may preserve the structure of data for adaptation, while others may not. To address this problem, Pan *et al.* [48] proposed Maximum Mean Discrepancy Embedding (MMDE) for domain adaptation by embedding both the source and target domain data onto a shared low-dimensional latent space. The key idea is to formulate this as a kernel learning problem using the kernel trick, and to learn the kernel matrix by minimizing the distance (measured by MMD) between the source and target domain data. The final model can be solved by Semi-Definite Programming (SDP).

After that, the embedding of data can be extracted by performing eigen-decomposition on the learned kernel matrix, and can be further used for training classifiers by standard classification methods, like SVM. However, the kernel matrix of samples in MMDE are learned separately using the MMD criterion defined on the input data only without considering any labels. While the use of labels in linear discriminant analysis usually helps extract more discriminative features, the label information from the source domains may be also useful to learn kernels or extract features for a better domain adaptation.

In addition, there are two main limitations associated with MMDE. First, MMDE is transductive and cannot generalize on unseen patterns. Second, it requires to solve an expensive SDP problem. Although polynomial-time solvers are available, current interior-point methods are still too computationally intensive for large-scale SDPs in real applications. Note that only the low dimensional

embedding of the data is extracted from the learned kernel matrix in MMDE, and is then used for the training of the decision classifiers. Therefore, not all components from the learned kernel matrix are required to train the classifiers for domain adaptation. Thus, Pan *et al.* further proposed an efficient feature extraction algorithm, known as Transfer Component Analysis (TCA) to reduce the computational burden to overcome the drawbacks of MMDE.

2.4.2. Instance-level Unsupervised domain adaptation

Besides the feature weighting approaches, several domain adaptation methods suggest to apply the instance weighting technique for domain adaption in various applications [37, 56, 57, 70, 27, 40, 6, 21]. Before we review the instance weighting methods, we should first review the empirical risk minimization (ERM). The basis of classification task is to learn a prediction function which can generate minimal expected loss where the samples contribute equally. However, in domain adaptation, due to the underlying distribution difference, the importance of a instance may change. Therefore, we should find a suitable weighting scheme to train the model in the source domain, so as to achieve high generalization ability.

There exists various ways to estimate the instance weights. Zadrozny [70] directly estimate the sample selection bias by constructing simple classification problems. Fan *et al.* [27] further analyzed the problems by using various classifiers to estimate the instance weights. Huang *et al.* [37] proposed a two-step approach Kernel Mean Matching (KMM). The first step is to diminish the difference of the mean of samples in Reproducing Kernel Hilbert Space (RKHS) between the two domains by re-weighing the samples in the source domain using the Maximum Mean Discrepancy criterion. The second step is to use the standard discriminative classification tools, like weighted SVM, to train the classifier in the source domain with the weighted samples. KMM can avoid performing density estimation which is usually difficult when the size of the data set is small or the dimensionality of the study space is very high. Sugiyama *et al.* proposed a Kullback-Leibler Importance Estimation Procedure (KLIEP) to estimate the

re-sampling weights directly by minimizing the Kullback-Leibler divergence between the two domains [57]. Dai *et al.* extended a traditional Naive Bayesian classifier for the transductive transfer learning problems [21]. Recently, Zhong *et al.* utilized the Kernel Discriminative Analysis (KDA) to make the marginal distributions from two domains closer by re-weighing the labeled data in the source domain for training [72]. More information on instance reweighting for covariate shift can be referred to [50].

2.4.3. Advantages of Our Model Over Existing Approaches

Both existing instance-level and feature-level approaches separate the domain adaptation framework into two separate steps. The first step attempts to minimize the domain gap, and then the second step is to train the predictive model based on the reweighted instances or transformed feature representation. They do not consider these two steps in a unified framework. Such a transformed representation may encode less information leading to an increase of the empirical loss on the labeled data. While the use of labels in linear discriminant analysis usually helps extract more discriminative features, the label information from the source domains may be also useful to learn kernels or extract features for better domain adaptation. Our proposed method LRSC discovers the low-rank shared concept space, where the empirical loss on the labeled data, as well as the distribution gap between the source domain and the target domain, are jointly minimized. Besides, we can transfer the predictive power from the extracted common features to the characteristic features in the target domain by the feature graph Laplacian.

Another drawback of existing instance-level and feature-level approaches is that they are all restricted to using the empirical mean difference in a Reproducing Kernel Hilbert Space (RKHS) to measure the distribution gap. However, intuitively, they may have a considerable limitation in matching two probability

distributions where only the first-order statistics are exactly the same. Our developed non-parametric distance metric called symmetric Stein's loss (SSL) can jointly considers the empirical mean (Location) and sample covariance (Scatter) difference. Based on our proposed SSL measure, we develop another new domain adaptation method called Location and Scatter Matching (LSM), which can lead to significant improvement over other existing domain adaptation techniques.

2.5. Learning bounds in domain adaptation

Some previous works focus on the theoretical analysis on conditions where the classifier trained in the source domain can lead to better performance in the target domain [5, 4, 42, 46]. The first theoretical analysis of the domain adaptation problem was presented by Ben-David *et al.* [5, 4], who gave VC-dimension-based generalization bounds for adaptation in classification tasks. Perhaps, the most significant contribution of this work is the definition and application of a distance between distributions, which is particularly relevant to the problem of domain adaptation and can be estimated from finite samples for a finite VC dimension, as previously shown by Kifer *et al.* [42]. This work was later extended by Blitzer *et al.* [7] who also gave a bound on the error rate of a hypothesis derived from a weighted combination of the source data sets for the specific case of empirical risk minimization. A theoretical study of domain adaptation was also presented by Mansour *et al.* [45, 46], where the analysis deals with the related but distinct case of adaptation with multiple sources, and where the target is a mixture of the source distributions.

CHAPTER 3

Discovering Low-Rank Shared Concept Space

3.1. A Brief Introduction

It can be observed that domain adaptation is reasonable and practical if the distributions between the source domain and the target domain is related. The relationship is mainly based on the fact that there exists a shared concept space in which the embedded distribution of each domain is close enough. Consequently it is very reasonable to believe that a good feature representation is able to encode this concept space and provides strong adaptive power from the source domain to the target domain. On the other hand, such a changed representation may encode less information leading to an increase of the empirical loss on the labeled data. To cope with this problem, we try to learn the ideal shared concept space with respect to two criteria: the empirical loss in the source domain, and the embedded distribution gap between the source domain and the target domain. Consider the job information extraction example. For the task of extracting the job requirement information in the domain of accounting, the most representative terms are "qualified", "year", "experience", "CPA", "CA", "ACCA", etc. Similarly for the domain of health care, the representative terms shift to "qualified", "degree", "year", "CCP", "Physiology", "experience", etc.

If we can automatically extract the shared domain independent features such as "qualified", "year", "experience" for the specific task, then the learnt extractor can be effectively adapted to the domain of health care.

In this chapter we propose a domain adaptation method which directly minimizes both the distribution gap between the source domain and the target domain, as well as the empirical loss on the labeled data in the source domain by extracting the low-rank concept subspace. Maximum Mean Discrepancy (MMD) [32] is adopted to measure the embedded distribution difference between the source domain with sufficient but finite labeled data and the target domain with sufficient unlabeled data. Then our objective is to minimize the empirical loss and the MMD measurement with respect to the parametric family (linear transformation) which parameterizes the embedded feature subspace. Furthermore, we apply the graph Laplacian [3] to exploit the predictive power for some domain dependent representative features in the target domain based on the co-occurrence with the shared features. This technique can help improve the performance especially when the common features are not sufficient in the target domain.

3.2. Problem Definition

In the sequel, we refer to the training set as the source domain $D_S = \{(x_i, y_i)\}_{i=1}^{n_1}$, where $x_i \in \mathbb{R}^d$ is the d dimensional input space, and y_i is the output label. The total number of samples in the source domain is n_1. We also assume that the testing samples are available. Denote the testing set as $D_T = \{x_i\}_{i=n_1+1}^{n_1+n_2}$ and $x_i \in \mathbb{R}^d$ is the input, and the total number of samples in the target domain is n_2. Let $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ (or $\mathcal{P}$ and $\mathcal{Q}$ for short) be the marginal distributions of the input sets from the source and target domains respectively. In general, $\mathcal{P}$ and $\mathcal{Q}$ can be different. For matrix notations, $tr(A)$ denotes the trace of matrix A, and matrix transpose is denoted by the superscript $\top$. A^+ is the pseudo-inverse of matrix A. We investigate the learning problem under multiclass setting, with

m decision classifiers $\{f_l(x)\}_{l=1}^m$. Let us denote the label indicator matrix as $Y \in \mathbb{R}^{n_1 \times m}$, and $Y_{il} = 1$ if the i-th sample belongs to the l-th class ($y_i = l$), and -1 if it is labeled as others ($y_i \neq l$).

3.3. Description of Adaptation Model

3.3.1. Basic Formulation of LRSC

We propose a unified domain adaptation learning framework that is able to find the discriminative concept subspace Θ, and to learn decision classifiers $f_l(x)$'s of all labels simultaneously. In particular, our proposed method minimizes the distribution difference between the samples of the source and target domains after the projection into the subspace Θ (*i.e.* Θx_i), as well as the structural risk functional of the n_1 labeled data from the source domain D_S. Similar to other concept extraction methods, we also let Θ be orthogonal on rows so that $\Theta\Theta^T = I$. As a result, our adaptation model can be formulated as an optimization problem as follows:

$$\begin{aligned} &\min_{f_l,\Theta} \sum_{l=1}^{m} \sum_{i=1}^{n_1} L(f_l(x_i), Y_{il}) + \alpha \sum_{l=1}^{m} \Omega(f_l) + \beta \mathrm{dist}_\Theta(D_S, D_T) \\ &\text{s.t. } \Theta\Theta^\top = I_{r\times r}, \end{aligned} \tag{3.1}$$

Here, the first term is the empirical risk functional of the decision functions f_l's on the labeled data from the source domain D_S, and $L(\cdot)$ is the empirical loss function which can be selected according to the application. The regularizer $\Omega(\cdot)$ controls the complexity of f_l, and the last term measures the distribution difference between the embedding of D_S and D_T. Note that the regularization condition on Θ is transformed into the orthogonal constraint $\Theta\Theta^\top = I_{r\times r}$. Thus there is no need to explicitly include in the objective function. Two tradeoff parameters, namely, $\alpha > 0$ and $\beta > 0$ are introduced to control the fitness of the decision functions, and to balance the difference of distributions from the two domains and the structural risk functional for the labeled patterns, respectively.

Hence, by solving Problem (3.1), the subspace Θ and the decision functions f_l's can be learned at the same time.

3.3.1.1. Design of Decision Function

To capture the label dependency, similar to [1] [39], we define the m decision functions:

$$f_l(x) = \mu_l^\top \Phi(x) + \nu_l^\top \Psi_\theta(x), \; l = 1, \cdots, m \tag{3.2}$$

where Φ is a known feature map projecting the data from the input space $\mathcal{X}$ to a high dimensional space $\mathcal{F}$. The other component Ψ_θ is a parameterized low dimensional space which aims at encoding the shared structure between the source domain D_S and the target domain D_T. The weight vector $\nu_l \in \mathbb{R}^r$ is defined in the projected subspace under the projection Ψ_θ, where r is the number of dimension in the space after the projection Ψ_θ. $\mu_l \in \mathbb{R}^d$ is the weight vector defined in $\mathcal{F}$, where d is the number of dimension in the original feature space. With the parametric form in Equation (3.2) of the m decision classifiers, the learned subspace Ψ_θ can capture the intrinsic structure of label dependency in multiclass problems, the weight vector ν_l is the discriminative direction in the subspace Ψ_θ for each class. In the following, we will discuss both the linear decision function as well as the kernel decision function and integrate them into a unified form.

• **Linear classifier**. We can consider a simple linear form of feature map, where $\theta = \Theta$ is an $r \times d$ dimensional matrix, and $\Psi_\theta(x) = \Theta\Psi(x)$, with a known d-dimensional vector function $\Psi(x)$. Furthermore, following [1], we can consider a simple model $\Phi(x) = \Psi(x) = x$. We now can write the linear predictor as:

$$f_l(x) = \mu_l^\top x + \nu_l^\top \Theta x, \; l = 1, \cdots, m \tag{3.3}$$

• **Kernel classifier**. If we consider kernel learning, and assume that the feature map $\Phi(x)$ and $\Psi(x)$ belong to reproducing kernel Hilbert space, then Equation (3.2) can be kernelized. One strategy is to kernelize the predictor weights μ_l which can be represented as $\sum_{i=1}^{n_1+n_2} \phi_l^i \Phi(x_i)$, where $\{\phi_l^i\}_{i=1}^{n_1+n_2}$ are dual

parameters. For Ψ_θ, first we denote the kernel matrix as $\mathbf{K}_\Psi = \langle \Psi(x_i), \Psi(x_j) \rangle$. Then we introduce the empirical kernel map as discussed in [55]:

$$\begin{aligned} \Psi_e : \quad & \mathcal{X} \to \mathbb{R}^{n_1+n_2} \\ & x \mapsto \mathbf{K}_\Psi(\cdot, x)|_{x_1,\cdots,x_N} = (\mathbf{K}_\Psi(x_1, x), \cdots, \mathbf{K}_\Psi(x_N, x)) \end{aligned} \tag{3.4}$$

where $N = n_1 + n_2$. Finally we can let $\Psi_\theta = \Theta\Psi_e$ where $\Theta \in \mathbb{R}^{r \times (n_1+n_2)}$ is used to transform the empirical kernel vector to a r-dimensional space. Let $\{\psi_l^i\}_{i=1}^r$ denote the weight parameters in the embedded kernel subspace for the l-th class. Hence, the kernelized decision functions become:

$$f_l(x) = \phi_l^\top \mathbf{K}(\cdot, x) + \psi_l^\top \Theta \mathbf{K}_e(\cdot, x), \; l = 1, \cdots, m \tag{3.5}$$

where $\phi_l = [\phi_l^1, \phi_l^2, \cdots, \phi_l^{n_1+n_2}] \in \mathbb{R}^{n_1+n_2}$, $\psi_l = [\psi_l^1, \psi_l^2, \cdots, \psi_l^r] \in \mathbb{R}^r$, and $\mathbf{K}$ is the kernel matrix induced by the kernel mapping Φ: $\mathbf{K}_{i,j} = \langle \Phi(x_i), \Phi(x_j) \rangle$. $\mathbf{K}(\cdot, x) = [\mathbf{K}(x_1, x), \cdots, \mathbf{K}(x_{n_1+n_2}, x)]^\top \in \mathbb{R}^{n_1+n_2}$. Generally speaking, the kernel feature map Φ and Ψ can be different. We simplify the computation by setting them to be the same. Therefore, $\mathbf{K}(\cdot, x) = \mathbf{K}_e(\cdot, x)$, and the kernelized decision function in Equation (3.5) can be further formulated as:

$$f_l(x) = \phi_l^\top \mathbf{K}(\cdot, x) + \psi_l^\top \Theta \mathbf{K}(\cdot, x), \; l = 1, \cdots, m \tag{3.6}$$

• **Unified form**. In order to model the linear case in Equation (3.3) and kernel case in Equation (3.6) into a common framework, we introduce:

$$w_l = \begin{cases} \mu_l + \Theta^\top \nu_l, & \text{(linear)} \\ \phi_l + \Theta^\top \psi_l, & \text{(kernel)} \end{cases} \tag{3.7}$$

Moreover, in the following, we use two symbols, namely, u_l and v_l, where u_l denotes μ_l in the linear case and denotes ϕ_l in the kernel case, and v_l denotes ν_l in the linear case and denotes ψ_l in the kernel case. Then Equation (3.7) becomes $w_l = u_l + \Theta^\top v_l$ for both linear and kernel cases. And we can represent the data in linear space and kernel space as follows:

$$\vec{x} = \begin{cases} x, & \text{(linear)} \\ \mathbf{K}(\cdot, x), & \text{(kernel)} \end{cases} \tag{3.8}$$

As a result, we can formulate the predictors, linear form as in Equation (3.3) and kernel form as in Equation (3.6), in a unified form as depicted in Equation (3.9):

$$f_l(x) = w_l^\top \vec{x},\ l\ =\ 1, \cdots, m. \tag{3.9}$$

where

$$w_l = u_l + \Theta^\top v_l, l = 1, \cdots, m \tag{3.10}$$

3.3.1.2. Design of Loss Function

For the empirical loss function on the labeled data

$$L:\ \mathcal{R} \times Y \to \mathbb{R}^+$$

represents the loss on replacing the true label y by the predicting value $f(x)$. The choice of the loss function typically depends on the application but must satisfy the convex assumption. We can consider the following loss functions

•**Sparse loss.** $L(f(x), y) = |y - f(x)| = |1 - yf(x)|$.

•**Square loss.** $L(f(x), y) = (y - f(x))^2 = (1 - yf(x))^2$.

•**Hinge loss.** $L(f(x), y) = \max(0, 1 - yf(x))$.

•**Logistic loss.** $L(f(x), y) = \log_2(1 + e^{-yf(x)})$

for classification. Fig. 3.1 depicts the four loss functions. All of these four loss functions are convex with respect to the variable $yf(x)$ and can lead to tractable optimization strategy, and they have been widely used in many problems. For example, the hinge loss is the most common loss function in SVM, the logistic loss and the square loss are usually applied in regression. Furthermore, the convexity of the loss function can lead to expectable error bound for the testing data in the target domain for domain adaptation problems, which will be discussed in detail in Section 3.4. Note that the decision function both in the kernel case and the linear case can be defined as in Equation (3.9), then the empirical loss on the training dataset can be expressed as:

$$\sum_{l=1}^{m} \sum_{i=1}^{n_1} L(w_l^\top \vec{x}_i, Y_{il}) \tag{3.11}$$

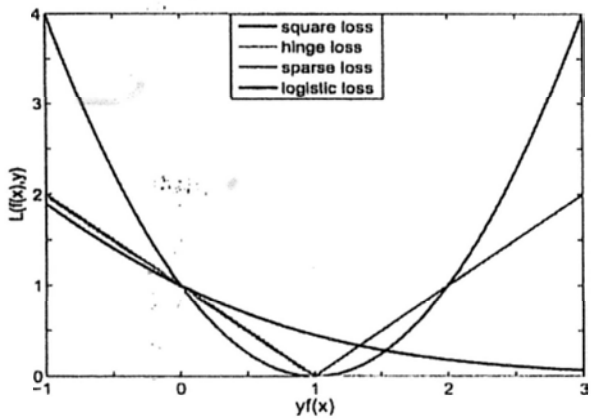

Figure 3.1: Demonstration of the four loss functions. x-axis is $yf(x)$, and y-axis is the loss function value $L(f(x), y)$.

3.3.1.3. Design of Regularization

Based on the parametric form in Equation (3.10) of the decision function f_l as in Equation (3.9), we introduce the following regularizer:

$$\Omega(f_l) = \|u_l\|^2 = \left\|w_l - \Theta^\top v_l\right\|^2, \tag{3.12}$$

which controls the complexity of each classifier independently. Besides, we treat the learning of the classifier for each class with equal importance. Therefore we set the coefficient of the regularization for each class as 1. Then the second term in (3.1) can be expressed as:

$$\sum_{l=1}^{m} \Omega(f_l) = \|u\|^2 = \left\|w - \Theta^\top v\right\|_F^2. \tag{3.13}$$

where $w = [w_1, \cdots, w_m]$, $u = [u_1, \cdots, u_m]$, and $v = [v_1, \cdots, v_m]$

3.3.1.4. Distribution Gap between Domains

Recall that the last term in Equation (3.1) measures the mismatch between the embedding of the source and target domain. We employ the Maximum Mean Discrepancy criterion [32] (MMD) as the nonparametric measure for comparing the distributions mismatch based on the Reproducing Kernel Hilbert Space

(RKHS) distance. Let the kernel-induced feature map be $\phi : \mathbb{R} \mapsto \mathcal{H}$, where $\mathcal{H}$ is the corresponding feature space. The MMD between the source domain D_S and the target domain D_T is defined as follows:

$$\begin{aligned} \mathrm{MMD}[D_S, D_T] &= \sup_{\|\varphi\|_{\mathcal{H}} \leq 1} \left(\mathrm{E}_Q[\varphi(x')] - \mathrm{E}_P[\varphi(x)]\right) \\ &= \|\mathrm{E}_Q[\phi(x')] - \mathrm{E}_P[\phi(x)]\|_{\mathcal{H}}. \end{aligned} \tag{3.14}$$

where x and x' represent the samples in D_S and D_T respectively, and φ is a function restricted in a unit ball in the RKHS $\mathcal{H}$ where $\varphi(x) = \langle \varphi, \phi(x) \rangle_{\mathcal{H}}$. The empirical measure of the MMD in Equation (3.14) is defined as:

$$\mathrm{MMD}[D_S, D_T] = \left\| \frac{1}{n_2} \sum_{x' \in D_T} \phi(x') - \frac{1}{n_1} \sum_{x \in D_S} \phi(x) \right\|_{\mathcal{H}} \tag{3.15}$$

Therefore, the distance between two distributions of two samples is simply the distance between the two mean elements in the RKHS.

Denote the gram matrix $K_{ij} = K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$ defined on all the data:

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}, \tag{3.16}$$

where $K_{S,S}$, $K_{T,T}$ and $K_{S,T}$ are the Gram matrices defined on the source domain, target domain, and cross domain data, respectively. Then the square of the MMD in (3.15) can be written as

$$\mathrm{tr}(KD), \tag{3.17}$$

where

$$D_{ij} = \begin{cases} \frac{1}{n_1^2} & \text{when } x_i, x_j \in D_S \\ \frac{1}{n_2^2} & \text{when } x_i, x_j \in D_T \\ \frac{-1}{n_1 n_2} & \text{otherwise.} \end{cases} \tag{3.18}$$

It has been proved in [10] that the empirical MMD in Equation (3.17) without the coefficients is an unbiased estimation of the squared MMD in Equation (3.14). Moreover, it can rapidly converge to the squared MMD when increasing the number of samples in both domains. In the following, we will investigate the feature map ϕ in both linear and kernel forms:

• **Linear case.** If $\phi(x) = \Theta x$, then $K = X^\top \Theta^\top \Theta X$, where $X = [x_1, \cdots, x_{n_1+n_2}]$.
• **Kernel case.** If $\phi(x) = \Theta \Psi_e(x) = \Theta \mathbf{K}(\cdot, x)$ where $\Psi_e(x)$ is the empirical kernel map defined in Equation (3.4), then $K = \mathbf{K}^\top \Theta^\top \Theta \mathbf{K}$

Recall that the definition of $\vec{x}$ in Equation (3.8), and denote $\vec{X}_\mathcal{T} = [\vec{x}_{n_1+1}, \cdots, \vec{x}_{n_1+n_2}]$ and $\vec{X} = [\vec{X}_\mathcal{S}, \vec{X}_\mathcal{T}]$, as the data matrices defined on the target domain and all input data, respectively. Then we can reformulate the gram matrix under both linear map and kernel map as $K = \vec{X}^\top \Theta^\top \Theta \vec{X}$. Then, the domain discrepancy criterion defined in Equation (3.17) becomes:

$$\begin{aligned} \mathrm{MMD}^2[D_\mathcal{S}, D_\mathcal{T}] &= \mathrm{tr}(\vec{X}^\top \Theta^\top \Theta \vec{X} D) \\ &= \mathrm{tr}(\Theta \bar{\mathbf{K}} \Theta^\top). \end{aligned} \tag{3.19}$$

where

$$\bar{\mathbf{K}} = \begin{cases} XDX^\top, & \text{(linear)} \\ \mathbf{K}D\mathbf{K}, & \text{(kernel)} \end{cases} \tag{3.20}$$

3.3.1.5. Final Formulation

Combining all the reformulations of the three items in Problem (3.1) as in Equation (3.11) (3.13) and (3.19), we arrive at the following minimization problem:

$$\begin{aligned} &\min_{\Theta, w, v} \sum_{l=1}^{m} \Big(\sum_{i=1}^{n_1} L(w_l^\top \vec{x}_i, Y_{il}) + \alpha \left\| w_l - \Theta^\top v_l \right\|^2 \Big) + \beta \mathrm{tr}(\Theta \bar{\mathbf{K}} \Theta^\top) \\ &\text{s.t.} \quad \Theta \Theta^\top = I_{r \times r}, \end{aligned} \tag{3.21}$$

which learns both the shared subspace Θ, and the parameters u and v in decision functions simultaneously.

3.3.2. Detailed Description of the Algorithm

In this section, we present the detailed optimization strategy for solving Problem (5.9). We can apply the following alternating optimization strategy to solve the optimal solution iteratively:

- Solving optimal w^* in Problem (5.9) with fixed (Θ, v).
- Solving optimal (Θ^*, v^*) in Problem (5.9) with fixed w.

In fact, there are some other possible alternating strategies, for example, optimizing (w, v) with fixed Θ first and then optimizing Θ with fixed (w, v).

3.3.2.1. Computing w^*

In the alternating optimization strategy mentioned above and given the convex loss function L discussed in Section 3.3.1.2, the first step of computing optimal w^* becomes a bound of convex optimization problems:

$$\{w_l^*\} = \arg\min_{w_l} \sum_{i=1}^{n_l} L(w_l^\top \vec{x}_i, Y_{il}) + \alpha \left\| w_l - \Theta^\top v_l \right\|^2 \tag{3.22}$$

where (Θ, v) are fixed. Problem (3.22) can be solved by many existing well-established methods, such as conjugate gradient (CG), stochastic gradient descending (SGD) [71] and so on. Furthermore, when we use the least square loss function, we can get the analytical solution for Problem (3.22) and greatly reduce the computation complexity without any gradient iterations [16].

3.3.2.2. Stochastic Gradient Descent

Stochastic gradient descent is an optimization method for minimizing an objective function that is written as a sum of differentiable functions.

$$L(w) = \sum_{i=1}^{n} L_i(w) + L_0(w) \tag{3.23}$$

where the parameter w is to be estimated and typically each summand function $L_i()|_{i=1}^n$ is associated with the i-th observation in the training data set, and $L_0(w)$ is a general differential function not specific for any training samples.

In classical statistics, sum-minimization problems arise in least squares of maximum-likelihood estimation for independent observations. It has been long recognized that local minimization is still too difficult to obtain for some problems of maximum-likelihood estimation. Therefore, contemporary statistical theorists usually only consider the stationary points of the likelihood function. In

statistical learning theory, there is a fundamental problem called empirical risk minimization where $L_i(w)|_{i=1}^n$ is the value of loss function at the i-th example, and $L(w)$ is the empirical risk. When we minimize the above function, a standard gradient descent method would perform the following iterations:

$$w := w - \tau \nabla L(w) = w - \tau \sum_{i=1}^{n} \nabla L_i(w) - \tau \nabla L_0(w) \tag{3.24}$$

where τ is a step size.

However, in other cases, evaluating the sum-gradient may require expensive evaluations of the gradients from all the functional values on the samples when the training set is enormous and there exists no simple formulas. To reduce the computational cost at every iteration, the true gradient of $L(w)$ is approximated by a gradient at a single example:

$$w := w - \tau \nabla L_i(w) - \tau \nabla L_0(w) \tag{3.25}$$

As the algorithm scans through the training set, it performs the above update for each training example. Several passes over the training set are made until the algorithm converges. A typical implementations is to randomly sample training examples at each pass and use an adaptive step size.

The convergence of stochastic gradient descent has been analyzed using the theories of convex minimization and of stochastic approximation. Briefly, stochastic gradient methods will not converge to a global minimum unless the objective function is convex. The second order stochastic gradient descent method corresponding to Equation 3.25 should be:

$$w := w - \tau \Phi \nabla L_i(w) - \tau \Phi_0 \nabla L_0(w) \tag{3.26}$$

where

$$\begin{aligned} &\Phi \approx H^{-1}(w), \quad H(w) = \nabla \nabla L_i(w) \quad \text{for } i = 1, \cdots, n \\ &\Phi_0 \approx H_0^{-1}(w), \quad H_0(w) = \nabla \nabla L_0(w) \end{aligned} \tag{3.27}$$

Back to our problem defined in Equation 3.22, we will update the gradient in each step as follows:

$$w_{l,t+1} := w_{l,t} - \tau \nabla L_i(w_{l,t}^\top x_t, Y_{tl}) x_i + \tau \Theta_l^\top v_l \tag{3.28}$$

Input: Data samples $\{(x_i, y_i)\}_{i=1}^{n}$ in source domain D_S, Θ_l, v_l, convergence rate upper-bound ϵ

Output: General classifier w_l for the l-th class

Initialize Choosing an initial vector $w_{l,0}$;

set t = 0.

repeat

1 Randomly sampling a training sample as $\{(x_t, y_t)\}$

2 compute $w_{l,t+1} := w_{l,t} - \tau \nabla L_i(w_{l,t}^{\top} x_t, Y_{tl}) x_i + \tau \Theta_l^{\top} v_l$

3 set t := t+1

4 compute $\tau := 1/\mathrm{t}$

until $\|w_{l,t+1} - w_{l,t}\| \leq \epsilon$

Figure 3.2: The outline of the adapted stochastic gradient descent (SGD) algorithm used in our algorithm

It have been proved that when the best update rate τ is equal to $1/t$, the convergence rate of SGD grows linearly with t. Then in our algorithm, we set the update learning rate τ directly as $1/t$. The pesudocode of this SGD can be summarized as shown in Figure 3.2.

It also have been investigated that the time complexity for SGD is very efficient given by:

$$T \sim n\mathrm{loglog}n \tag{3.29}$$

The parameter w_T of the stochastic algorithm also has been proved that it converges to the local optimal. Comparing the accuracies of both algorithms, it shows that the stochastic algorithm asymptotically provides a better solution by a factor $(\log \log n)$.

$$E[(w_T - w^*)^2] \sim \frac{1}{n \log \log n} \ll \frac{1}{n} \sim E[(w_n - w^*)^2] \tag{3.30}$$

where w_n represents the iterative value obtained by the gradient methods running on all n training samples. It can be seen that SGD can greatly economize the optimizing process by only involving one data sample, at the same time, it can

converge to the local optimum quickly, such that it can be widely applied in large scale text mining problems.

3.3.2.3. Computing (Θ^*, v^*)

It is easy to see that Problem (5.9) with fixed w^* is equivalent to the following optimization problem:

$$\begin{aligned} \min_{\Theta,v} & \sum_{l=1}^{m} \alpha \left\| w_l^* - \Theta^\top v_l \right\|^2 + \beta \mathrm{tr}(\Theta \bar{\mathbf{K}} \Theta^\top) \\ \text{s.t.} \quad & \Theta\Theta^\top = I_{r\times r}, \end{aligned} \tag{3.31}$$

with fixed Θ, we can optimize v_l for different l individually, and we have

$$\left\| w_l^* - \Theta^\top v_l \right\|^2 \geq \|w_l^*\|^2 - \|\Theta w_l^*\|^2 \tag{3.32}$$

and $v_l^* = \Theta w_l^*$ holds the equality. Then we can substitute the optimal v^* back to Problem (3.31) and using the equality in Equation (3.32) to get the following optimization problem:

$$\min_{\Theta} - \sum_{l=1}^{m} \alpha \|\Theta w_l^*\|^2 + \beta \mathrm{tr}(\Theta \bar{\mathbf{K}} \Theta^\top), \qquad \text{s.t. } \Theta\Theta^\top = I, \tag{3.33}$$

Using simple linear algebra, we know that

$$\sum_{l=1}^{m} \|\Theta w_l^*\|^2 = \|\Theta w\|_F^2 = \mathrm{tr}(\Theta w^* w^{*\top} \Theta^\top) \tag{3.34}$$

Then with fixed w^*, we can get the optimal Θ^* by solving

$$\min_{\Theta} \mathrm{tr}(\Theta(\beta \bar{\mathbf{K}} - \alpha w^* w^{*\top})\Theta^\top), \text{ s.t. } \Theta\Theta^\top = I \tag{3.35}$$

It is well known that the optimal solution of Problem (3.35) is given by the eigen-decomposition of $\beta \bar{\mathbf{K}} - \alpha w^* w^{*\top}$. Suppose

$$\beta \bar{\mathbf{K}} - \alpha w^* w^{*\top} = H \Lambda H^\top \tag{3.36}$$

be the eigen-decomposition of $\beta \bar{\mathbf{K}} - y\alpha w^* w^{*\top}$. If we arrange the diagonal elements of the diagonal matrix Λ in ascending order, then the rows of Θ^* are given by the first r rows of H, which corresponds to the smallest r eigenvalues of $\beta \bar{\mathbf{K}} - \alpha w^* w^{*\top}$.

3.3.2.4. Overall Algorithm

Combining all the derivations in Section 3.3.2.1 and Section 3.3.2.3 corresponding to the optimization strategy, we develop our Low-Rank Shared Concept (LRSC) domain adaptation algorithm as depicted in Figure 3.3.

3.3.3. Prediction in Operational Setting

After extracting the shared subspace Θ, and the weight vectors u_l and v_l for each class, one can perform prediction using Equation (3.9). However, the weight vector u_l is learned to minimize the empirical loss of the labeled data in the source domain D_S, and may not be the discriminative direction for the testing data in the target domain D_T.

Recall that the subspace Θ is learned to minimize the MMD criterion in Equation (3.19), and capture the intrinsic structure of data for domain adaptation. Moreover, the weight vector v_l is the discriminative direction defined on the projected subspace Φ, so the prediction on the data x for the l-th class in the target domain D_T can be performed by a decision classifier

$$f_l(x) = v_l^\top \Theta \vec{x} = \begin{cases} v_l^\top \Theta x, & \text{(linear)} \\ v_l^\top \Theta \mathbf{K}(\cdot, x), & \text{(kernel)} \end{cases} \tag{3.37}$$

instead of $f_l(x)$ in Equation (3.9), and $\Theta^\top v_l$ is the discriminative direction for the l-th class in the target domain.

Figure 3.4 depicts the intrinsic mechanism for prediction where the green (lighter) circles and red (darker) circles represent the samples in the source domain D_S and the target domain D_T respectively. Both of the two domains contain two classes. By using other low-rank shared space extraction, we can discover the shared space which is shown as the vertical space labeled by Θx. The learned classifier in the source domain is $u^\top \vec{x} + v^\top \Theta \vec{x}$ represented by the green dashed line (the dashed line on the left), which can be decomposed to two components, $u^\top \vec{x}$ and $v^\top \Theta \vec{x}$. However, it can be seen the shared space is

Input: Data samples $\{(x_i, y_i)\}_{i=1}^{n}$ in source domain D_S;

Data samples $\{(x_i)\}_{i=n_1+1}^{n_1+n_2}$ in target domain D_T;

Linear or nonlinear feature map Φ.

Parameters: r and trade-off coefficients (α, β).

Output: Optimal concept subspace projection Θ;

the corresponding adaptive classifiers $\{v_l\}_{l=1}^{m}$ in the embedded space;

the general classifiers $\{w_l\}_{l=1}^{m}$

Initialize Constructing $\vec{x}_i$ as in Equation (3.8) and $\bar{K}$ as in Equation (3.20), set $\Theta = I$, $v_l = 0$, for $l = 1, \cdots, m$.

repeat

1 for $l = 1$ to m do

with fixed $(\Theta, \{v_l\})$, solve the optimization problem:

$$\{w_l^*\} = \arg\min_{w_l} \sum_{i=1}^{n_1} L(w_l^\top \vec{x}_i, Y_{il}) + \alpha \left\| w_l - \Theta^\top v_l \right\|^2$$

using the stochastic gradient descent method depicted in Figure 3.2

end

2 Do the eigen-decomposition $\beta\bar{\mathbf{K}} - y\alpha w^* w^{*\top} = H\Lambda H^\top$(with the diagonals of Λ in ascending order), and let the rows of Θ^* be the first r rows of H.

3 Compute $v_l = y\Theta w_l$, $l = 1, \cdots, m$.

until convergence

Figure 3.3: The outline of our Low-Rank Shared Concepts (LRSC) domain adaptation algorithm

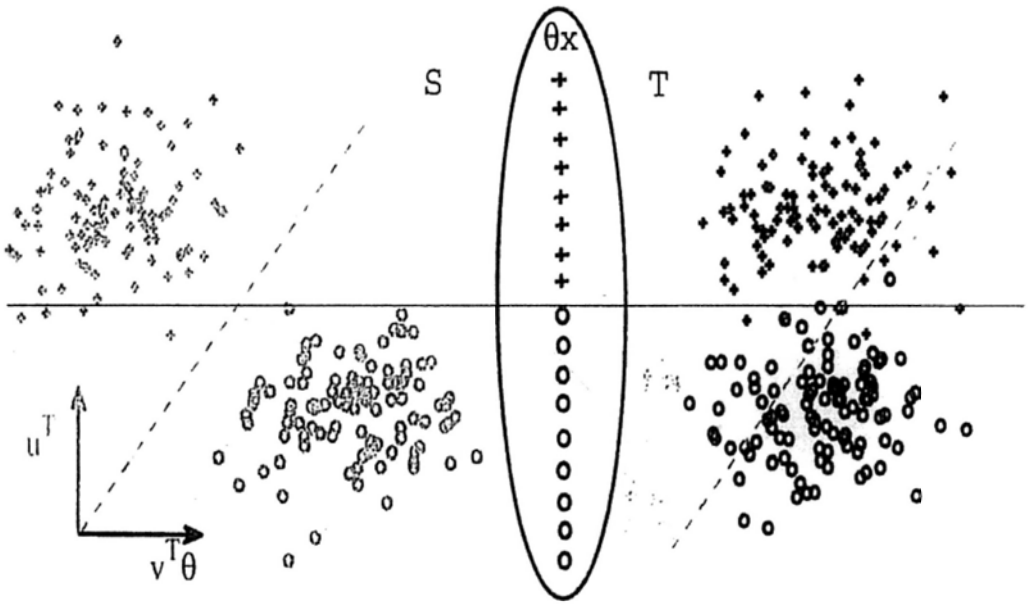

Figure 3.4: Demonstration of the prediction in operatioal setting. The green dashed line on the left represent the classifier trained in D_S, and its horizon decomposing component plotted as red line will be used as the classifier in D_T. The middle Ellipse represents the shared space between D_S and D_T

$\Theta\vec{x}$, which means the other component $u^\top\vec{x}$ is useless for the target domain D_T. Therefore, our final prediction function for the target domain is $v^\top\Theta\vec{x}$.

3.4. Error Analysis on Adaptation Model

We investigate the error analysis of our proposed domain adaptation method in the target domain D_T both in the linear and the nonlinear case. First, we denote the prediction function in D_T for the l-th class as follows:

$$f_{Tl}(x) = \begin{cases} v_l^T\Theta\vec{x} & \text{if } -1 \le v_l^T\Theta\vec{x} \le 1, \\ 1 & \text{if } 1 < v_l^T\Theta\vec{x}, \\ -1 & \text{if } v_l^T\Theta\vec{x} < -1, \end{cases} \tag{3.38}$$

where $\vec{x}$ is defined in Equation (3.8) and $f_{Tl}(x)$ can take fractional value when x is not predicted deterministically in the l-th class. Then we denote the truth

labeling function $h_l(x) : \mathcal{X} \to \{-1, 1\}$. Let $L(x)$ be a continuous loss function defined in Section 3.3.1.2. Then the expected loss of $f_{\mathcal{T}l}$ in $D_{\mathcal{T}}$ is defined as:

$$\epsilon_{\mathcal{T}}(h_l, f_{\mathcal{T}l}) = \mathrm{E}_{x\sim D_{\mathcal{T}}}[L(h_l(x), f_{\mathcal{T}l}(x))]$$

Note that $f_{\mathcal{S}l}(x) = u_l^{\top}\vec{x} + v_l^{\top}\Theta\vec{x}$ is the proposed decision function in Equation (3.2) for the labeled data in the source domain $D_{\mathcal{S}}$, then we also define the expected loss of $f_{\mathcal{S}l}$ in $D_{\mathcal{S}}$ as:

$$\epsilon_{\mathcal{S}}(h_l, f_{\mathcal{S}l}) = \mathrm{E}_{x\sim D_{\mathcal{S}}}[L(h_l(x), u_l^{\top}\vec{x} + v_l^{\top}\Theta\vec{x})].$$

For simplicity, we denote $\mathrm{E}_{x\sim D_{\mathcal{S}}} = \mathrm{E}_{\mathcal{P}}$, $\mathrm{E}_{x\sim D_{\mathcal{T}}} = \mathrm{E}_{\mathcal{Q}}$, $\epsilon_{\mathcal{T}}(h_l, f_{\mathcal{T}l}) = \epsilon_{\mathcal{T}}(h_l)$ and $\epsilon_{\mathcal{S}}(h_l, f_{\mathcal{S}l}) = \epsilon_{\mathcal{S}}(h_l)$. Based on the definition of $f_{\mathcal{T}l}(x)$ in Equation (3.38), we know that $0 \leq L(h_l(x), f_{\mathcal{T}l}(x)) \leq 1$. With a mild assumption that $\|L(h_l(x), f_{\mathcal{T}l}(x)\|_{\mathcal{H}}$ is bounded by a finite number γ, where $\mathcal{H}$ is a RKHS, we obtain the following theorem:

Theorem 1. Suppose $\|\vec{x}\| = 1$, the expected loss of $f_{\mathcal{T}l}$ in $D_{\mathcal{T}}$ is bounded by

$$\epsilon_{\mathcal{T}}(h_l) \leq \epsilon_{\mathcal{S}}(h_l) + \gamma \mathrm{MMD}[D_{\mathcal{S}}, D_{\mathcal{T}}] + \delta\|u_l\|^2 + \delta \tag{3.39}$$

Proof. First we investigate the property of the loss function $L(h_l(x), f_{\mathcal{T}l}(x))$. We can denote $g(h_l(x)f_{\mathcal{T}l}(x)) = L(h_l(x), f_{\mathcal{T}l}(x))$, where $g(t)$ is convex continuous function.

- **Sparse loss**: $L(h_l(x), f_{\mathcal{T}l}(x)) = |h_l(x) - f_{\mathcal{T}l}(x)| \Rightarrow g(t) = |1 - t| \Rightarrow g'(t) = 1 \text{or} -1 \Rightarrow |g'(t)| = 1$.
- **Square loss**: $L(h_l(x), f_{\mathcal{T}l}(x)) = (h_l(x) - f_{\mathcal{T}l}(x))^2 \Rightarrow g(t) = (1 - t)^2 \Rightarrow g'(t) = 2(t - 1) \Rightarrow |g'(t)| \leq 4$.
- **Hinge loss**: $L(h_l(x), f_{\mathcal{T}l}(x)) = \max(0, 1 - h_l(x)f_{\mathcal{T}l}(x)) \Rightarrow g(t) = \max(0, 1 - t) \Rightarrow g'(t) = 0 \text{ or } -1 \Rightarrow |g'(t)| \leq 1$.
- **Logistic loss**: $L(h_l(x), f_{\mathcal{T}l}(x)) = \log_2(1+e^{-h_l(x)f_{\mathcal{T}l}(x)}) \Rightarrow g(t) = \log_2(1+e^{-t}) \Rightarrow g'(t) = -\frac{-e^{-t}}{1+e^{-t}} \Rightarrow |g'(t)| \leq 1$.

We can observe that for all the above common convex loss functions, the first

order derivative satisfy $|g'(t)| \leq 4\delta$ where δ is a positive scalar selected depending on the loss function. Then we have

$$\begin{aligned}
& g(x_0 + \xi) \geq g(x_0) + g'(x_0)\xi \\
\Rightarrow & g(h_l(x)(u_l^\top \vec{x} y + v_l^\top \Theta \vec{x})) \\
& \geq g(h_l(x) v_l^\top \Theta \vec{x}) + \frac{1}{2} g'(h_l(x) v_l^\top \Theta \vec{x}) h_l(x) u_l^\top \vec{x} \\
\Rightarrow & -g(h_l(x)(u_l^\top \vec{x} + v_l^\top \Theta \vec{x})) \\
& \leq -g(h_l(x) v_l^\top \Theta \vec{x}) + \frac{1}{2} |g'(h_l(x) v_l^\top \Theta \vec{x}) u_l^\top \vec{x}| \\
\Rightarrow & -L(h_l(x), u_l^\top \vec{x} + v_l^\top \Theta \vec{x}) \leq -L(h_l(x), v_l^\top \Theta \vec{x}) + 2\delta |u_l^\top \vec{x}|
\end{aligned}$$

Then recall the definitions and abbreviations above, we have

$$\begin{aligned}
& \epsilon_{\mathcal{T}}(h_l) \\
= & \epsilon_{\mathcal{S}}(h_l) + \epsilon_{\mathcal{T}}(h_l) - \epsilon_{\mathcal{S}}(h_l) \\
= & \epsilon_{\mathcal{S}}(h_l) + \epsilon_{\mathcal{T}}(h_l) - \mathrm{E}_{\mathcal{P}}[L(h_l(x), u_l^\top \vec{x} + v_l^\top \Theta \vec{x})] \\
\leq & \epsilon_{\mathcal{S}}(h_l) + \epsilon_{\mathcal{T}}(h_l) - \mathrm{E}_{\mathcal{P}}[L(h_l(x), v_l^\top \Theta \vec{x})] + 2\delta \mathrm{E}_{\mathcal{P}}[|u_l^\top \vec{x}|] \\
\leq & \epsilon_{\mathcal{S}}(h_l) + \mathrm{E}_{\mathcal{Q}}[L(h_l(x), f_{\mathcal{T}l}(x))] - \mathrm{E}_{\mathcal{P}}[L(h_l(x), f_{\mathcal{T}l}(x))] \\
& + 2\delta \mathrm{E}_{\mathcal{P}}[|u_l^\top \vec{x}|]
\end{aligned} \tag{3.40}$$

The first inequality is due to the convexity property of the loss function, and the last inequality holds due to the definition of $f_{\mathcal{T}l}(x)$ in Equation (3.38) and

$$|h_l(x) - f_{\mathcal{T}l}(x)| \leq |h_l(x) - v_l^\top \Theta \vec{x}|.$$

Moreover, using the Cauchy-Schwarz inequality, we have:

$$2\mathrm{E}_{\mathcal{P}}[|u_l^\top \vec{x}|] \leq \mathrm{E}_{\mathcal{P}}[\|u_l\|^2 + \|\vec{x}\|^2] = \|u_l\|^2 + \mathrm{E}_{\mathcal{P}}[\|\vec{x}\|^2].$$

Since $\|\vec{x}\| = 1$, so that

$$\mathrm{E}_{\mathcal{P}}(|u_l^\top \vec{x}|) \leq \|u_l\|^2 + 1. \tag{3.41}$$

By the virtual of RKHS property, for any function $L(x)$ in this RKHS, it can be expressed as $L(x) = \langle L, \phi(x) \rangle_{\mathcal{H}}$. Then, we can obtain the following bound:

$$\begin{aligned} & \mathrm{E}_{\mathcal{Q}}[L(h_l(x), f_{\mathcal{T}l}(x))] - \mathrm{E}_{\mathcal{P}}[L(h_l(x), f_{\mathcal{T}l}(x))] \\ = \; & \mathrm{E}_{\mathcal{Q}}[\langle \phi(x), L\rangle_{\mathcal{H}}] - \mathrm{E}_{\mathcal{P}}[\langle \phi(x), L\rangle_{\mathcal{H}}] \\ = \; & \langle \mathrm{E}_{\mathcal{Q}}[\phi(x)] - \mathrm{E}_{\mathcal{P}}[\phi(x)], L\rangle_{\mathcal{H}}. \end{aligned}$$

Assume $\|L\|_{\mathcal{H}} \leq \gamma$, similar to Equation (3.14), we have:

$$\begin{aligned} \langle \mathrm{E}_{\mathcal{Q}}[\phi(x)] - \mathrm{E}_{\mathcal{P}}[\phi(x)], L\rangle_{\mathcal{H}} &\leq \gamma \left\| \mathrm{E}_{\mathcal{Q}}[\phi(x)] - \mathrm{E}_{\mathcal{P}}[\phi(x)] \right\|_{\mathcal{H}} \\ &= \gamma \mathrm{MMD}[D_{\mathcal{S}}, D_{\mathcal{T}}]. \end{aligned} \tag{3.42}$$

Substitute Equation (3.41) and Equation (3.42) into Equation (3.40), this completes the proof. □

According to the expected error bound in Equation (3.39), we can calculate the total expected loss for all the m classes in the target domain $D_{\mathcal{T}}$

$$\begin{aligned} \epsilon_{\mathcal{T}} &= \sum_{l=1}^{m} \epsilon_{\mathcal{T}}(h_l) \\ &\leq \sum_{l=1}^{m} (\epsilon_{\mathcal{S}}(h_l) + \delta \|u_l\|^2 + \delta) + m\gamma \mathrm{MMD}[D_{\mathcal{S}}, D_{\mathcal{T}}] \\ &\doteq \sum_{l=1}^{m} (\sum_{i=1}^{n_1} L(f_l(x_i), Y_{il}) + \delta \|u_l\|^2) + m\gamma (\mathrm{tr}(\Theta \bar{\mathbf{K}} \Theta^{\top})^{\frac{1}{2}}) \\ &+ m\delta \end{aligned} \tag{3.43}$$

where the last approximation is due to replacing the expected loss in $D_{\mathcal{S}}$ by the empirical loss. We can see the total expected loss in the target domain $D_{\mathcal{T}}$, is bounded by the combinations of the total empirical loss in the source domain $D_{\mathcal{S}}$, the maximum mean discrepancy value which evaluates the distribution difference between $D_{\mathcal{T}}$ and $D_{\mathcal{S}}$, and the regularization for all the classifiers in $D_{\mathcal{S}}$. Therefore, we can conclude that minimizing the objective function in Problem (5.9) can also minimize the expected loss in the target domain $D_{\mathcal{T}}$. Moreover, the error bound established in Equation (3.43) can supply advices for setting the trade-off coefficients α and β by estimating δ and γ. For example, if we employ the hinge loss or sparse loss as our empirical loss function, we can directly set

$\alpha = \delta = 0.25$. While if we use least square loss function, we will set $\alpha = \delta = 1$. We can also set $\beta = mr/\text{MMD}[D_S, D_T]$, which is depend on the class number and employed kernels.

3.5. Discriminative Feature Propagation

One major problem in text mining is the sparsity of features in the high dimensional space. Specifically, some discriminative features occur frequently in the target domain D_T but seldom appear or even are absent in the source domain D_S. For example, for the task of extracting sentences corresponding to job requirements from job Web sites, some common terms between the healthcare industry and the accounting industry may be "qualified", "year", "experience" and so on as shown in Figure 3.5. However, some characteristic words are dependent of the job nature. For instance, "CPA", "CA", "ACCA" are discriminative terms for the "accounting" domain whereas "CCP", "physiology" are discriminative terms for the domain of "health care". To address this issue, we develop the following feature propagation strategy.

According to the discussion in Section 3.2, we can extract a common feature set $\mathcal{F}$ from both domains for each specific task l by selecting the features which have high weight in $\Theta' v_l$ and also have close empirical mean between the source domain and the target domain. Based on the co-occurrence information in the target domain, we can compute the similarity between the common features in the set $\mathcal{F}$ and the remaining features (non-common features) in another set $\bar{\mathcal{F}}$. For each non-common feature, we can sum up its similarity with all the common features. Finally we rank all the non-common features by its similarity with the common feature set in descending order. By selecting the top K highly similar non-common terms, and combining with all the existing common features, we can get a set of characteristic features $\mathcal{F}_c \subset \mathcal{F} \cup \bar{\mathcal{F}}$ for the target domain.

Based on the assumption that similar features should have similar prediction power in the target domain, we can construct a feature similarity graph $\mathcal{G}$. In $\mathcal{G}$,

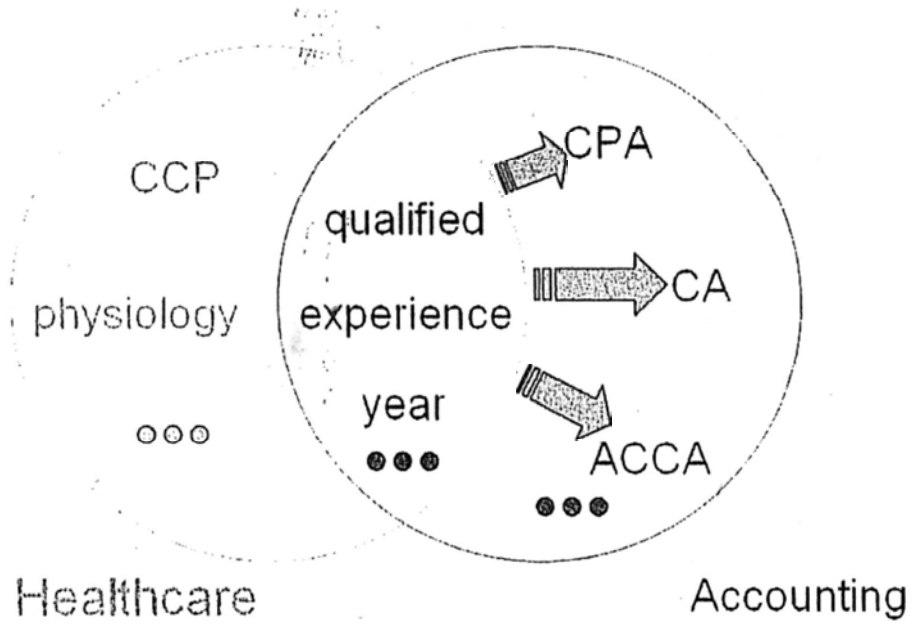

Figure 3.5: Demonstration of the feature propagation from the shared common features between domain healthcare and domain accounting to discriminative features in domain accounting

each vertex v represents a feature, and edge weights are given by a symmetric matrix $E \in \mathbb{R}^{d\times d}$, whose entries $E_{uv} = \langle \pi_u, \pi_v \rangle \geq 0$, where $\langle \cdot, \cdot \rangle$ means the inner product, π_i represents the vector of normalized occurrence in the target domain. Define the degree of vertex v as $d_v = \sum_{u\sim v} E_{uv}$, then we can define the normalized graph Laplacian matrix:

$$\mathcal{L}_{uv} = \begin{cases} 1 - E_{uv}/d_u & \text{if } u = v \text{ and } d_u \neq 0 \\ -E_{uv}/\sqrt{d_u d_v} & \text{if } u \text{ and } v \text{ are } adjacent \\ 0 & \text{otherwise.} \end{cases} \tag{3.44}$$

We also define a column vector $\rho = [\rho_1, \ldots, \rho_d]^\top \in \mathbb{R}^d$ representing the discriminative weight vector of characteristic features. Intuitively, similar features should have similar weights. Therefore, we introduce a manifold regularizer using the feature graph Laplacian matrix in (3.44) as:

$$\rho^T \mathcal{L} \rho = \sum_{u,v} E_{uv} \left(\frac{\rho_u}{\sqrt{d_u}} - \frac{\rho_v}{\sqrt{d_v}} \right)^2,$$

which propagates the weight of the common features to other characteristic features via the manifold structure of the feature graph.

Moreover, we also require the discriminative weight vector ρ be close to the discriminative direction learned for each class in the target domain. Thus, we arrive at the following optimization problem:

$$\min \quad \|\rho_l - \Theta^\top v_l\|^2 + \gamma \rho_l^T \mathcal{L}_l \rho_l, \tag{3.45}$$

where the first term minimizes the difference between ρ_l and $\Theta^\top v_l$, and the second term enforces that the assignment of the weight of the characteristic features is propagated from the common features. In addition, the optimization problem (3.45) can be solved according to the following lemma:

Lemma 3.1. Let v_l be the classifier for the class l on the shared feature subspace Θ, therefore the corresponding optimal ρ_l has a closed form in term of Θ and v_l.

Proof. We first rewrite the objective function as follows:

$$\begin{aligned} & \|\rho_l - \Theta^\top v_l\|^2 + \gamma \rho_l^\top \mathcal{L}_l \rho_l \\ = \; & \rho_l^\top \rho_l - 2 v_l^\top \Theta \rho_l + v_l^\top v_l + \gamma \rho_l^\top \mathcal{L}_l \rho_l \\ = \; & \rho_l^\top (I + \gamma \mathcal{L}_l) \rho_l - 2 v_l^\top \Theta \rho_l + v_l^\top v_l \end{aligned} \tag{3.46}$$

Setting the derivation of (3.46) with respect to ρ_l to zeros, we have:

$$\rho_l = (I + \gamma \mathcal{L}_l)^{-1} \Theta^\top v_l.$$

This completes the proof. □

Therefore, the prediction on the testing patterns in the target domain can be performed by:

$$f_{Tl}(x') = v_l^\top \Theta (I + \gamma \mathcal{L}_l)^{-1} x'.$$

However, computing the matrix inversion $(I + \gamma\mathcal{L}_l)^{-1}$ is still computational intensive (with complexity $O(d^3)$). Note that when the predefined parameter γ satisfies $0 < \gamma < 1$, we have the following Taylor expansion:

$$(I + \gamma\mathcal{L}_l)^{-1} = I - \gamma\mathcal{L}_l + \gamma^2\mathcal{L}_l^2 - \gamma^3\mathcal{L}_l^3 + ...$$

As $\mathcal{L}_l$ is usually very sparse, especially when γ is small, one can approximate $(I + \gamma\mathcal{L}_l)^{-1}$ as $I - \gamma\mathcal{L}_l$ and the revised discriminative direction is:

$$\rho_l = \Theta^\top v_l - \gamma\mathcal{L}_l\Theta^\top v_l,$$

Then the decision function on the testing patterns in the target domain becomes:

$$f_{Tl}(x') = v_l^\top\Theta(I - \gamma\mathcal{L}_l)x',$$

As a result, the computation of the prediction is much reduced.

As discussed above, $\Theta^\top v_l$ is the optimal discriminative direction of the l-th class in (5.9). From the propagation of the feature graph $\mathcal{G}$, the discriminative information from other characteristic features $\mathcal{F}_c$ can be used to compute the weight vector $-\gamma\mathcal{L}_l\Theta^\top v_l$ to correct the discriminative direction.

3.6. Experiments

We demonstrate the effectiveness of our proposed domain adaptation method by conducting experiments on a number of data sets covering two common text mining problems, namely, document classification and information extraction.

3.6.1. Document Classification

3.6.1.1. Data Sets

20-Newsgroup. The first dataset is derived from the 20-Newsgroup corpus for document classification. The original 20-Newsgroup corpus contains more than 18,000 newsgroup articles collected from 20 different Usenets newsgroups. Table

3.1 depicts the newsgroups ID for identification in this thesis. We observe that the articles in some newsgroups are related to the same topic. For example, the newsgroups *rec.auto* and *rec.motorcycle* are related to the topic *car*; the newsgroups *rec.baseball* and *rec.hockey* are related to the topic *ball game*; the newsgroups *comp.sys.ibm.pc.hardware* and *comp.sys.mac.hardware* are related to the topic *hardware*; and the newsgroups *comp.windo-ws.x* and *comp.os.ms-windows.misc* are related to the topic *OS*, and so on. Therefore, the articles originated from the related newsgroups can be labeled by the same topic. However, there exists distribution shift from one newsgroup to another, even the two newsgroups are related. Table 3.2 shows the dataset derived and used in our experiments. There are six class labels, namely, *car*, *ball game*, *hardware*, *OS*, *religious*, and *politics*. For each class label, there are two related newsgroups as described above. Therefore, we can treat one newsgroup as the source domain and use the articles contained as the labeled data. The other related newsgroup can be considered as the target domain and the articles contained are regarded as unlabeled data. The first three datasets, namely, NG1, NG2 and NG3, depicted in the table consist of two class labels. For example, NG1 contains the class labels *car* and *ball game*. The newsgroups *rec.auto* and *rec.baseball* are treated as the source domain. 400 articles from each of these two newsgroups are randomly collected as the labeled examples, constituting a total of 800 labeled examples. The target domain is composed of 400 articles from each of the newsgroups *rec.motorcycles* and *rec.hockey*, constituting a total of 800 unlabeled data. The datasets, namely, NG{4-7}, contain 4 class labels in each dataset. The last two datasets, NG8 and NG9 contain 6 class labels. The composition of articles in each class label in the source and target domains is clearly shown in Table 3.2. Each article is represented by the vector space model and normalized to unit length.

Reuters-21578. The second dataset is the Reuters-21578 corpus for document classification. There are three top categories of documents, namely, *people*, *place*, and *organization*, in the corpus. We derive datasets used in our experiments by treating documents from one of these class labels as the source domain

Table 3.1: Newsgroups ID for identification in this thesis.

newsgroup ID	Newsgroup	newsgroup ID	Newsgroup
auto	rec.auto	motor	rec.motorcycle
baseball	rec.baseball	hockey	rec.hockey
ibm	comp.sys.ibm.pc.hardware	mac	comp.sys.mac.hardware
wsx	comp.windows.x	wsmisc	comp.os.ms-windows.misc
chrsrg	soc.religion.christian	miscrg	talk.religion.misc
mept	talk.politics.mideast	miscpt	talk.politics.misc

and labeled examples. Next, we treat the documents from another class as the unlabeled target domain. The three datasets derived and used in our experiments are denoted by People-Place, Org-Place, and Org-People respectively. For example, People-Place refers to the dataset treating documents from the class label People and documents from the class label Place as the source and target domain respectively. The detailed setting can be referred to [20]. Similar to the 20-Newsgroup dataset, each document is represented by the vector space model and normalized to unit length.

3.6.1.2. Comparison Algorithms

In order to verify the effectiveness of our method, we compare with two typical classification methods: Support Vector Machine (SVM), Transductive Support Vector Machine (TSVM), and three cross domain classification methods: Kernel Mean Matching (KMM) [37], Transfer Component Analysis (TCA) [49], and Cross Domain Spectral Clustering (CDSC) as presented in [44]. They represent supervised classification, semi-supervised classification, and recent domain adaptation methods respectively.

SVM and TSVM [41] are implemented by[1] SVM^{light}, which are the state-

[1]http://svmlight.joachims.org

Table 3.2: The data collected from 20-Newsgroup for document classification experiments.

Data set	Domain	class label						#doc
		car	ball game	hardware	OS	religion	politics	
NG1	source	auto	baseball	N/A	N/A	N/A	N/A	800
	target	motor	hockey	N/A	N/A	N/A	N/A	800
NG2	source	N/A	N/A	ibm	wsx	N/A	N/A	800
	target	N/A	N/A	mac	wsmisc	N/A	N/A	800
NG3	source	N/A	N/A	N/A	N/A	chrsrg	mept	800
	target	N/A	N/A	N/A	N/A	miscrg	miscpt	800
NG4	source	auto	baseball	ibm	wsx	N/A	N/A	1600
	target	motor	hockey	mac	wsmisc	N/A	N/A	1600
NG5	source	motor	hockey	mac	wsmisc	N/A	N/A	1600
	target	auto	baseball	ibm	wsx	N/A	N/A	1600
NG6	source	auto	baseball	N/A	N/A	chrsrg	mept	1600
	target	motor	hockey	N/A	N/A	miscrg	miscpt	1600
NG7	source	motor	hockey	N/A	N/A	miscrg	miscpt	1600
	target	auto	baseball	N/A	N/A	chrsrg	mept	1600
NG8	source	auto	baseball	ibm	wsx	chrsrg	mept	2400
	target	motor	hockey	mac	wsmisc	miscrg	miscpt	2400
NG9	source	motor	hockey	mac	wsmisc	miscrg	miscpt	2400
	target	auto	baseball	ibm	wsx	chrsrg	mept	2400

Table 3.3: The data collected from Reuters-21578 for document classification experiments.

Data Set	# of documents in D_S	# of documents in D_T
People-Place	1079	1080
Org-Place	1239	1210
Org-People	1016	1046

of-the-art classification methods and have been proved to be powerful in vast applications, especially in text mining fields. The parameters are almost all set as default in the package, and we use the RBF kernel for both two algorithms.

CDSC introduces the spectral clustering strategy to maintain the consistence in both the source domain and the target domain, where the objective function is minimizing the cut size on all the data with the least inconsistency of the data in D_S , and at the same time maximizing the separation of the data in the target domain D_T. Intuitively, the regularization is regarded as the balance between the in-domain supervision and the out-of-domain structure. The parameter setting is exactly the same as report in their paper [44].

KMM is proposed to reduce the mismatch between the two different domains D_S and D_T [37], which is a two-step approach. The first step is to diminish the difference of means of samples in RKHS between the two domains by re-weighting the samples $\phi(x_i)$ in the source domain as $\beta_i\phi(x_i)$, where β_i is learned by minimizing the MMD criterion in (3.15).

$$\mathrm{MMD}[D_S, D_T] = \left\| \frac{1}{n_1}\sum_{i=1}^{n_1}\beta_i\phi(x_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_i{}') \right\|_{\mathcal{H}}^2,$$

subjected to $\beta_i \in [0,1]$ and $|\sum_1^{n_1}\beta_i - n_1| \leq n_1\epsilon$, where ϵ is a small value to ensure that the corresponding measure $\beta_i(x)\mathcal{P}(x)$ is close to a probability distribution. Then the second step is to learn a decision classifier $f(x) = w^\top\phi(x) + b$ that separates patterns of opposite classes using the loss function re-weighted by β_i in the objective. The parameter ϵ is set as the same as reported in their paper.

TCA is implemented by us according to the algorithm as described in [49]:

$$\begin{aligned} \min_{W} \quad & \mathrm{tr}((W^\top KHKW)^+ W^\top (I + \mu KLK)W) \\ \text{s.t.} \quad & W^\top KHKW = I_{m\times m}, \end{aligned} \tag{3.47}$$

where the global kernel matrix K is defined as

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}, \tag{3.48}$$

where $K_{S,S}$, $K_{T,T}$ and $K_{S,T}$ are the Gram matrices defined on the source domain, target domain, and cross domain data, respectively. We can check that the solutions in Problem 3.47 is spanned by the eigenvectors corresponding to the m leading eigenvalues of $(I + \mu KLK)^{-1}KHK$. The parameter μ is set as chosen with the best performance in their report.

We experiment with RBF kernel for feature representation or instance re-weighting used by KMM and TCA, and adopt SVM for the final prediction. The kernel adopted in the SVM for the final prediction is a default linear kernel. For all those comparison algorithms, since they can only handle binary classification, we transform the multiclass problems to the 1-vs-rest problem setting for training [52]. For our proposed LRSC method, RBF kernel is used in our LRSC (kernel) with the kernel width as 0.1, and we set the parameters α and β according to the discussion in the error analysis in Section 3.4. Specifically, because we employ hinge loss, then we set $\alpha = 0.25$, $\beta = 0.1$ for kernel case, and $\beta = 1$ for linear case. The number of extracted concepts r is set to 30. For each data set, we repeated all the algorithms 10 times by randomly sampling the articles in each run and calculate the average performance, so as to decrease the sampling bias.

Table 3.4: The performance measured by F-measure of different sets of experiments in 20-Newsgroup dataset.

Data set	SVM	TSVM	KMM	CDSC	TCA	LRSC (linear)	LRSC (kernel)
NG1	0.849	0.874	0.909	0.899	0.932	0.945	**0.947**
NG2	0.716	0.732	0.779	0.782	0.806	**0.847**	0.841
NG3	0.694	0.754	0.653	0.770	0.694	0.824	**0.834**
NG4	0.659	0.670	0.747	0.726	0.761	0.780	**0.789**
NG5	0.645	0.675	0.672	0.681	0.707	**0.740**	0.736
NG6	0.566	0.644	0.620	0.607	0.610	0.667	**0.677**
NG7	0.555	0.694	0.679	0.673	0.710	**0.691**	0.683
NG8	0.538	0.531	0.563	0.624	0.664	0.662	**0.671**
NG9	0.475	0.542	0.559	0.599	0.613	0.643	**0.660**
Average	0.633	0.679	0.687	0.707	0.723	0.755	**0.760**
People-Place	0.774	0.775	0.744	0.791	0.783	0.824	**0.837**
Org-Place	0.706	0.713	0.720	0.748	0.777	0.820	**0.827**
Org-People	0.618	0.627	0.545	0.651	0.651	0.696	**0.709**
Average	0.700	0.705	0.670	0.730	0.737	0.780	**0.791**

3.6.1.3. Results and Discussion

We adopt the recall, precision, and F-measure as the evaluation metrics. Recall is defined as the number of articles that are correctly classified, divided by the actual number of articles in each class. Precision is defined as the number of articles that are correctly classified, divided by the number of all the articles predicted as the same class. F-measure is defined as the harmonic mean of recall and precision.

$$\texttt{F-measure} = \frac{2 \times \texttt{precision} \times \texttt{recall}}{\texttt{precision} + \texttt{recall}} \tag{3.49}$$

Results of all the methods on all data sets depicted in Table 3.2 and Table 3.3 are summarized in Table 3.4 with the best results shown in bold font. It can be observed that the supervised method, namely, SVM, which trains only in the source domain and tests in the target domain always gets the worst performance among the seven algorithms. Semi-supervised learning method TSVM outperforms the supervised learning method SVM by taking advantages of the unlabeled data in the target domain. Since the articles in the source domain and target domain are related, the unlabeled data in the target domain will supply some distribution information for the training process so as to improve the prediction in the target domain. CDSC has been reported for the good performance in two-class cross-domain adaptation. Those results are verified again in our experiments especially when the two classes in the target domain are well separated such as the data sets NG{1-3}. However, for multiclass problems especially when the multiple classes in the target domain are not very easy to separate such as the data sets NG{4-9}, the performance of CDSC is not as good as that in two-class problems. On the other hand, our domain adaptation method can get comparable results with CDSC for the well separated two-class problems and achieve better performance for all the other data sets. For both KMM and TCA, their strategies are to minimize the domain difference first by reweighing the features or samples. Then they employ standard classification model like SVM for final prediction. However, the first step may lose much useful information for learning the final predictors because it does not consider the label information. Therefore, the performance of KMM and TCA are not as good as our proposed LRSC method, which minimizes the domain gap and empirical loss on the labeled data simultaneously. Regarding the computational time aspect, according to our algorithm outlined in Figure 3.3, singular value decomposition (SVD) is the major computational issue with the complexity $\mathcal{O}(n^2)$, which is also the computation complexity of KMM and TCA.

3.6.2. Information Extraction

3.6.2.1. Experiment Setup

We conducted a set of experiments in the task of information extraction. The objective of information extraction is to extract precise text fragments, which are basically chunks of consecutive tokens, for each field of interest from a semi-structured text document. In our experiments, we aim at extracting the job related information from Web pages in some recruitment Web sites. The fields of interest are *job title*, *company*, *location*, *salary*, *post-date*, *education*, *experience*, and *duty*. The online job advertisement documents were collected from different recruitment Web sites in three different domains (or industries). Table 3.5 depicts the details of the collected data. The first, second, and third columns refer to the domain label, domain name, and the number of job advertisements collected in the domain respectively. For each online job advertisement collected, we automatically segment the document into a number of text fragments by considering the document object model (DOM)[2] and extract the text contained in the text nodes of the DOM structure. Long paragraphs contained in text nodes are further segmented into sentences by an automatic sentence segmentator for finer granularity. The fourth column of the table shows the number of text fragments in the domain after segmentation. Each text fragment should be labeled as one of the eight job fields mentioned above, or the "not-a-field" label. Two human accessors were invited to manually label all the text fragments in the three domains. If there was any disagreement on the judgment between the two accessors, it was resolved by a discussion among them. The manual label information is used as the ground truth in the experiments. Figure 3.6 depicts a sample web page with the labeled fields by the box in the accounting domain.

In each domain, we have conducted different sets of experiments to demonstrate the performance and compare with existing methods. We use the labeled training example in the source domain and the unlabeled data in the target do-

[2]The details of the document object model can be found in *http://www.w3.org/DOM*.

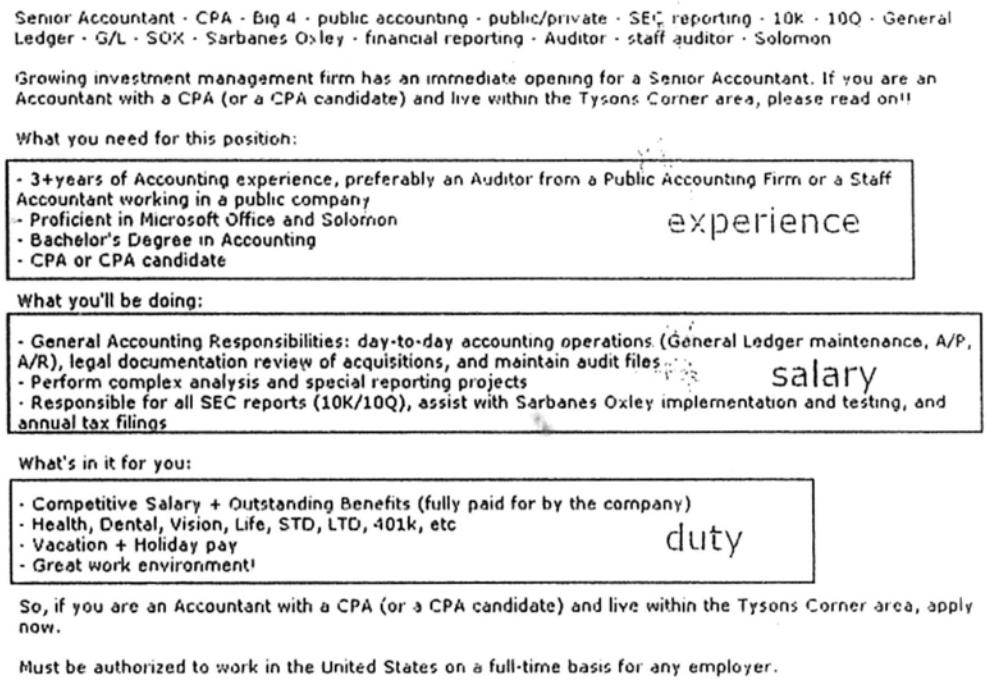

Figure 3.6: Sample web page showing its field labels in the accounting domain

Table 3.5: The details of the data collected for the information extraction experiments.

Domain Label	Domain Name	# of Job Advertisements	# of Text Fragments
D1	Accounting	273	7462
D2	Logistic	202	5636
D3	Health	201	6402

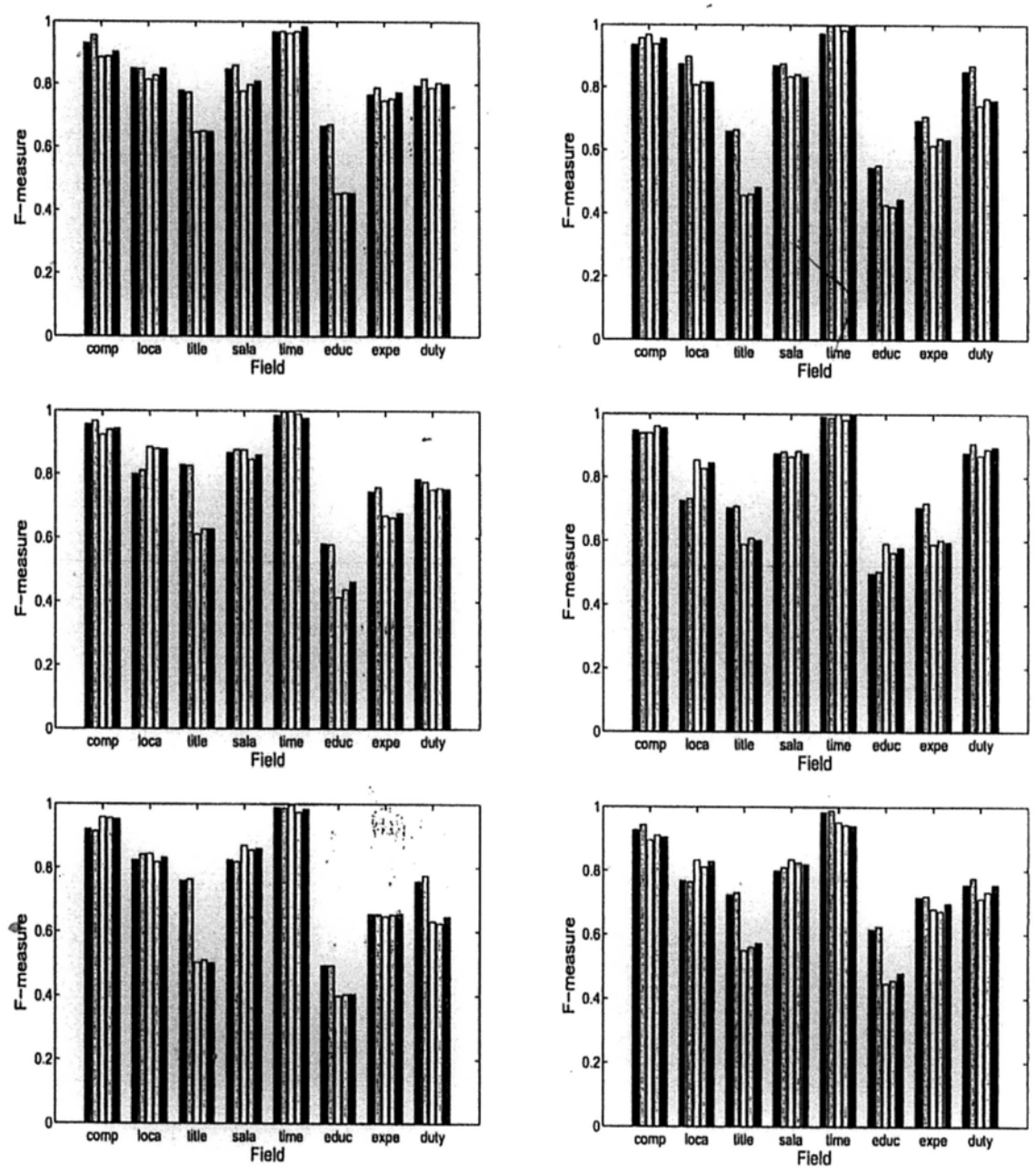

Figure 3.7: Comparison of the extraction performance of each job field with different source domain and target domain. 1^{st} Row : D_1-D_2, D_1-D_3, 2^{nd} Row : D_2-D_1, D_2-D_3, 3^{rd} Row : D_3-D_1, D_3-D_2. The fields in the x-axis from left to right are company, location, job title, salary, post-date, education, experience, and duty. Different color represent different comparison algorithm. blue: LRSC(linear), cyan: LRSC(kernel), green: TSVM, yellow: KMM, grey: TCA.

Table 3.6: The extraction performance of different sets of experiments. P, R, and F refer to the precision, recall, and F-measure respectively.

Data Set		TSVM			KMM			TCA			LRSC(linear)			LRSC(kernel)		
D_S	D_T	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
D1	D2	0.730	0.815	0.759	0.895	0.680	0.735	0.801	0.788	0.781	0.814	0.845	0.825	0.826	0.856	**0.841**
D1	D3	0.717	0.771	0.731	0.915	0.671	0.722	0.795	0.777	0.768	0.813	0.804	0.800	0.814	0.817	**0.815**
D2	D1	0.782	0.772	0.766	0.914	0.766	0.816	0.800	0.798	0.786	0.866	0.789	0.807	0.874	0.793	**0.832**
D2	D3	0.782	0.797	0.770	0.741	0.882	0.796	0.770	0.826	0.774	0.830	0.762	0.765	0.844	0.782	**0.812**
D3	D1	0.742	0.739	0.731	0.784	0.798	0.781	0.790	0.790	0.778	0.790	0.789	0.779	0.803	0.824	**0.813**
D3	D2	0.727	0.784	0.737	0.742	0.752	0.747	0.748	0.797	0.771	0.793	0.791	0.786	0.811	0.796	**0.803**
Average		0.743	0.775	0.744	0.832	0.758	0.766	0.784	0.796	0.776	0.820	0.800	0.799	0.829	0.811	**0.819**

main to learn the extraction model using our domain adaptation method. The learned model is then applied to the testing data in the target domain and the performance is measured. For example, let D1 and D2 be the source and target domains respectively. We use the labeled training fragments in D1 and the unlabeled fragments in D2 to learn a model. Then the learned model is applied to predict the fields of the text fragments in D2. As can be seen, in each training, the total number of text fragments in the source domain and target domain is larger than 10,000. Since CDSC needs to compute and store the pairwise similarity for any two fragments, it cannot handle this information extraction data set. We cannot compare with it due to memory consumption problem. The parameters setting is similar as the setting in document classification experiments. Note that each text fragment is represented by the vector space model and normalized to unit length.

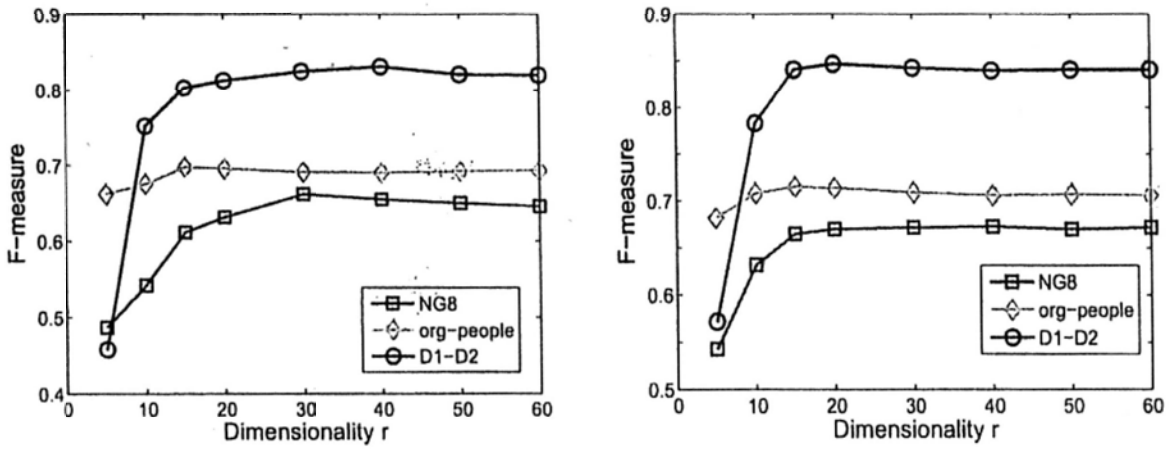

Figure 3.8: The effect of the dimensionality r on the performance of LRSC in three datasets. The left and right subfigures correspond to the linear and kernel case respectively

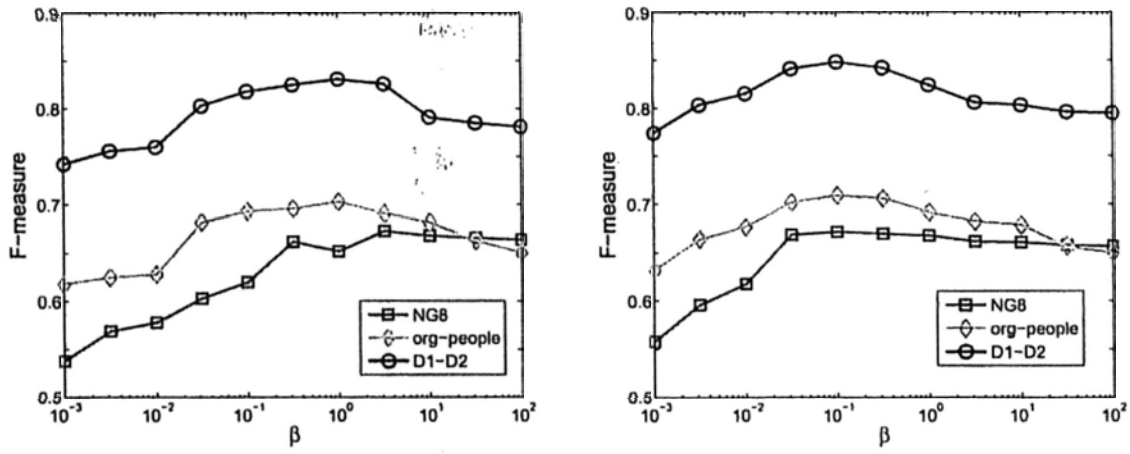

Figure 3.9: The effect of the weight β on the performance of LRSC in three datasets. The left and right subfigures correspond to the linear and kernel case respectively.

3.6.2.2. Results and Discussion

We adopt the recall, precision, and F-measure as the evaluation metrics. Recall is defined as the number of text fragments that are correctly labeled by our framework, divided by the actual number of text fragments. Precision is defined as the number of text fragments that are correctly labeled by our framework, divided by the number of predicted text fragments using our framework. F-measure is defined as the harmonic mean of recall and precision.

In each set of experiments, we have conducted 6 runs using different combination of the source and target domains. Table 3.6 depicts the performance of the experiments. In each run, we measure the recall, precision, and F-measure for each field. The figure in each cell of Table 3.6 is the average performance among the 8 fields of interest in the corresponding experiment. For example, our LRSC(kernel) method achieves an average precision, recall, and F-measure of 0.8262, 0.8563, and 0.8410 respectively in the target domain when the source and target domains are D1 and D2 respectively. Moreover, it achieves an average precision, recall, and F-measure of 0.8290, 0.8117, and 0.8198 on the whole six data sets. It outperforms TSVM, KMM and TCA, which obtains F-measure of 0.744, 0.766 and 0.776 respectively.

Figure 3.7 depicts the detailed comparison between our LRSC method and other methods, namely, TSVM, KMM and TCA. The x-axis denotes the eight job fields and the y-axis denotes the extraction performance measured by F-measure. In each plot, we show the F-measure on each job field when training is conducted in one domain and adapting to the other two domains. Different colors represent different comparison methods, and one subfigure contains the comparison in one dataset. It can be observed that all of the four methods obtain similar extraction performance in the fields of company, location, salary and post-date because the text fragments which describe those fields are almost the same in different domains. However, for the other fields, such as education, title, duty, and experience, the words or phrases vary among different domains.

TSVM does not consider such domain difference, then its performance is the worst among all the four methods as shown in the figures. KMM and TCA can re-weigh the features or samples so as to minimize the domains gap, but they do not use the label information. LRSC combines those ideas together and outperforms the other three methods in almost all the datasets.

3.6.3. Experimental Parameter Investigation

3.6.3.1. The effect of the dimensionality r

In this section, we investigate the effect of the dimensionality r on the domain adaptation performance. We select three datasets, namely, NG8, org-people, and D1-D2, as representatives of 20-Newsgroup, Reuters and job information respectively. The left sub-figure in Figure 3.8 depicts the performance on those three datasets in relation to $r = 5, 10, 15, 20, 30, \cdots, 60$ for kernel LRSC. We also implement the same experiment for linear LRSC, which can also get similar figure with kernel LRSC. The result shows that our proposed method LRSC, in both linear and kernel case, are not very sensitive to the change of the dimensionality r, which is a significant advantage for dimension reduction.

3.6.3.2. The effect of the weight β

As discussed above, β is the relative weight of the domain gap between D_S and D_T against the empirical loss on the labeled data in D_S. The higher value of β, the more concepts will be extracted to shrink the gap between D_S and D_T. Furthermore, when β tends to infinity, then kernelized LRSC will become similar as TCA. On contrary, if we decrease the value of β, LRSC tends to learn the classifiers without paying much attention on the distribution gap. The extreme case is when β tends to 0, LRSC will degenerate to the standard logistic regression. We fix the dimensionality r as 30, and vary β from 0.001 to 100. The datasets used are the same as in Section 3.6.3.1. The right sub-figure in Figure 3.9 demonstrates the performance of LRSC with varying β for kernel case. It can

be observed that the best performance β exists in the range [0.1,1] for almost all the datasets. Similar conclusion can be get for the linear LRSC.

3.6.4. Discussion

For those feature and instance weighting approaches, they all separate the domain adaptation framework to two steps, the first step is trying to minimize the domain gap, and then training the predictive model based on the learned feature or instance representations. They seldom consider these two steps interactively. For our proposed discriminative concept domain adaptation (LRSC) method, we can ensure the extracted concepts are shared by source and target domains, and favored by the classifiers learned by the labeled data in the source domain. Moreover, some extremal situations of our proposed methods can be degenerated to existing approaches. For example, if we tune the trade-off coefficient β to infinity, the extracted low-rank concepts should be mostly used for decrease the domain gap. Therefore, kernelized LRSC degenerates to TCA. Similarly, if we decrease β to zero, it is easy to see our method is the traditional logistic regression.

CHAPTER 4

Modeling Domain Difference Using High-order Statistics

4.1. Distribution Gap Measuring Metric

Both instance level and feature level domain adaptation approaches try to reduce the distribution gap between the training and testing set so as to propagate the label information. They have been proved to be effective in various applications. However, it is extremely hard to estimate the density function especially when the feature space is of high dimensional. Another major difficulty is how to incorporate an effective statistical criterion measuring distribution discrepancy into a tractable framework. Currently, most existing instance-level and feature-level approaches are restricted to the first-order statistics matching and enforce the empirical means of the training and testing instances be closer in a Reproducing Kernel Hilbert Space (RKHS). In Chapter 1, we demonstrate that those empirical mean based strategies can make the domain adaptation tasks possible to achieve good performance. However, intuitively, they may have a considerable limitation in matching two probability distributions where only the first-order statistics are exactly the same. Moreover, for many text mining applications, it is not appropriate to ignore the feature dependency which can be explored by considering the document covariance. Specifically, we can observe that the

sample covariance matrix on text data with zero mean is exactly the same as the term similarity matrix. This motivates us to utilize the covariance information to evaluate the distribution discrepancy. First it can strengthen the distribution matching criterion than only considering the mean. The second advantage is that we can utilize the feature dependency to distinguish domain specific features and common features, and then filter such features whose similarity with other features varies greatly from the training data to the testing data by investigating the sample covariance matrices.

In this chapter, in order to overcome the limitations mentioned above, we develop a new non-parametric distance metric called symmetric Stein's loss (SSL) to measure the distribution gap between two domains with finite samples. It jointly considers the empirical mean (Location) and sample covariance (Scatter) difference, and it can map the location and scatter information to one item smoothly which can avoid treating them separately. More specifically, we propose an improved symmetric Stein's loss (SSL) function which combines the mean and covariance discrepancy into a unified Bregman matrix divergence of which Jensen-Shannon divergence between normal distributions is a particular case.

4.2. Improved Symmetric Stein's Loss

Traditional domain adaptation methods try to reduce the distribution between D_S and D_T by evaluating the mean vector $\bar{x}$ and $\bar{x'}$ within a unit ball in RKHS. However, the main shortcoming of such methods is that they fail to capture the scatter information for both domains, which is crucial for classification. We try to tackle the problem by considering the sample scatter matrix to reduce the distribution gap, at the same time, consider the discrepancy between the mean vectors in RKHS. More importantly, we can further integrate those two important components in one framework by extracting the shared subspace between the source domain and target domain. For simplicity, we re-center the data in $\mathcal{S}$ by shifting the mean $\sum_{i=1}^{n_1} x_i/n_1$ to 0, that is, let $\mathbf{x} = \mathbf{x} - \sum_{i=1}^{n_1} x_i/n_1$ for any $\mathbf{x}$ in

both D_S and D_T. To simplify the notation without causing confusion, we still use x and x' to denote the data in D_S and D_T respectively.

We define the tuple (u, Σ) as representing the location vector and sample scatter matrix of the data in each domain D_S and D_T. Then we have:

$$\begin{aligned} u_S &= \frac{1}{n_1}\sum_{i=1}^{n_1} x_i,\ \Sigma_S = \frac{1}{n_1}\sum_{i=1}^{n_1}(x_i - u_S)(x_i - u_S)^\top \\ u_T &= \frac{1}{n_2}\sum_{i=1}^{n_2} x'_i,\ \Sigma_T = \frac{1}{n_2}\sum_{i=1}^{n_2}(x'_i - u_T)(x'_i - u_T)^\top \end{aligned} \tag{4.1}$$

However, due to the different nature of the two domains D_S and D_T, there may exist discrepancy between the tuples (u_S, Σ_S) and (u_T, Σ_T). In fact, there are many criteria to evaluate the difference between the matrices and vectors, for example, the 2-norm for the vectors and F-norm for the matrices. Here we introduce the Stein's loss [38], denoted by $B(\bullet, \bullet)$, to evaluate the difference between two matrices, which is originally adopted for estimating the covariance matrix and proved to be efficient for dominating the difference between two scatter matrices.

$$B(\Sigma_S, \Sigma_T) = \mathrm{tr}(\Sigma_S^+ \Sigma_T) - \mathrm{logdet}(\Sigma_S^+ \Sigma_T) - d \tag{4.2}$$

As can be seen the Stein's loss, also known as Bregman matrix divergence is exactly the generalized likelihood ratio when both distributions in D_S and D_T are multivariate normal distribution. Moreover, it can be shown that Stein's loss is the unique scale invariant loss function for which the unbiased estimator of the uniform minimum variance is also a minimum risk equivariant estimator. Scale invariance implies that the Bregman matrix divergence remains invariant under any scaling or invertible linear transformation P, since:

$$B(\Sigma_S, \Sigma_T) = B(P\Sigma_S P^\top, P\Sigma_T P^\top) \tag{4.3}$$

Furthermore, it has been shown that the Kullback-Leibler (KL) divergence between two multivariate Gaussians can be expressed as the convex combination of a Mahalanobis distance between the mean vectors and the Bregman divergence between the two scatter matrices [25]. In the sequel, we all use $p(\mathcal{S})$ and $p(\mathcal{T})$ to represent the probability density function $p(x|u_S, \Sigma_S)$ and $p(x|u_T, \Sigma_T)$ to simplify

the notation without causing confusion. The KL-divergence between $P(\mathcal{S})$ and $P(\mathcal{T})$ is:

$$\begin{aligned} \mathrm{KL}(p(\mathcal{S})||p(\mathcal{T})) &= \int p(\mathcal{S}) \frac{p(\mathcal{S})}{p(\mathcal{T})} dx \\ &= \frac{1}{2} B(\Sigma_{\mathcal{S}}, \Sigma_{\mathcal{T}}) + \frac{1}{2} M_{\Sigma_{\mathcal{T}}^{+}}(u_{\mathcal{S}}, u_{\mathcal{T}}) \end{aligned} \tag{4.4}$$

where $M_{\Sigma_{\mathcal{T}}^{+}}(u_{\mathcal{S}}, u_{\mathcal{T}}) = (u_{\mathcal{S}} - u_{\mathcal{T}})^{\top} \Sigma_{\mathcal{T}}^{+} (u_{\mathcal{S}} - u_{\mathcal{T}})$ is the Mahalanobis distance, parameterized by the covariance matrix $\Sigma_{\mathcal{T}}$. However, the KL-divergence is not symmetric and the *logdet* item is hard to compute. Therefore we introduce the Jensen-Shannon divergence instead to express the distribution of the two multivariate Gaussians:

$$\begin{aligned} &\mathrm{JS}(p(\mathcal{S})||p(\mathcal{T})) \\ &= \frac{1}{2}(\mathrm{KL}(p(\mathcal{S})||p(\mathcal{T})) + \mathrm{KL}(p(\mathcal{T})||p(\mathcal{S}))) \\ &= \frac{1}{4}(B(\Sigma_{\mathcal{S}}, \Sigma_{\mathcal{T}}) + B(\Sigma_{\mathcal{T}}, \Sigma_{\mathcal{S}})) + \frac{1}{4} M_{\Sigma_{\mathcal{S}}^{+} + \Sigma_{\mathcal{T}}^{+}}(u_{\mathcal{S}}, u_{\mathcal{T}}) - d \\ &= \frac{1}{4} \mathrm{tr}(\Sigma_{\mathcal{S}}^{+} \Sigma_{\mathcal{T}} + \Sigma_{\mathcal{T}}^{+} \Sigma_{\mathcal{S}}) + \frac{1}{4} M_{\Sigma_{\mathcal{S}}^{+} + \Sigma_{\mathcal{T}}^{+}}(u_{\mathcal{S}}, u_{\mathcal{T}}) - d \end{aligned} \tag{4.5}$$

Denote $\tilde{B}(\cdot, \cdot)$ as the symmetric Stein's loss function which is defined as:

$$\begin{aligned} \tilde{B}(\Sigma_{\mathcal{S}}, \Sigma_{\mathcal{T}}) &= \frac{1}{4}(B(\Sigma_{\mathcal{S}}, \Sigma_{\mathcal{T}}) + B(\Sigma_{\mathcal{T}}, \Sigma_{\mathcal{S}})) \\ &= \frac{1}{4} \mathrm{tr}(\Sigma_{\mathcal{S}}^{+} \Sigma_{\mathcal{T}} + \Sigma_{\mathcal{T}}^{+} \Sigma_{\mathcal{S}}) - d \end{aligned} \tag{4.6}$$

Hence we can represent the distribution gap between two multivariate Gaussians by the convex combination of the symmetric Stein's loss and the Mahalanobis distance between the mean vectors. However, we can see it is not trivial to combine them in a unified framework which is one of our goals in this paper. First we propose the following proposition.

Proposition 1. (i) For any d there is a 1-1 correspondence between the triple (u, Σ, λ) and matrix $A \in \mathcal{P}^{d+1}$, where $u \in \mathbb{R}^d$, $\lambda \in \mathbb{R}$ and $\Sigma \in \mathcal{P}^{d+1}$, given by $A = A(u, \Sigma, \lambda)$

$$A(u, \Sigma, \lambda) = \begin{pmatrix} \Sigma + \lambda^2 u u^{\top} & \lambda u \\ \lambda u^{\top} & 1 \end{pmatrix} \tag{4.7}$$

(ii) For any $A \in \mathcal{P}^{d+1}$, we have

$$A^{-1}(u, \Sigma, \lambda) = \begin{pmatrix} \Sigma^{-1} & -\lambda\Sigma^{-1}u \\ -\lambda u^{\top}\Sigma^{-1} & 1 + \lambda^2 u^{\top}\Sigma^{-1}u \end{pmatrix} \tag{4.8}$$

It is not difficult to verify the proposition above. According to the above proposition, we can map the mean vector and scatter matrix to a high order matrix in $\mathcal{P}^{d+1}$ by a 1-1 correspondence transformation. Moreover, we can just use the quadratic loss between $A(u_S, \Sigma_S, \lambda)$ and $A(u_T, \Sigma_T, \lambda)$ in the matrix group $\mathcal{P}^{d+1}$ to dominate the difference between the tuples (u_S, Σ_S) and (u_T, Σ_T), which can greatly harness the difficulty of handling the mean vector and scatter matrix separately. In the sequel, we denote $A(u_S, \Sigma_S, \lambda)$ and $A(u_T, \Sigma_T, \lambda)$ as A_S and A_T respectively to simplify the notation.

Theorem 2. The Jensen-Shannon divergence between two multivariate Gaussians parametrized by (u_S, Σ_S) and (u_T, Σ_T) can be represented by a special case of the symmetric Stein's loss between A_S and A_T, specifically,

$$\mathrm{JS}(p(S)||p(T)) = \tilde{B}(A_S, A_T)|_{\lambda=1} \tag{4.9}$$

Proof. As defined in the Proposition 1 we have:

$$A_S^{+} = \begin{pmatrix} \Sigma_S^{+} & -\lambda\Sigma_S^{+}u_S \\ -\lambda u_S^{\top}\Sigma_S^{+} & 1 + \lambda^2 u_S^{\top}\Sigma_S^{+}u_S \end{pmatrix}, \quad A_T = \begin{pmatrix} \Sigma_T + \lambda^2 u_T u_T^{\top} & \lambda u_T \\ \lambda u_T^{\top} & 1 \end{pmatrix}$$

It follows that:

$$\begin{aligned} \mathrm{tr}(A_S^{+}A_T) &= \mathrm{tr}(\Sigma_S^{+}\Sigma_T) + \lambda^2(u_S - u_T)^{\top}\Sigma_S^{+}(u_S - u_T) \\ &= \mathrm{tr}(\Sigma_S^{+}\Sigma_T) + \lambda^2 M_{\Sigma_S^{+}}(u_S, u_T) \end{aligned}$$

Similarly we can get:

$$\mathrm{tr}(A_T^{+}A_S) = \mathrm{tr}(\Sigma_T^{+}\Sigma_S) + \lambda^2 M_{\Sigma_T^{+}}(u_S, u_T) \tag{4.10}$$

According to the definition of symmetric Stein's loss in Equation 4.6, we have:

$$\begin{aligned} \tilde{B}(A_S, A_T) &= \tfrac{1}{4}\mathrm{tr}(A_S^{+}A_T + A_T^{+}A_S) - d \\ &= \tfrac{1}{4}\mathrm{tr}(\Sigma_S^{+}\Sigma_T + \Sigma_T^{+}\Sigma_S) + \tfrac{\lambda^2}{4} M_{\Sigma_S^{+}+\Sigma_T^{+}}(u_S, u_T) - d \end{aligned}$$

Combining with Equation 4.5 and setting $\lambda = 1$, the proof is complete. □

Theorem 2 guarantees that if the distributions are multivariate Gaussian distribution in $\mathbb{R}^d$, then our proposed symmetric Stein's loss in $\mathbb{R}^{d+1}$ not only considers the covariance difference but also consider the mean shift. λ can be viewed as the tradeoff coefficient between the two kinds of losses. Moreover, the state-of-the-art Jensen-Shannon divergence is just one particular case of our proposed distribution distance/divergence metric.

4.3. Empirical Test on Two-Sample Problems

In this section, we present the empirical experiment of our proposed symmetric Stein's loss function in various datasets, including the synthetic datasets, and the real-world datasets.

4.3.1. Related Test Methods

Various empirical methods have been proposed to test whether two random samples are generated from the same underlying distributions or not [15]. The most simplest method is the generalized multivariate t-test [34, 35], which assumes that both distributions are Gaussians with the same covariance. Kolmogorov-Smirnov statistic and Wald-Wolfowitz test are powerful when the null hypothesis is the fact that the underlying distribution is $\mathcal{P} = \mathcal{Q}$ for finite samples. Both of these two testing approaches are model-free univariate tests. Friedman and Rafsky [29] proposed to generalize to multivariate problems by counting the number of edges in the minimal spanning tree over the aggregated data that connects the points in D_S to the points in D_T. The computational complexity of the generalized multivariate test is $O((n_1 + n_2)^2 log(n_1 + n_2))$.

Hall and Tajvidi [33] proposed to aggregate the data in both domains as $Z = \{D_S, D_T\}$, and then finds the k points in Z closest to each points in D_S for all $k \in 1, ..., n_1$. Among the found k closest points, we count how many of these

are from $D_{\mathcal{T}}$, and compare with the number of points expected under the null hypothesis. The shortcoming of this method is that it is very time consuming and can be only applied to tens of points. Another direction is to use some distance metric, such as L_1 and L_2 to calculate the estimated densities as the statistic directly. One problem with this kind of approaches is to require the partition of the space, and it will become difficult or even impossible for high dimensional problems, such as text mining and bioinformatics.

Maximal Mean Discrepancy (MMD) was proposed based on the fact that two distributions are different if and only if there exists at least one function having different expectation on the two distributions. Consequently the maximum discrepancy between the function means can be used as the basis of a test statistic as demonstrated in Equation 3.14. MMD can take advantage of the kernel trick so that it can be applied to not only the vector space, but also strings, sequences and graphs. Moreover, it is easy to implement, memory efficient, and fast to compute. However, at the same time, due to the over flexibility, it is difficult to choose the most suitable kernel for different datasets.

4.3.2. Convergence to the Jensen-Shannon Divergence

As described in Theorem 2, we have proved that Jensen-Shannon divergence can be regarded as a special case of our proposed SSL to measure the distribution gap. In this section, we present some empirical results on two synthetic datasets to verify the convergence of the proposed model-free estimator.

The first synthetic dataset is composed of two zero-mean Gaussians with variance 1 and 2. In each round, we randomly generate n_1 samples according to $\mathcal{N}(0,1)$ as the samples in $D_{\mathcal{S}}$, and also generate $n_2 = n_1$ samples according to $\mathcal{N}(0,2)$ as the samples for $D_{\mathcal{T}}$. Then we calculate symmetric Stein's loss (SSL) $\tilde{B}(A_{\mathcal{S}}, A_{\mathcal{T}})$ with $\lambda = 1$. For each n_1, we repeat the process for 100 times to calculate the mean value and its associated two standard deviation confidence intervals. We vary the sample number n_1 from 10 to 100000 to investigate the convergence rate to the true Jensen-Shannon divergence.

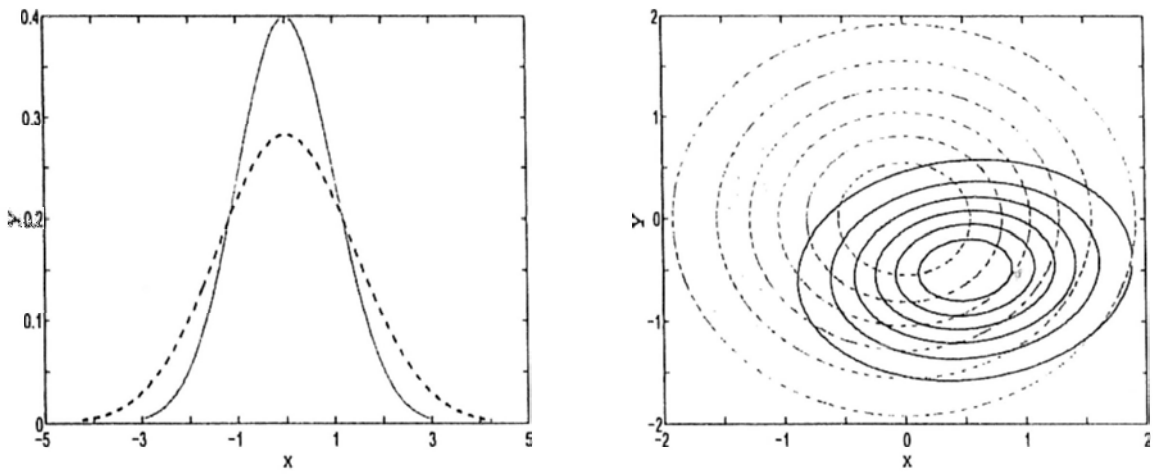

Figure 4.1: Sample distribution on the two synthetic dataset. Left: cumulative density function of $\mathcal{N}(0,1)$ and $\mathcal{N}(0,2)$. Right: Contour of the two Gaussian distributions given in Equation 4.11.

The second synthetic dataset is generated in 2-dimensional space according to the following distribution functions:

$$\mathcal{P}(x) = \mathcal{N}(\mathbf{0}, \mathrm{I}) \tag{4.11}$$

$$\mathcal{Q}(x') = \mathcal{N}(\begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & 0.3 \end{bmatrix}) \tag{4.12}$$

It can be seen that both the means and covariance structures of the two distributions are different, and we can directly calculate the Jensen-Shannon divergence between $\mathcal{P}(x)$ and $\mathcal{Q}(x')$ is 0.659. In each iteration, we calculate SSL value with $\lambda = 1$.

The experiment setting for the two synthetic dataset are exactly the same, and the detailed results are depicted in Figure 4.2. It can be seen that SSL can rapidly converge to the true Jensen-Shannon divergence with the increased sample number. Meanwhile, from the plot corresponding to the 1 standard deviation confidence interval as shown by blue lines in Fig. 4.2, we can judge that the rapid convergence is with high confidence.

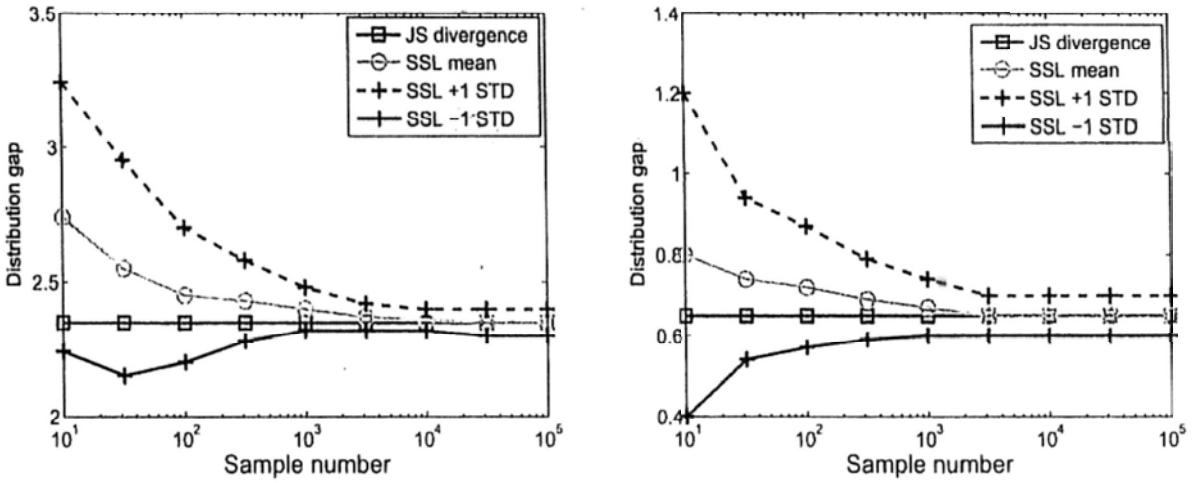

Figure 4.2: Performance of our proposed symmetric Stein's loss measure of the distribution gap on the synthetic dataset. Left: distribution gap between $\mathcal{N}(0,1)$ and $\mathcal{N}(0,2)$. Right: distribution gap between the distributions given in Equation 4.11. JS in the legends refer to Jensen-Shannon.

4.3.3. Convergence test on covariance structure

As discussed in Section 4.2, the properties of our proposed symmetric Stein's loss is that SSL can make use of the second order statistics to measure the distribution gap, which is much more statistical sufficient than other method only considering the kernel mean information. We conduct a group of experiments on synthetic datasets to to demonstrate this advantage.

The left diagram of Fig. 4.3 depicts the data distribution for source domain D_S and the target domain D_T, where we randomly generate 1000 samples according to a Gaussian distribution given in Equation 4.13, and fix the location of samples in D_S as [00], then we rotate all the samples with the angle θ to generate the samples in D_T. Obviously, the empirical mean for the two domains are the same, but the covariance structure shifts. Then we varies the angle θ from 0 to π to demonstrate the SSL and MMD value changes. Here we set the kernel of MMD as RBF kernel with the kernel width $\sigma = 0.1$, and we also normalize the

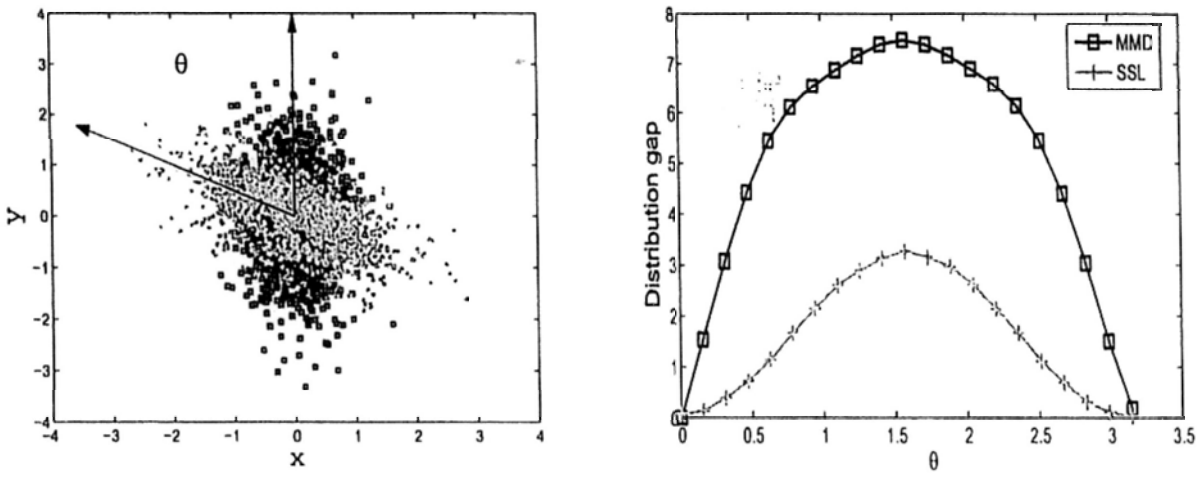

Figure 4.3: Performance of our Proposed SSL measure of the distribution gap on the synthetic dataset with shifting covariance structure. Left: samples in D_S and D_T . Right: distribution gap between the distributions with angel θ rotating.

features to unit length.

$$\mathcal{P}(x) = \mathcal{N}(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0.2 & 0\\0 & 1.0\end{bmatrix}) \tag{4.13}$$

$$\mathcal{Q}(x') = \mathcal{N}(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\\sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}0.2 & 0\\0 & 1.0\end{bmatrix}\begin{bmatrix}\cos(\theta) & \sin(\theta)\\-\sin(\theta) & \cos(\theta)\end{bmatrix}) \tag{4.14}$$

The right diagram in Fig. 4.3 depicts the distribution gap changes with the angel Θ rotating from 0 to π, where we multiply the MMD value with 100 so as to force the MMD and SSL value in the same scale. It is obviously that the SSL value can accurately measure the difference between D_S and D_T with the θ changes and the curve plotted by green line fitted as the $cos(2\Theta)$ function. However, MMD seems preferring to enlarge the difference more quickly even then Θ is very small, such that it is not so smooth as SSL.

4.3.4. Microarray dataset

In the current biological study, it is very essential to conduct statistical test of whether two microarray measurements correspond to the same subject. The test obtained by two different labs or on two different platforms, can be used for joint analysis. We can treat such test as the basic two-sample test. If the test statistic rejects the null hypothesis that the microarray data are generated from the same distribution, then we regard the two microarray datasets as non-comparable. Conversely if the test statistic holds the hypothesis, then we treat them as comparable.

We obtain two public microarray benchmark datasets from Warnat *et al.* [60]. Both of the two datasets comprise of the same set of 2166 genes, in other words, the dimensionality of the two datasets are all 2166. However, they are generated from different platforms. We conduct four groups of experiment on this microarray datasets based on different combinations of the platform similarity and the hypothesis. Take the first group of experiment for example, we randomly select 25 samples as the source domain data, and 25 samples as the target domain data, both from the same platform without replacement, while the hypothesis is that the samples in the source domain and the target domain are generated from the same platform. Then we use different test methods to determine whether the test statistic holds the hypothesis. If it holds, then the count number add 1. We repeat the testing for 100 times to avoid the bias and coincidence.

We compared SSL to the multivariate t-test, Worf *et al.* test and Simirnov *et al.* test using spanning tree (denoted as ST_{worf} and $ST_{simirnov}$), and MMD. We set the significance level α as 0.05 for all the methods. For MMD, we employ the Gaussian kernel with the kernel width $\sigma = 20$.

Detailed results are reported in Table 4.1, showing the number of times SSL and other four methods regard two samples as generating from the same distribution when data are selected from same or different platforms. Generally

Table 4.1: Cross platform empirical test using microarray dataset

Platforms	H_0	t-test	ST_{worf}	$ST_{simirnov}$	MMD	SSL
same	accepted	100	93	95	100	100
same	rejected	0	7	5	0	0
different	accepted	95	0	29	0	0
different	rejected	5	100	71	100	100

speaking, SSL and MMD can get comparative performance, both of them can correctly recognize whether the data samples were generated from the same platform or not, thus they make no Type I or Type II error, while other three methods cannot distinguish the cross platform samples. Moreover, we also note that the sample size is relatively small (only 25 for each domain), which usually cause the difficulty, or even impossible to converge to the true distribution measure. The good performance of SSL thus significantly demonstrates its advantages.

CHAPTER 5

Location and Scatter Matching

5.1. INTRODUCTION

Most of the current domain adaptation approaches only consider the first-order statistics to evaluate the distribution difference due to the difficulty of modeling high-order statistics into a non-parametric distribution metric. In order to overcome the limitations, we develop another new method called Location and Scatter Matching (LSM) that is composed of a non-parametric distance metric with a good property which jointly considers the empirical mean (Location) and sample covariance (Scatter) difference. More specifically, based on our proposed symmetric Stein's loss function presented in Chapter 4, which combines the mean and covariance discrepancy into a unified Bregman matrix divergence of which Jensen-Shannon divergence between normal distributions is a particular case, we try to find a good feature representation which can reduce the embedded distribution gap, at the same time, ensure the new derived representation can encode sufficient discriminants with respect to the label information. Then a standard machine learning algorithm can be adapted to train classifiers in the new feature subspace across domains.

5.2. Motivation and Illustration by Synthetic Data

5.2.1. Motivation of Our Approach

As stated above, most domain adaptation techniques (either instance-level or feature-level approaches) try to reduce the distribution discrepancy with respect to a specific statistic criterion. Currently, the empirical mean is the most common statistics used to evaluate the distribution distance, and its non-parametric form MMD, has been applied in many algorithms such as MMED, KMM, TCA, etc. However, intuitively, just considering the first-order statistics such as the mean match is not statistical efficient to test the hypothesis of distribution closeness. In many situations, even the empirical means both in the original feature space, and in the Reproducing Kernel Hilbert Space are strictly matched, the distributions still differ greatly. On the other hand, if we can take the second-order statistics such as sample covariance into consideration, then the statistics should be more sufficient to characterize a probability distribution.

Another observation which motivates us to consider the sample covariance matrix comes from the text data property. Typically the model trained in D_S shows degradation in performance in D_T. One major reason is that the domain specific features which are very discriminative in D_S become unimportant in D_T. Then the prediction power of the common features which are discriminative in both domains will be decreased. In fact, the discriminative power of most terms in text documents may also vary slightly from domain to domain. By investigating the variance of the term similarity matrix from D_S to D_T, we can discover the domain specific terms and deal with them appropriately. One important observation is that the document covariance matrix is exactly the same as the term similarity matrix with constant permutation. Based on the above motivation, we try to extract a feature subspace on which the documents can be represented so as to reduce the distribution gap characterized by both

the mean and sample covariance.

5.2.2. Illustration by Synthetic Data

In order to demonstrate the shortcoming of the existing algorithms based on first-order statistics, such as the mean match, we generate a synthetic dataset to investigate in-depth the advantages of our proposed algorithm over existing ones. There are 400 instances in the synthetic dataset, 200 positive and 200 negative, in both D_S and D_T. Both domains have the same set of 400 features. The 200 positive instances in D_S, denoted as PS_{200}, are generated by the normal distribution as follows:

$$PS_{200} \sim \mathcal{N}([6\mathbf{e}, \mathbf{e}, \mathbf{0}, \mathbf{e}], \Sigma)$$

where $\mathbf{e}$ is all-one vector with length 100, and $\mathbf{0}$ is a null vector with length 100. Σ is a 400×400 identity matrix in $\mathbb{R}^{400}$. The mean of the features with identifier $1 - 100$ attain a larger value of 6. The mean of the features with identifier $101 - 200$ and $301 - 400$ attain a value of 1. The 200 negative instances in the source domain, denoted by NS_{200}, are generated by another normal distribution as follows:

$$NS_{200} \sim \mathcal{N}([\mathbf{0}, \mathbf{0}, \mathbf{e}, \mathbf{e}], \Sigma)$$

Let PT_{200} and NT_{200} denote the 200 positive and 200 negative instances in the target domain respectively. The data distributions in the target domain are given as follows:

$$PT_{200} \sim \mathcal{N}([3\mathbf{e}, \mathbf{e}, \mathbf{0}, \mathbf{0}], \Sigma), \quad NT_{200} \sim \mathcal{N}([3e, \mathbf{0}, \mathbf{e}, 2\mathbf{e}], \Sigma)$$

Figure 5.2.2 depicts the synthetic data in D_S and D_T. The grey level is proportional to the value whenever it is larger than 0.1.

It can be observed that the features with identifier $1-100$ are very discriminative in D_S but not in D_T, and features with identifier $301-400$ are very discriminative in D_T but not in D_S. Features with identifier $100 - 300$ are very discriminative

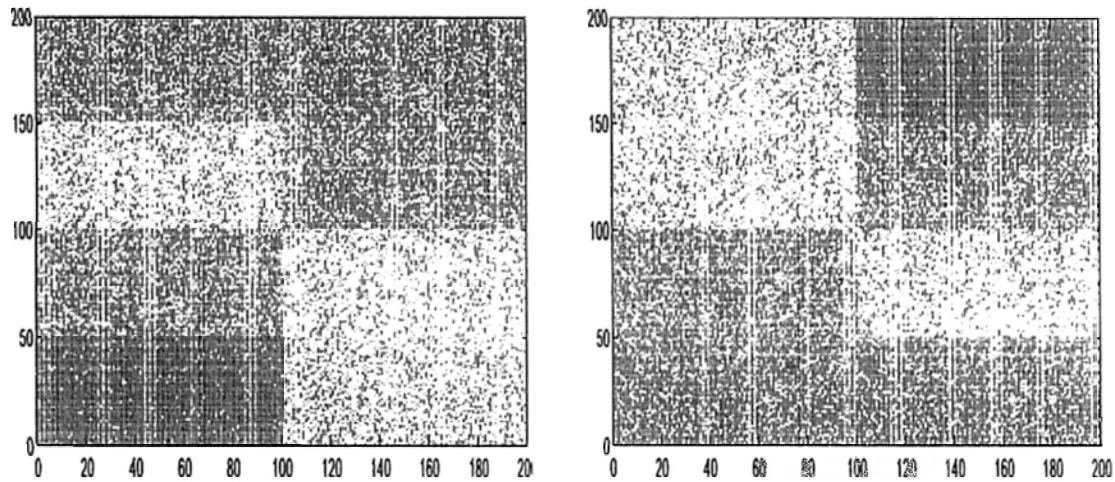

Figure 5.1: The synthetic data in the source domain (left) and in the target domain (right). x-axis represents the sample identifier, and the y-axis represents the feature identifier.

in both domains. This setting is a common setting found in many text mining applications in which common features and domain specific features exist. Obviously the empirical mean of the data in D_S and D_T is almost same. However, the data distribution still differs greatly from D_S to D_T. In fact, by constructing the data scatter matrix, we can find that the correlations between features with identifier (1-100, 301-400) and features with identifier 101-300 change greatly from D_S to D_T.

We have implemented two existing domain adaptation algorithms based on empirical mean match. One is Kernel Mean Matching (KMM) [37]. The other one is Transfer Component Analysis (TCA) [49]. We applied TCA, KMM, and our approach (called LSM) on this synthetic dataset. For both KMM and TCA, we investigated the linear kernel. For TCA, we varied the number of derived features from 4 to 20, and selected the best performance to report. For our LSM method, we set the number of derived features as 4. After re-weighing or feature extraction, Support Vector Machine (SVM) is employed as the classifier[1]. We

[1]http://svmlight.joachims.org

also applied SVM as a traditional learning algorithm in the original feature space on this dataset. We use precison, recall, and F-measure as the evaluation metrics whose definition are given in Section 3.

The performance of each algorithm is given in Table 5.1. The trained model by traditional learning algorithm such as SVM gives wrong predictions to the negative samples in D_T due to the discriminants decrease of the features with identifier $1-100$. Overall, the performance of KMM and TCA is much worse than our proposed LSM algorithm because the empirical mean in this synthetic data cannot distinguish the distribution difference between D_S and D_T. For LSM, after we filter such domain specific features, such as the features with identifier 1-100, by considering the covariance matrix shift, the performance increases greatly. We also observe that the performance of KMM is close to SVM because the weights among all training samples in D_S are similar even after de-biasing.

Table 5.1: Classification performance of the synthetic data

	performance		
Algorithms	P	R	F1
SVM	0.606	0.600	0.594
KMM	0.623	0.616	0.611
TCA	0.731	0.708	0.701
LSM	**0.935**	**0.935**	**0.935**

5.3. Location and Scatter Matching

Given the data samples in both source domain and target domain, and the same domain adaptation problem definition as described in Section 3.2. Our proposed LSM domain adaptation method aims to find a linear transformation (projection) $\Theta \in \mathbb{R}^{m \times d}$ such that the discrepancy on the sample covariance matrix and mean vector between the source domain and target domain are minimized. In other

words, we try to learn an optimal representation which can decrease the distribution gap between D_S and D_T on high order statistics. Based on Theorem 2, we can just minimize the following objective function:

$$F(A_S, A_T) = \frac{1}{2}\mathrm{tr}((\Theta A_S \Theta^\top)^+ \Theta A_T \Theta^\top + (\Theta A_T \Theta^\top)^+ \Theta A_S \Theta^\top)$$

For high dimensional data, especially when the number of samples is less than the dimension, the estimation of the total covariance (scatter) matrix is often unreliable. The regularization technique is commonly applied to improve the estimation as follows:

$$A_S = A_S + \epsilon I_d \tag{5.1}$$

However, in order to avoid extracting some noise features which are not important in both D_S and D_T, we do not add the regularization item for A_T. The first reason is that it is hard to tune an optimal regularization coefficient for both sample covariance matrices, and the second reason is that we can reduce the distortion from the original.

5.3.1. Solving the optimal transformation Θ

Theorem 3. Suppose we have ensured that A_S is positive definite matrix in Equation 5.1, there exists an invertible matrix Φ which can diagonalize them simultaneously, such that:

$$\Phi A_S \Phi^\top = I_{d+1} \ \ and \ \ \Phi A_T \Phi^\top = \begin{pmatrix} \Lambda_T & 0 \\ 0 & 0 \end{pmatrix} = \Gamma_T \tag{5.2}$$

where I_{d+1} is the identity matrix in $\mathbb{R}^{d+1}$, and $\Lambda_T = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_q)$ is a diagonal matrix in $\mathbb{R}^q$ which satisfies that $0 < \lambda_1 + \frac{1}{\lambda_1} \leq ... \leq \lambda_q + \frac{1}{\lambda_q}$.

Proof. A_S is a positive-definite matrix in Equation 5.1, then we can find an orthogonal matrix P which can diagonalize A_S as $PA_SP^\top = \Gamma = diag(\sigma_1, .., \sigma_{d+1})$, and $\sigma_i > 0$ for $i = 1, ..., d+1$ [31]. Denote $\Psi = \Gamma^{-\frac{1}{2}} P^\top A_T P \Gamma^{-\frac{1}{2}}$, then it can be verified that Ψ is a positive semi-definite matrix. Then there exists an orthogonal matrix Q which can diagonalize Ψ as $Q\Psi Q^\top = diag(\lambda_1, ..., \lambda_q, 0, ..., 0)$ where

$q = rank(A_{\mathcal{T}})$ and all the diagonal elements larger than 0. Let $\Phi = Q^{\top}\Gamma^{-\frac{1}{2}}P^{\top}$, then we have:

$$\Phi A_{\mathcal{S}}\Phi^{\top} = I_{d+1}, \quad \Phi A_{\mathcal{T}}\Phi^{\top} = \begin{pmatrix} \Lambda_{\mathcal{T}} & 0 \\ 0 & 0 \end{pmatrix} = \Gamma_{\mathcal{T}} \tag{5.3}$$

If the ranking of $\lambda_1, ..., \lambda_q$ dose not satisfy $\lambda_1 + \frac{1}{\lambda_1} \leq ... \leq \lambda_q + \frac{1}{\lambda_q}$, we can always find a permutation matrix Δ to permutate the ranking so as to satisfy the requirement. Then $\Phi = \Delta\Phi$ can hold the hypothesis in Theorem 2. □

Theorem 4. Let Φ be the matrix defined above and Φ_p be the submatrix spanned by the first p rows of Φ where $0 < p \leq q$ and $q = \text{rank } (A_{\mathcal{T}})$. Then $\Theta = M\Phi_p \in \mathbb{R}^{p\times d}$ minimizes $F(A_{\mathcal{S}}, A_{\mathcal{T}})$ for any non-singular matrix $M \in \mathbb{R}^{p\times p}$.

Proof. Based on the result in Eq. 5.2, we have

$$\Theta A_{\mathcal{S}}\Theta^{\top} = \Theta\Phi^{-1}(\Phi A_{\mathcal{S}}\Phi^{\top})(\Phi^{-1})^{\top}\Theta^{\top} = \hat{\Theta}\hat{\Theta}^{\top} \tag{5.4}$$

$$\Theta A_{\mathcal{T}}\Theta^{\top} = \Theta\Phi^{-1}(\Phi A_{\mathcal{T}}\Phi^{\top})(\Phi^{-1})^{\top}\Theta^{\top} = \hat{\Theta}\Gamma_{\mathcal{T}}\hat{\Theta}^{\top} \tag{5.5}$$

where $\hat{\Theta} = \Theta\Phi^{-1}$. Then let $\hat{\Theta} = (\Theta_1, \Theta_2)$ be the partition of $\hat{\Theta}$ so that $\Theta_1 \in \mathbb{R}^{p\times q}$ and $\Theta_2 \in \mathbb{R}^{p\times(d-q)}$, we have

$$\Theta A_{\mathcal{S}}\Theta^{\top} = \Theta_1\Theta_1^{\top}, \qquad \Theta A_{\mathcal{T}}\Theta^{\top} = \Theta_1\Lambda_{\mathcal{T}}\Theta_1^{\top} \tag{5.6}$$

Hence

$$\begin{aligned} F(A_{\mathcal{S}}, A_{\mathcal{T}}) &= y\tfrac{1}{2}\text{tr}((\Theta_1\Theta_1^{\top})^{+}\Theta_1\Lambda_{\mathcal{T}}\Theta_1^{\top} + (\Theta_1\Lambda_{\mathcal{T}}\Theta_1^{\top})^{+}\Theta_1\Theta_1^{\top}) \\ &= \tfrac{1}{2}\text{tr}(\Theta_1^{\top}(\Theta_1\Theta_1^{\top})^{+}\Theta_1\Lambda_{\mathcal{T}} + \Theta_1^{\top}(\Theta_1\Lambda_{\mathcal{T}}\Theta_1^{\top})^{+}\Theta_1) \\ &= \tfrac{1}{2}\text{tr}(\Theta_1^{+}\Theta_1(\Lambda_{\mathcal{T}} + \Lambda_{\mathcal{T}}^{+})(\Theta_1^{+}\Theta_1)^{\top}) \end{aligned}$$

where the last equality is based on the conclusion that $(AA^{\top})^{+} = (A^{+})^{\top}A^{+}$, and its generalized conclusion that $(A\Lambda A^{\top})^{+} = (A^{+})^{\top}\Lambda^{+}A^{+}$ for any matrix A and diagonal matrix Λ.

Remind that $\Lambda_{\mathcal{T}} = \text{diag}(\lambda_1, ..., \lambda_q)$, then $\Lambda_{\mathcal{T}} + \Lambda_{\mathcal{T}}^{+} = \text{diag}(\lambda_1 + \frac{1}{\lambda_1}, ..., \lambda_q + \frac{1}{\lambda_q})$, where $0 < \lambda_1 + \frac{1}{\lambda_1} \leq ... \leq \lambda_q + \frac{1}{\lambda_q}$.

Let $\Theta_1 = R\begin{pmatrix}\Lambda_\Theta & 0\end{pmatrix}S^\top$ be the SVD of Θ_1 where $R \in \mathbb{R}^{p\times p}$ and $S \in \mathbb{R}^{q\times q}$, $\Lambda_\Theta \in \mathbb{R}^{p\times p}$ is diagonal matrix. Then we have $\Theta_1^+\Theta_1 = S\begin{pmatrix} I_p & 0 \\ 0 & 0\end{pmatrix}S^\top$. It follows that

$$\begin{aligned} F(A_S, A_T) &= \tfrac{1}{2}\mathrm{tr}(S\begin{pmatrix} I_p & 0 \\ 0 & 0\end{pmatrix}S^\top(\Lambda_T + \Lambda_T^+)S\begin{pmatrix} I_p & 0 \\ 0 & 0\end{pmatrix}S^\top) \\ &= \tfrac{1}{2}\mathrm{tr}(S^\top S S_\Theta^\top(\Lambda_T + \Lambda_T^+)S_\Theta) \\ &= \tfrac{1}{2}\mathrm{tr}(S_\Theta^\top(\Lambda_T + \Lambda_T^+)S_\Theta) \\ &\geq \tfrac{1}{2}(\lambda_1 + \tfrac{1}{\lambda_1} + ... + \lambda_p + \tfrac{1}{\lambda_p}) \end{aligned}$$

where $S_\Theta = S\begin{pmatrix} I_p \\ 0\end{pmatrix}$ are the first p columns of the orthogonal matrix S. The equality holds when $S_\Theta = \begin{pmatrix} W \\ 0\end{pmatrix}$, where $W \in \mathbb{R}^{p\times p}$ is an arbitrary orthogonal matrix. It follows that

$$\Theta_1 = R\begin{pmatrix}\Lambda_\Theta & 0\end{pmatrix}S^\top = R\Lambda_\Theta(W\ \mathbf{0}) = (R\Lambda_\Theta W\ \mathbf{0}) \tag{5.7}$$

Here we can observe that R and S_Θ are both arbitrary orthogonal matrices, Λ_Θ is an arbitrary diagonal matrix. Denote $M = R\Lambda_\Theta W$, Hence $M \in \mathbb{R}^{p\times p}$ is an arbitrary matrix. Remind that the minimal value of $F(A_S, A_T)$ is independent of Θ_2, so we can let Θ_2 be $\mathbf{0}$. Based on the definition of $\hat{\Theta}$, we can conclude that

$$\Theta = \hat{\Theta}\Phi = (M\ \mathbf{0})\Phi = M\Phi_p \tag{5.8}$$

This completes the proof of this theorem. □

From Theorem 4 we can verify that for any non-singular matrix M, $M\Phi_p$ can minimize $F(A_S, A_T)$. Specifically, after the same linear transformation $M\Phi_p$, the distribution gap between domain $\mathcal{S}$ and $\mathcal{T}$ can be minimized. The most simplest one is let $M = I_p$. However, for our practical use, we need to find an optimal M which can fit our classification problem very well. In the following, we try to

find this optimal M under the Empirical Risk Minimization framework (ERM) proposed in [58].

5.3.2. Training on the optimal M

In fact, in practical text mining problems, many terms behave similarly in D_S and D_T, for example, the common non-discriminative features may vary slightly from D_S to D_T. We should use the label information to further filter such features by learning an optimal M. In order to capture the label dependency, we define the m decision functions as $f_l(x) = w_l^\top \Theta x$ where $l = 1, ..., m$ where w_l is the prediction weight vector for the class label l. We employ the square loss function $\ell(f_l, x_i, Y_{il}) = (f_l(x_i) - Y_{il})^2$ to measure the empirical loss on the labeled data in D_S. Then the total loss can be formulated as:

$$\sum_{l=1}^{m} \sum_{i=1}^{n_1} \ell(f_l, Y_{il}, x_i) \quad = \quad \left\| W^\top X_S - Y^\top \right\|^2$$

where $X_S = [x_1, \cdots, x_{n_1}] \in \mathbb{R}^{d \times n_1}$ is the data matrix of the source domain, $W = [w_1, \cdots, w_m] \in \mathbb{R}^{d \times m}$, $Y_{il} = 1$ if the i-th sample belongs to the l-th class, and 0 if it is labeled as others.

Based on the parametric form defined in the decision function f_l, we introduce $\left\| w^\top \Theta \right\|$ as the regularizer. Recall that $\Theta = M\Phi_p$, we arrive at the following minimization problem on learning the optimal M and its corresponding predictors:

$$\min_{M,W} \quad \left\| W^\top M \Phi_p X_S - Y^\top \right\|_F^2 + \alpha \left\| (M\Phi_p)^\top W \right\|_F^2 \tag{5.9}$$

It learns both the optimal linear transformation M and the parameters W in decision functions simultaneously.

5.3.2.1. Computing W^*

First, we show that the optimal W^* in the optimization problem (5.9) can be expressed in term of M.

Proposition 2. For a fixed M, the optimal W^* that solves the optimization problem (5.9) is

$$W^* = (M\Phi_p(\alpha I + X_S X_S^\top)\Phi_p^\top M^\top)^{-1} M\Phi_p X_S Y \tag{5.10}$$

Proof. As shown in the problem (5.9), we can expand the objective function in terms of W and M as follows:

$$\begin{aligned} & \left\|Y^\top - W^\top M\Phi_p X_S\right\|_F^2 + \alpha \left\|W\Phi_p^\top M^\top\right\|_F^2 \\ = \; & \mathrm{tr}(Y^\top - W^\top M\Phi_p X_S)(Y^\top - W^\top M\Phi_p X_S)^\top \\ & +\alpha \mathrm{tr}(W^\top M\Phi_p\Phi_p^\top M^\top W) \\ = \; & \mathrm{tr}(W^\top(M\Phi_p(X_S X_S^\top + \alpha I)\Phi_p^\top M^\top)W - 2(Y^\top X_S^\top \Phi_p^\top M^\top)W) \\ & +tr(Y^\top Y) \end{aligned} \tag{5.11}$$

Taking the derivation of Equation 5.11 with respect to W and according to Lagrange Condition, we have:

$$W = (M\Phi_p(\alpha I + X_S X_S^\top)\Phi_p^\top M^\top)^{-1} M\Phi_p X_S Y \tag{5.12}$$

This completes the proof. □

5.3.2.2. Computing M^*

When we get the current optimal W, and we can replace W as the right part in Equation 5.12, then it is easy to verify that the original optimization problem (5.9) is equivalent to the following problem after ignoring some constant items:

$$\max_M \; \mathrm{tr}((M\Phi_p(X_S X_S^\top + \alpha I)\Phi_p^\top M^\top)^{-1} M\Phi_p X_S Y Y^\top X_S^\top \Phi_p^\top M^\top) \tag{5.13}$$

Denote $A = \Phi_p(X_S X_S^\top + \alpha I)\Phi_p^\top$ and let $B = \Phi_p X_S Y Y^\top X_S^\top \Phi_p^\top$. From the definition of Φ_p in the last section, we can see that $\Phi_p(X_S X_S^\top)\Phi_p^\top$ is non-singular, so A is invertible, then problem (5.13) can be rewritten as follows:

$$\max_M \; \mathrm{tr}((MAM^\top)^{-1}(MBM^\top)) \tag{5.14}$$

It is the well known generalized Rayleigh quotient optimization problem, and the solution M^* is spanned by the generalized eigenvectors of $A^{-1}B$ [30].

5.3.3. Overall Algorithm

Combining all the derivations in Section 5.3.1 and Section 5.3.2 corresponding to the optimization strategy, we develop our Location and Scatter Matching (LSM) domain adaptation algorithm depicted in Figure 5.2. In this algorithm, the firstly learn a linear transformation Θ_p which can decrease the distribution gap measured by the second-order statistics parameterized by a non-singular matrix M. Then in the second step, we will take use of the label information to learn the optimal M which can leads minimal loss on the labeled data in the source domain. Finally in the transformed space by $\Theta_p M$, we use SVM for the final classifier training.

Input: labeled patterns $\{(x_i, y_i)\}_{i=1}^{n_1}$ in D_S;
unlabeled patterns $\{(x_i')\}_{i=1}^{n_2}$ in D_T;
feature subspace dimension number p and tradeoff coefficient λ.

Output: The optimal projection matrix Θ for feature subspace, and the prediction label for the data in D_T

1 Calculate the sample mean and sample covariance matrix (u_S, Σ_S) and (u_T, Σ_T) respectively, and map them into A_S and A_T according to Proposition 1.

2 Compute Φ as stated in the Theorem 3 and construct Θ_p which is spanned by the first p rows of Φ.

3 Compute M by solving SVD in Equation 5.14.

4 Project the original data into the new feature space by $\Theta = M\Phi_p$.

5 Use SVM to do the training and testing on the projected data.

Figure 5.2: The outline of our location and scatter matching (LSM) algorithm for Domain Adaptation

5.3.4. Relation to Linear Discriminative Analysis (LDA)

Linear Discriminative Analysis [30] is a classical statistical learning approach for feature extraction, which tries to compute an optimal transformation to project the original data into the new feature space, where the within-class distance is minimized but between-class distance is maximized. Based on the labeled data in D_S, with the empirical mean u_S being 0, we can compute the within-class scatter and total scatter following [67]:

$$S_b = X_S L L^\top X_S^\top, \quad S_t = X_S X_S^\top + \sigma I_d \tag{5.15}$$

where $L = [L_1, ..., L_m]$, and $L_j = Y_j / \sqrt{\Omega_j}$. Ω_j is the instance number of the j-th class. σ is the regularization coefficient avoiding the scatter matrix singular. Then the objective function is:

$$\max_{\hat{\Theta}} \operatorname{tr}((\hat{\Theta} S_t \hat{\Theta}^\top)^+ \hat{\Theta} S_b \hat{\Theta}^\top) \tag{5.16}$$

Comparing the optimization problem in (5.13) derived from ERM framework and (5.16) from LDA, we can observe that regardless of the regularization coefficient difference (σ and α) and label matrix scale (Y and L), the major difference is that we have learned Φ_p in (5.13), then the solution space of $\hat{\Theta}$ is larger than Θ. However, we sacrifice this optimal solution in the source domain D_S to extract the feature space parameterized by Φ_p where the distribution gap between D_S and D_T is closer. Then the generality of the trained model in D_S should be increased in D_T.

5.4. Experiments

5.4.1. Experiment Setup

We have conducted extensive experiments on several datasets to demonstrate the effectiveness of our approach. The datasets are the same as the ones used in Chapter 3, namely, the 20-Newsgroup dataset, Reuters-21578 dataset, and the

online job advertisement dataset. The detailed description of those datasets can be found in Section 3.6, Chapter 3.

We compare our method with an existing method, which is a famous domain adaptation methods known as Kernel Mean Matching (KMM) [37]. Since this method can only support binary classification approaches, multiple-class datasets, namely, NG{4-9} in the 20-Newsgroup dataset, and the online job advertisements dataset, will be transformed into 1-vs-rest binary classification problems. We also compare with our method LRSC presented in Chapter 3 with its linear form. Because LSM is based on linear transformation, it is better to conduct the comparison both in the linear case. In our LSM method, we set the value of the parameter λ, which controls the contribution of the second-order statistics, to 10. We adopt the precision, recall, and F-measure as the evaluation metrics. Precision is defined as the number of instances that are correctly classified by the system divided by the total number of instances that are classified by the system in each class; recall is defined as the number of instances that are correctly classified by the system divided by the actual number of instance in each class. F1-measure is defined as the harmonic mean of precision and recall with equal weight.

5.4.2. Results and Discussion

Table 5.2 summarizes the average domain adaptation performance of different methods on all the datasets. Our approaches LSRC and LSM achieve very promising result comparing with the existing domain adaptation method. Take the 20-Newsgroup dataset for example, the average precision, recall, and F-measure of our approach LSM are 0.786, 0.773, and 0.772 respectively. LSRC can achieve an average precision, recall, and F-measure of 0.832, 0.703 and 0.755 respectively. KMM achieves an average precision, recall, and F-measure of 0.760, 0.659, and 0.687 respectively. The major limitation of KMM is that it just consider the first-order statistics and cannot well generalize the model. Besides, KMM separates the training precess as two steps, which cannot globally take use

of the label information. However, our approach LSM considers the second-order statistics between the source and target domains, leading to a better generalization of our model. The results on the Reuters-21578 and online job advertisement datasets are similar to that on the 20-Newsgroup dataset. It can be observed that our approaches LRSC and LSM can outperform KMM on all the datasets significantly.

Through comparing with KMM, we can conclude that considering second-order statistics using covariance matrix match can obviously improve the domain adaptation performance. However, comparing with the other our proposed approached LRSC presented in Chapter 2, we can observe, LRSC can get comparative performance with LSM as shown for 20-Newsgroup and online job advertisement dataset, or it can even outperform LSM on the Reuters dataset, which indicates that jointly consider the label information and matching the distribution difference can greatly improve the domain adaptation results.

5.4.3. Experimental Parameter Investigation

We have conducted analysis on the effect of the number of selected features in the new feature subspace in our approach. In this analysis, we have carried out several runs of experiments with an increasing number of selected features. Fig. 5.3 illustrates the effect of different number of selected features in the job advertisement datasets. It can be observed that the performance increases with the number of features at the beginning. The reason is that more information can be obtained from additional features for prediction when the number of features increases. Our approach achieves similar or slightly lower performance with the number of selected features after an optimal number has been reached. The small drop is probably due to the increase in the distribution difference between the source domain and the target domain as the increase in the number of features. We have conducted the same analysis on the 20-Newsgroup dataset and the Reuters-21578 dataset. Similar observation has been found. For the sensitivity analysis on the trader-off coefficient λ, when $\lambda = 1$, our proposed

divergence metric is an approximate Jensen-Shannon divergence as proved in Theorem 2. With the increase of λ, the divergence metric will rely more on the first-order statistics until there exists an optimal λ which can combine the empirical mean and covariance well to characterize the distribution discrepancy so as to get the best performance. If λ tends to infinity, our proposed divergence metric converges to the empirical mean discrepancy.

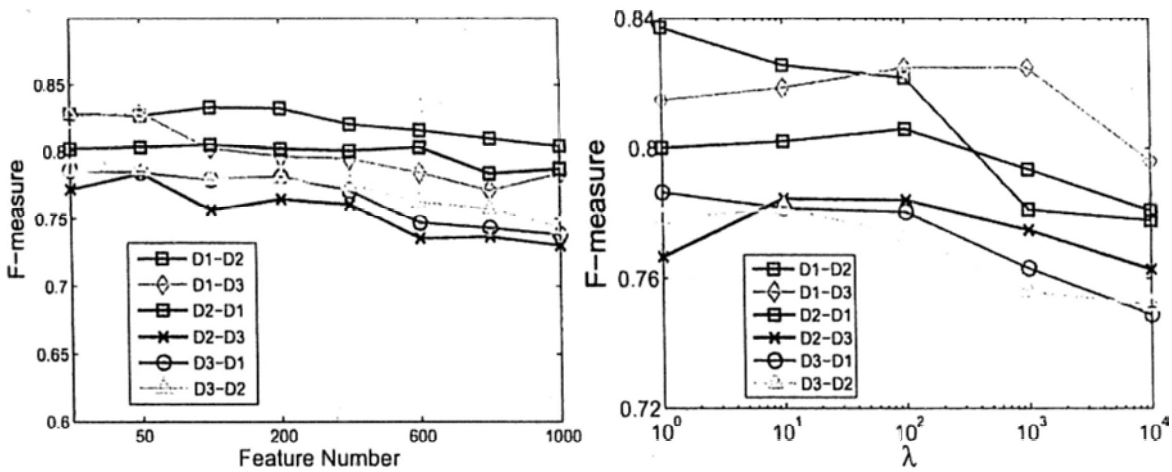

Figure 5.3: The analysis on the effect of the number of features in the new feature subspace (left) and the trade-off coefficient λ (right) using our approach on the online job advertisement dataset. Y-axis in both sub-figures refer to the average F-measure obtained. X-axis in the left and the right sub-figure refers to the number of features and λ respectively. D1, D2, and D3 refers to the domain Accounting, Logistics, and Health respectively. Di-Dj refers to the setting where Di is the source domain and Dj is the target domain.

Table 5.2: The domain adaptation performance in different sets of experiments. NG{1-9} are datasets obtained from the 20-Newsgroup dataset for document classification; People-Place, Place-Org, and Or-People are data datasets obtained from the Reuters-21578 dataset for document classification. P, R, and F refer to the precision, recall, and F-measure respectively.

Data set	KMM			LRSC			LSM		
	P	R	F	P	R	F	P	R	F
NG1	0.911	0.909	0.909	0.952	0.931	0.945	0.962	0.961	**0.961**
NG2	0.781	0.780	0.779	0.862	0.827	0.847	0.847	0.847	**0.847**
NG3	0.695	0.666	0.653	0.817	0.835	**0.824**	0.753	0.805	0.778
NG4	0.747	0.749	0.747	0.775	0.793	0.780	0.803	0.808	**0.803**
NG5	0.702	0.673	0.672	0.753	0.732	**0.740**	0.726	0.713	0.713
NG6	0.737	0.607	0.620	0.683	0.675	0.667	0.698	0.734	**0.697**
NG7	0.748	0.658	0.679	0.768	0.664	0.691	0.777	0.720	**0.740**
NG8	0.808	0.432	0.563	0.813	0.588	0.662	0.752	0.695	**0.717**
NG9	0.709	0.462	0.559	0.746	0.592	0.643	0.762	0.672	**0.695**
Average	0.760	0.659	0.687	0.832	0.703	0.755	0.786	0.773	**0.772**
People-Place	0.752	0.747	0.744	0.828	0.820	**0.824**	0.811	0.812	0.811
Org-Place	0.739	0.717	0.720	0.813	0.832	**0.820**	0.751	0.749	0.750
Org-People	0.545	0.545	0.545	0.732	0.668	**0.696**	0.669	0.672	0.670
Average	0.678	0.670	0.670	0.782	0.781	**0.780**	0.744	0.744	0.744

Table 5.3: The extraction performance of different sets of experiments on online job advertisement dataset. P, R, and F refer to the precision, recall, and F-measure respectively.

Data Set		KMM			LRSC			LSM		
D_S	D_T	P	R	F	P	R	F	P	R	F
D1	D2	0.895	0.680	0.735	0.814	0.845	0.825	0.834	0.849	**0.837**
D1	D3	0.915	0.671	0.722	0.813	0.804	0.800	0.850	0.839	**0.835**
D2	D1	0.914	0.766	**0.816**	0.866	0.789	0.807	0.853	0.802	0.814
D2	D3	0.741	0.882	**0.796**	0.830	0.762	0.765	0.850	0.787	0.788
D3	D1	0.784	0.798	0.781	0.790	0.789	0.779	0.806	0.809	**0.794**
D3	D2	0.742	0.752	0.747	0.793	0.791	0.786	0.782	0.809	**0.786**
Average		0.832	0.758	0.766	0.820	0.800	0.799	0.829	0.816	**0.809**

CHAPTER 6

Conclusions and Future Works

In this thesis, we have developed two novel methods based on discovering shared concept space for domain adaptation in text mining problems. The first method is to learn a low-rank shared concept (LRSC) space with respect to two criteria simultaneously: the empirical loss in the source domain, and the embedded distribution gap between the source domain and the target domain. Besides, our model can transfer the predictive power from the extracted common features to the characteristic features in the target domain by the feature graph Laplacian. Moreover, we can kernelize our proposed method in the Reproducing Kernel Hilbert Space (RKHS) so as to generalize our model by making use of the powerful kernel functions. We theoretically analyze the expected error evaluated by common convex loss functions in the target domain under the empirical risk minimization framework, showing that the error bound can be controlled by the expected loss in the source domain, and the embedded distribution gap.

The second method is another new domain adaptation method called Location and Scatter Matching (LSM) based on our proposed distribution gap measure called symmetric Stein's loss (SSL). The SSL measure is developed based on second order statistics, which combines the mean and covariance discrepancy into a unified Bregman matrix divergence of which Jensen-Shannon divergence between normal distributions is a particular case. The target of LSM is to find a good feature representation which can reduce the embedded distribution gap

measured by SSL between the source domain and the target domain, at the same time, ensure the new derived representation can encode sufficient discriminants with respect to the label information. Then a standard machine learning algorithm, such as Support Vector Machine (SVM), can be adapted to train classifiers in the new feature subspace across domains.

We conduct experiments comparing our two proposed domain adaptation approaches with other existing approaches. The results show that both LRSC and LSM can significantly improve over existing domain adaptation approaches which only use the first order statistics to measure the distribution gap, or consider the distribution matching and model training separately.

In the future, we will investigate how the prior domain knowledge can be considered in our framework. Exploration of domain knowledge, such as structure information, for extracting more discriminative concepts is a possibility. Another direction is to extract discriminative concepts in multiple source domain adaptation problems.

Although the two-sample test on synthetic datasets and real-world datasets demonstrate the advantages of our proposed distribution gap measure, a thorough examination of its merit is needed, such as the theoretical analysis on the convergence. Furthermore, we will also investigate how to incorporate higher order information rather than second order to measure the distribution gap more accurately.

BIBLIOGRAPHY

[1] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[2] Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. In *Workshops of the 7th IEEE International Conference on Data Mining*, pages 77 82, 2007.

[3] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 12:2399–2434, 2006.

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151 175, 2010.

[5] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, pages 137–144, 2006.

[6] Steffen Bickel, Christoph Sawade, and Tobias Scheffer. Transfer learning by distribution matching for targeted advertising. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, pages 145–152, 2008.

[7] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, 2007.

[8] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 187–205, 2007.

[9] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 120–128, 2006.

[10] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alexander J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. In *Proceedings of the 14th annual international conference on Intelligent Systems for Molecular Biology (Supplement of Bioinformatics)*, pages 49–57, 2006.

[11] Yee Seng Chan and Hwee Tou Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

[12] Yee Seng Chan and Hwee Tou Ng. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[13] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press.

[14] Ciprian Chelba and Alex Acero. Adaptation of maximum entropy capitalizer: Little data can help a lo. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 285–292, 2004.

[15] Bo Chen and Wai Lam. Generalized brownian distance covariance and its application to feature selection. Working paper.

[16] Bo Chen, Wai Lam, Ivor Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 179–188, 2009.

[17] Bo Chen, Wai Lam, Ivor W. Tsang, and Tak-Lam Wong. Discovering low-rank shared concept space for adapting text mining models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. submitted.

[18] Bo Chen, Wai Lam, Ivor W. Tsang, and Tak-Lam Wong. A semi-supervised framework for feature mapping and multiclass classification. In *Proceedings of the 9th SIAM International Conference on Data Mining*, pages 341–352, 2009.

[19] Bo Chen, Wai Lam, Ivor W. Tsang, and Tak-Lam Wong. Location and scatter matching for dataset shift in text mining. In *Proceedings of the 10th IEEE International Conference on Data Mining*, 2010.

[20] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 210–219, 2007.

[21] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classifiers for text classification. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 540–545, 2007.

[22] Georges A. Darbellay and Igor Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, 1999.

[23] Hal Daumé, III. Bayesian multitask learning with latent hierarchies. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 135 142, 2009.

[24] H. Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263, June 2007.

[25] Jason V. Davis and Inderjit S. Dhillon. Differential entropic clustering of multivariate gaussians. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, pages 337 344, 2006.

[26] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109 117, 2004.

[27] Wei Fan, Ian Davidson, Bianca Zadrozny, and Philip S. Yu. An improved categorization of classifier's sensitivity on sample selection bias. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 605 608, 2005.

[28] Jenny Rose Finkel and Christopher D. Manning. Hierarchical bayesian domain adaptation. In *HLT-NAACL*, pages 602–610, 2009.

[29] Jerome H. Friedman and Lawrence C. Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, 7(4):697 717, 1979.

[30] Keinosuke Fukunaga. *Introduction to statistical pattern recognition.* Academic Press, 1990.

[31] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1996.

[32] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample problem. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, pages 513–520, 2007.

[33] Peter Hall and Nader Tajvidi. Permutation tests for equality of distributions in high dimensional settings. *Biometrika*, 89(2):359–374, 2002.

[34] Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293 – 325, 1948.

[35] Wassily Hoeffding. A generalized t test and measure of multivariate dispersion. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, page 23 – 41, 1951.

[36] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[37] Jiayuan Huang, Alex Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, pages 601–608, 2007.

[38] Willard James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 360–380, 1961.

[39] Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. Extracting shared subspace for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 381–389, 2008.

[40] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 264–271, 2007.

[41] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms.* Kluwer/Springer, 2002.

[42] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB*, pages 180–191, 2004.

[43] Xiao Li and Jeff Bilmes. A bayesian divergence prior for classifier adaptation. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 2007.

[44] Xiao Ling, Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Spectral domain-transfer learning. In *Proceedings of the 14th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 488 496, 2008.

[45] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, pages 1041–1048, 2008.

[46] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. 2009.

[47] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3):103 134, 2000.

[48] Sinno Jialin Pan, James T. Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI conference on Artificial Intelligence*, pages 677 682, 2008.

[49] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

[50] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT press, 2009.

[51] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th Annual International Conference on Machine Learning*, pages 759–766, 2007.

[52] Ryan M. Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101 141, 2004.

[53] Sandeepkumar Satpal and Sunita Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *Proceedings of European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 224–235, 2007.

[54] Bernhard Schölkopf and Alex J. Smola. *Learning with Kernels: Support Vector Machines,Regularization, Optimization and Beyond*. MIT Press, 2002.

[55] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[56] Amos Storkey and Masashi Sugiyama. Mixture regression for covariate shift. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, pages 1337–1344, 2007.

[57] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, 2008.

[58] Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Information Science and Statistics. Springer-Verlag, 2000.

[59] Qing Wang, Sanjeev R. Kulkarni, and Sergio Verdú. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Transactions on Information Theory*, 51(9):3064 3074, 2005.

[60] Patrick Warnat, Roland Eils, and Benedikt Brors. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, 6(265), 2005.

[61] Kilian Q. Weinberger, Anirban Dasgupta, John Langford, Alexander J. Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *the 26th Annual International Conference on Machine Learning,*, pages 1113–1120, 2009.

[62] Tak-Lam Wong, Wai Lam, and Bo Chen. Mining employment market via text block detection and adaptive cross-domain information extraction. In *Proceedings of the 32st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 283–290, 2009.

[63] Dikan Xing, Wenyuan Dai, Gui-Rong Xue, and Yong Yu. Bridged refinement for transfer learning. In *Proceedings of European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 324 335, 2007.

[64] Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 627–634, 2008.

[65] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35 63, 2007.

[66] Jun Yang, Rong Yan, and Alexander G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM International Conference on Multimedia*, pages 188–197, 2007.

[67] Jieping Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, 6:483–502, 2005.

[68] Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceeding of the Twenty-Second International Conference on Machine Learning*, pages 1012–1019, 2005.

[69] Xiaofeng Yu, Wai Lam, and Bo Chen. An integrated discriminative probabilistic approach to information extraction. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 325–334, 2009.

[70] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21st Annual International Conference on Machine Learning*, pages 903–910, 2004.

[71] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21th Annual International Conference on Machine Learning*, pages 919–926, 2004.

[72] Erheng Zhong, Wei Fan, Jing Peng, Kun Zhang, Jiangtao Ren, Deepak S. Turaga, and Olivier Verscheure. Cross domain distribution adaptation via kernel mapping. In *Proceedings of the 15th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 1027–1036, 2009.

[73] Xiaojin Zhu. Semi-supervised learning literature survey. In *Technical report, Carnegie Mellon University*, 2005.