$N°$ d'Ordre : D.U. ...
EDSPIC : ...

Paris 13 University - Sorbonne Paris Cité

Computer Science Laboratory of Paris-North (LIPN)

UMR 7030, CNRS

Thesis

presented by

Redko Ievgen

for the degree of

DOCTOR OF COMPUTER SCIENCE

Nonnegative Matrix Factorization for Transfer learning

approved on 26th November, 2015 by the committee composed of:

Thesis supervisor:	
Younès BENNANI	Professor, Université Paris 13
Chair:	
Patrick GALLINARI	Professor, Université Paris 6
Examining committee :	
Marc SEBBAN	Professor, Université Jean Monnet
Stéphane CANU	Professor, INSA de Rouen
Christophe FOUQUERÉ	Professor, Université Paris 13
Vincent LEMAIRE	Researcher(HDR), Orange Labs
Basarab MATEI	Associate Professor(HDR), Université Paris 13

$N°$ d'Ordre : D.U. ...
EDSPIC : ...

Université Paris 13 - Sorbonne Paris Cité

Laboratoire d'Informatique de Paris Nord (LIPN)

UMR 7030 du CNRS

Thèse

présentée par

Ievgen REDKO

pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ PARIS 13

Spécialité: INFORMATIQUE

Factorisation matricielle non-négative pour l'apprentissage par transfert

Soutenue le 26 novembre 2015 devant le jury composé de :

Directeur de thèse :	
Younès BENNANI	Professeur, Université Paris 13
Président :	
Patrick GALLINARI	Professeur, Université Paris 6
Rapporteurs :	
Marc SEBBAN	Professeur, Université Jean Monnet
Stéphane CANU	Professeur, INSA de Rouen
Examinateurs :	
Christophe FOUQUERÉ	Professeur, Université Paris 13
Vincent LEMAIRE	Chercheur (HDR), Orange Labs
Basarab MATEI	Maître de conférences (HDR), Université Paris 13

I would like to dedicate this thesis to my loving family.

Acknowledgements

First of all, I would like to express my gratitude to Prof. Younès Bennani for his permanent guidance during these three years, for his interesting suggestions, fruitful ideas and being a very easy, intelligent and at the same time understandable person. His contribution to this thesis cannot be underestimated.

I want to thank Prof. Marc Sebban and Prof. Stéphane Canu who have reported my thesis and for theirs constructive remarks. I thank also Prof. Christophe Fouqueré, Prof. Patrick Gallinari, Dr. Basarab Matei and Dr. Vincent Lemaire for accepting to be part of my thesis jury and for their interesting and relevant comments.

I would also like to thank the Computer Science Laboratory of Paris North University (LIPN), especially the Machine Learning and Applications Team (A3) for their help and support.

I am also grateful to several other researchers and students who encouraged me during my work. Dr. Taras Shalaiko who helped me a lot when I was feeling desperate finding a solution not only as a brilliant mathematician but also as a friend. Dr. Nistor Grozavu and Dr. Basarab Matei who were always open for a discussion and provided me with some interesting insights about my work. Finally, I express my deepest appreciation and gratitude to my friend Kostiantyn Klekota who was always there and ready to help no matter what.

I also appreciate a lot the help and support provided by people outside: Oksana Tovstolytkina for her cooking and constantly positive mood; my brother's wife Viktoria and her family, Pavlo Voitsekhivsky, Ievgen Ovcharenko, Olesia Burak, Tamara Kiziria, Ilona Matiash, Vitaliy Fedoseev and his wife Elena for knowing that there are friends waiting for me when I

come back home; Kamel Satouri, Ariane Bachelet and Élise Blouri for being my climbing mates and good friends.

I am thankful to all the people that I was sharing my office with - for the laughs, excellent working environment and hospitality.

Let me reserve my final appreciation to my father Viktor, my mother Nataliia and my brother Oleksandr. Without the nurturing, care and love from them, I definitely could not have completed my doctoral degree.

Résumé

L'apprentissage par transfert consiste à utiliser un jeu de tâches pour influencer l'apprentissage et améliorer les performances sur une autre tâche. Cependant, ce paradigme d'apprentissage peut en réalité gêner les performances si les tâches (sources et cibles) sont trop dissemblables. Un défi pour l'apprentissage par transfert est donc de développer des approches qui détectent et évitent le transfert négatif des connaissances utilisant très peu d'informations sur la tâche cible. Un cas particulier de ce type d'apprentissage est l'adaptation de domaine. C'est une situation où les tâches sources et cibles sont identiques mais dans des domaines différents. Dans cette thèse, nous proposons des approches adaptatives basées sur la factorisation matricielle non-négative permettant ainsi de trouver une représentation adéquate des données pour ce type d'apprentissage. En effet, une représentation utile rend généralement la structure latente dans les données explicite, et réduit souvent la dimensionnalité des données afin que d'autres methods de calcul puissent être appliquées. Nos contributions dans cette these s'articulent autour de deux dimensions complémentaires : théorique et pratique.

Tout d'abord, nous avons proposé deux méthodes différentes pour résoudre le problème de l'apprentissage par transfert non supervisé basé sur des techniques de factorisation matricielle non-négative. La première méthode utilise une procédure d'optimisation itérative qui vise à aligner les matrices de noyaux calculées sur les bases des données provenant de deux tâches. La seconde représente une approche linéaire qui tente de découvrir un plongement pour les deux tâches minimisant la distance entre les distributions de probabilité correspondantes, tout en préservant la propriété de positivité.

Nous avons également proposé un cadre théorique basé sur les plongements Hilbert-Schmidt. Cela nous permet d'améliorer les résultats théoriques de l'adaptation au domaine, en introduisant une mesure de distance naturelle et intuitive avec de fortes garanties de calcul pour son estimation. Les résultats proposés combinent l'étanchéité des bornes de la théorie d'apprentissage de Rademacher tout en assurant l'estimation efficace de ses facteurs clés.

Les contributions théoriques et algorithmiques proposées ont été évaluées sur un ensemble de données de référence dans le domaine avec des résultats prometteurs.

Abstract

The ability of a human being to extrapolate previously gained knowledge to other domains inspired a new family of methods in machine learning called transfer learning. Transfer learning is often based on the assumption that objects in both target and source domains share some common feature and/or data space. If this assumption is false, most of transfer learning algorithms are likely to fail. In this thesis we propose to investigate the problem of transfer learning from both theoretical and applicational points of view.

First, we present two different methods to solve the problem of unsupervised transfer learning based on Non-negative matrix factorization techniques. First one proceeds using an iterative optimization procedure that aims at aligning the kernel matrices calculated based on the data from two tasks. Second one represents a linear approach that aims at discovering an embedding for two tasks that decreases the distance between the corresponding probability distributions while preserving the non-negativity property.

We also introduce a theoretical framework based on the Hilbert-Schmidt embeddings that allows us to improve the current state-of-the-art theoretical results on transfer learning by introducing a natural and intuitive distance measure with strong computational guarantees for its estimation. The proposed results combine the tightness of data-dependent bounds derived from Rademacher learning theory while ensuring the efficient estimation of its key factors.

Both theoretical contributions and the proposed methods were evaluated on a benchmark computer vision data set with promising results. Finally,

we believe that the research direction chosen in this thesis may have fruitful implications in the nearest future.

Contents

List of Figures

List of Tables

Avant Propos

L'apprentissage par transfert est le processus par lequel un individu utilise un apprentissage acquis dans une situation pour l'appliquer à une autre situation. Le transfert est la capacité à utiliser nos expériences antérieures dans de nouveaux apprentissages.

Ce paradigme d'apprentissage par transfert, consiste donc à utiliser un jeu de tâches pour influencer l'apprentissage et améliorer les performances sur une autre tâche. Cependant, l'apprentissage par transfert peut en réalité gêner les performances si les tâches sont trop dissemblables. Un défi pour l'apprentissage par transfert est donc de développer des approches qui détectent et évitent le transfert négatif des connaissances utilisant très peu d'informations sur la tâche cible. Dans cette thèse nous nous intéressons aussi à un cas particulier de l'apprentissage par transfert : l'adaptation de domaine. C'est une situation où les tâches sources et cibles sont identiques mais dans des domaines différents.

L'apprentissage par transfert implique deux problèmes corrélés, ayant comme but l'utilisation de la connaissance acquise sur un jeu de tâches et améliorer les performances pour une autre tâche liée. Particulièrement, l'apprentissage par transfert d'une certaine tâche cible - la tâche sur laquelle les performances sont mesurées - est très dépendant de l'apprentissage d'un ou des tâches auxiliaires. Par exemple, les athlètes se servent de l'apprentissage par transfert quand ils pratiquent des activités auxiliaires afin de s'améliorer dans leur activité principale plus compétitive.

L'apprentissage par transfert est un processus cognitif qui peut avoir des effets positifs ou négatifs sur les conduites à venir. Autrement dit : facilitation d'un apprentissage en fonction d'un apprentissage antérieur. Il y a trois catégories typiques :

- Le transfert bilatéral : la capacité de faire avec une main ce qui a été appris avec l'autre. Exemple : un jongleur qui apprend à jongler par la main gauche aura par la suite plus de facilité à apprendre à jongler par la main droite.

- Apprendre à apprendre : d'une manière générale plus on apprend une tache d'un

même types plus vite on apprend, c'est ce qu'on retrouve dans la vie courante, le premier jeu de vidéo facilitera les suivants.

- Le transfert négatif : en effet un premier apprentissage peut gêner le suivant, par exemple si on a appris à taper sur un clavier d'ordinateur à deux doigts on aura du mal par la suite à apprendre à taper avec les dix doigts.

Le défi clé de l'apprentissage par transfert est d'identifier quelle connaissance doit être transférée et comment ?

Par ailleurs, un problème fondamental dans de nombreuses tâches en apprentissage artificiel est de trouver une représentation adéquate des données. Une représentation utile rend généralement la structure latente dans les données explicite, et réduit souvent la dimensionnalité des données afin que d'autres méthodes de calcul puissent être appliquées. La factorisation matricielle est une approche couramment utilisée pour la compréhension de la structure latente de la matrice observée des données pour diverses applications. Ces méthodes matricielles ont suscité récemment une attention croissante en raison de leur élégance mathématique et les résultats empiriques encourageants pour une variété d'applications. L'objectif de cette thèse, est donc de développer et d'étudier des méthodes de factorisation matricielle pour trouver une représentation adéquate des données dans le cadre de l'apprentissage par transfert et l'adaptation au domaine, d'identifier quelle connaissance doit être transférée et comment ? et d'exhiber les avantages et les inconvénients de ce paradigme d'apprentissage automatique avec des illustrations sur des données réelles.

Cette thèse est organisée en cinq principaux chapitres encadrés par une introduction et des conclusions et annexes. Le contenu de chaque chapitre est résumé ci-dessous :

Chapitre 2. Dans ce chapitre, nous introduisons les notions de base relatives à la famille des méthodes d'apprentissage artificiel appelée la factorisation matricielle non-négative (Non-negative Matrix Factorization: NMF). Nous décrivons les extensions et modifications du modèle de base de la NMF et donnons une motivation pour leur application éventuelle dans le contexte de la classification automatique (clustering). Nous présentons également les règles de mise à jour multiplicatives pour chaque algorithme NMF et nous montrons comment on peut prouver théoriquement qu'ils convergent vers un optimum local. Enfin, nous donnons deux exemples complets qui démontrent clairement la signification de chaque facteur découlant de la NMF sur deux ensembles

de données.

Chapitre 3. Dans ce chapitre, nous donnons d'abord une définition du problème de l'apprentissage par transfert et expliquons comment nous classons les méthodes d'apprentissage par transfert. Après la catégorisation proposée, nous décrivons les méthodes d'état de l'art et les résultats théoriques qui ont été proposés par différents chercheurs pour résoudre ce problème. Nous présentons également dans une section distincte la description détaillée des méthodes d'apprentissage par transfert qui se basent sur la NMF. Enfin, nous introduisons un ensemble de données que nous utilisons pour l'évaluation des performances des approaches proposées dans les chapitres suivants.

Chapitre 4. Dans ce chapitre, nous proposons une approche d'apprentissage par transfert non supervisé qui minimise de manière itérative la distance entre les distributions de probabilités source et cible en optimisant l'alignement des noyaux (Kernel target alignment) calculés sur les jeux de données initiaux. Nous montrons que cette procédure est adaptée à l'apprentissage par transfert en la rapportant à la maximisation du critère d'indépendance de Hilbert-Schmidt (HSIC) et de l'information mutuelle quadratique (QMI). Nous évaluons notre méthode sur des ensembles de données réelles de référence et montrons qu'elle peut surpasser certaines méthodes d'apprentissage par transfert existantes.

Chapitre 5. Dans ce chapitre, nous présentons une nouvelle méthode pour l'adaptation de domaine non supervisée qui vise à aligner deux domaines (distributions de probabilities) en utilisant un ensemble commun de vecteurs de base dérivés de vecteurs propres de chaque domaine. Nous utilisons des techniques de factorisation matricielle non-négative pour générer un plongement non-négatif qui minimise la distance entre les projections des données source et cible. Nous présentons une justification théorique de notre approche en montrant la cohérence de la fonction de similarité définie en utilisant la projection obtenue. Nous validons notre approche sur des ensembles de données de référence et montrons qu'elle surpasse plusieurs méthodes d'adaptation de domaine.

Chapitre 6. Dans ce chapitre, nous commenons avec une présentation des résultats théoriques pour l'adaptation de domaine. Ces résultats théoriques comprennent les bornes de généralisation de Vapnik-Chervonenkis et celles issues de la théorie de l'apprentissage de Rademacher. Nous présentons des restrictions principales de deux

paradigmes et montrons comment on peut obtenir des bornes plus intéressantes en utilisant les plongements de Hilbert-Schmidt. Les résultats présentés remplacent les mesures de divergence proposées dans des travaux antérieurs par une distance naturelle et intuitive appelée la distance maximale moyenne (Maximum mean discrepancy) qui bénéficie de l'existence d'un estimateur en temps linéaire.

Chapitre 7. Dans ce chapitre, nous résumons les principaux résultats présentés dans cette thèse. Nous discutons également les perspectives d'avenir possibles pour chacune des contributions proposées, incluant des versions multi-sources des méthodes d'apprentissage par transfert non supervisé des deux derniers chapitres et les tests d'hypothèses statistiques pour les contributions théoriques.

Chapter 1

Introduction

Over the past two decades, the majority of research in the area of data mining was concentrated around a task of supervised classification. A large number of techniques has been developed to tackle this problem to be further applied successfully in many real-world applications. What's more, an extensive theoretical study was conducted to prove theoretical guarantees of supervised learning and to show under which conditions it succeeds. A pursuit for even more efficient supervised algorithms has finally led to a human-like performance of Machine Learning in such tasks as automatic speech recognition, image classification and natural language processing. Altogether, this allowed machine learning to become a powerful tool for data analysis that, nowadays, is widely integrated into our lives. For instance, most of the page-ranking algorithms use extensively classification techniques in order to define the relevancy of a given page based on its content, a link structure and a visiting score; web stores provide us with a list of additional items that we would likely want to buy based on our preferences and the overall customers history; finally, the automatic translation systems usually proceed by using samples of translations to find the most relevant match.

In order to have a good performance, these models must be trained on huge amounts of labeled data that represent adequately the underlying probability distribution. In some cases, however, reliably labeled data cannot be collected. This issue has led to a new branch of data mining that was called semi-supervised machine learning. In semi-supervised algorithms, one tries to make use of a large amount of unlabeled data in order to discover general patterns that are used further in combination with small number of labeled instances to build a robust classifier. The vast family of semi-supervised

algorithms has found its application in image and text classification, areas where it is relatively easy to automatically obtain a large amount of data but hard to label them. Finally, semi-supervised learning mimics well how a human being learns, i.e., a small number of direct instructions are further combined with a large number of unlabeled observations.

However, the algorithms that correspond to both supervised and semi-supervised learning work well only under a common assumption: the training and test data are from the same feature space and the same distribution. When the distribution changes, most statistical models must be rebuilt from new collected data that can be expensive or even impossible for some applications. Therefore it became necessary to develop approaches that reduce the need and the effort of collecting new labeling samples by combining data from related areas to further use them in the learning procedures. This, in its turn, gave rise to a new family of machine learning algorithms called transfer learning. For example, applying a classifier trained on the images from Amazon online merchants to web camera photos could be beneficial but only in case if the shift between domains has been taken into account. Transfer learning involves two interrelated problems, aiming at learning a robust classifier in source domain hoping that it will perform well in the related target domain by reducing the discrepancy between their distributions. Contrary to supervised and semi-supervised paradigms, transfer learning reflects the reality as it shows what actually happens when a system trained under perfect conditions on preprocessed data faces the real-world applications' sample.

Typical example that illustrate the main idea of transfer learning and its difference from traditional machine learning is sentiment classification. In general, the task of sentiment classification consists in distinguishing between positive and negative reviews based on the corresponding characteristic words. These characteristic words, however, are usually domain dependent, i.e., for a positive review written for a book we would expect to see words like "intriguing", "outstanding" and "interesting" while a positive review on a shoes brand would rather contain words like "comfortable", "cozy" and "soft". Obviously, applying a classifier learned on book reviews directly to shoes brand reviews will lead to a poor classification accuracy. In this case, transfer learning algorithms can be used in order to match the corresponding terms from both domains.

Another important motivation for ongoing interest in transfer learning is its potential to become a vital tool for building human-like Artificial Intelligence systems. Contrary to traditional machine learning, transfer learning gives learning systems an ability to generalize knowledge across different domains and thus, to assure the autonomous behavior of a learning system in time. All these factors contribute to current high research interest in transfer learning that is confirmed by numerous workshops and dedicated tutorials at top machine learning venues.

Thesis structure

The rest of this thesis is organized into 7 chapters. The main contents of each chapter are summarized below:

Chapter 2. In this chapter, we introduce basis notions related to the family of machine learning methods called Non-negative matrix factorization (NMF). We describe the extensions and modifications of the simplest form of NMF and give a motivation for their eventual application in context of clustering. We also present multiplicative update rules for each NMF algorithm and show how one can prove theoretically that they converge to a local optima. Finally, we give two comprehensive examples that demonstrate clearly the meaning of each factor arising from NMF on both artificial and real-world data sets.

Chapter 3. In this chapter, we first give a definition of the transfer learning problem and explain how we categorize the transfer learning methods. Following the proposed categorization, we describe the state-of-the-art methods and theoretical results that were proposed by machine learning scientists to tackle this problem. We also present in a separate section the detailed description of the state-of-the-art transfer learning methods that make use of NMF to perform the transfer of knowledge. Finally, we introduce a benchmark computer vision data set that we use for algorithm's performance evaluation further in this thesis.

Chapter 4. In this chapter, we propose a simple and intuitive unsupervised transfer learning approach that minimizes iteratively the distance between source and target task distributions by optimizing the kernel target alignment (KTA). We show that this procedure is suitable for transfer learning by relating it to Hilbert-Schmidt Independence Criterion (HSIC) and Quadratic Mutual Information (QMI) maximization. We

run our method on benchmark computer vision data sets and show that it can outperform some state-of-the-art transfer learning methods.

Chapter 5. In this chapter, we present a new method for fully unsupervised domain adaptation that seeks to align two domains using a shared set of basis vectors derived from eigenvectors of each domain. We use non-negative matrix factorization (NMF) techniques to generate a non-negative embedding that minimizes the distance between projections of source and target data. We present a theoretical justification for our approach by showing the consistency of the similarity function defined using the obtained embedding. We validate our approach on benchmark data sets and show that it outperforms several state-of-the-art domain adaptation methods.

Chapter 6. In this chapter, we start with a description of theoretical results for a special case of transfer learning called domain adaptation. These theoretical results include Vapnik-Chervonenkis generalization bounds and Rademacher complexity learning bounds. We outline two main restrictions of both paradigms and show how one can combine data-dependent Rademacher bounds with the original ones using Hilbert-Schmidt embeddings of probability functions. The proposed results replace the divergence distances from the prior work on domain adaptation theory by a natural and intuitive Maximum Mean Discrepancy (MMD) distance that enjoys the existence of a linear time estimator for its quadratic empirical counterpart.

Chapter 7. In this chapter, we summarize the main results presented in this thesis. We also discuss possible future perspectives for each of the proposed contributions. They include multi-source versions of the unsupervised transfer learning methods from last two chapters and statistical two-sample test for the third one.

Chapter 2

Learning with Non-negative Matrix Factorization

2.1 Introduction

Clustering is a well-known machine learning technique used for unsupervised classification of patterns (observations, data items or feature vectors) into groups of similar objects. The groups given by a clustering algorithm are called "clusters", each cluster consists of objects that are similar between themselves but different from objects in other clusters. There are three main types of machine learning algorithms:

- supervised learning (when data is labeled in both training and test sets);
- semi-supervised learning (data is labeled only in small training test);
- unsupervised learning (no labeled data available).

Clustering is usually associated with unsupervised learning. Unsupervised learning itself is extremely important setting of machine learning as it occurs in numerous real-world applications. Main reasons that show why unsupervised learning can prove beneficial are:

- labeling a set of objects manually can be hard or even impossible on large amounts of data;
- it can be used to classify a huge amount of unlabeled data to further label it manually;
- it can be used to find a set of variables that can be useful for further categorization.

There exists numerous unsupervised learning methods that were applied in many contexts and by researchers in many disciplines. Typical applications of clustering are: statistics [Arabie, 1996], pattern recognition [Duda et al., 2000], image segmentation and computer vision [Jain et al., 1999], multivariate statistical estimation [Scott, 1992]. Clustering is also widely used for data compression in image processing, which is also known as vector quantization [Gersho and Gray, 1991]. An exhaustive survey about clustering methods can be found in [Han and Kamber, 2000].

There exist lots of well-known clustering algorithms, namely: k-means, mixture models, hierarchical clustering, non-negative matrix factorization etc. Among all the

methods used for clustering we will discuss the one called Non-negative matrix factorization (NMF).

NMF is a group of algorithms that aims to factorize a given matrix into (usually) two matrices where all matrices involved into factorization have no negative elements. This non-negativity makes the resulting matrices easier to interpret. We consider one of the matrices as a matrix containing the prototypes of a data set and the other one as a data partition matrix. Since this optimization problem is not convex in general, it is commonly approximated numerically.

In this chapter, we introduce basic notions related to NMF and describe some of its extensions and modifications. We also present multiplicative update rules for each NMF algorithm and show how one can prove theoretically that they converge to a local optima. Finally, we give two comprehensive examples that demonstrate clearly the meaning of each factor arising from NMF for both artificial and real-world data sets.

2.2 Standard and Semi- NMF

A standard NMF [Lee and Seung, 1999] seeks the following decomposition:

$$X \simeq FG^T, X \in \mathbb{R}^{m \times n}, F \in \mathbb{R}^{m \times k}, G \in \mathbb{R}^{n \times k}$$

$$X, F, G \geq 0,$$

where

- X is an input data matrix;
- columns of F can be considered as basis vectors;
- columns of G are considered as cluster assignments for each data object;
- k is the desired number of clusters.

Standard NMF can be represented as a following optimization problem:

$$\min_{F,G \geq 0} \left\| X - FG^T \right\|_{(\cdot)}^2,$$

where $(\cdot)$ is an arbitrary measure of divergence.

Multiplicative update rules that are usually used to solve NMF related problems were first introduced in [Lee and Seung, 1999]. To ensure the non-negativity of the resulting factors and keeping in mind that initial matrices are also non-negative, multiplicative update rules of NMF can be calculated using the following general approach:

$$Z = Z \circledast \frac{\left[\frac{\partial J}{\partial X}\right]_-}{\left[\frac{\partial J}{\partial Z}\right]_+},$$

where Z represents all the variables involved in the cost function, $\left[\frac{\partial J}{\partial Z}\right]_+$ stands for positive part of gradient of the cost function J and $\left[\frac{\partial J}{\partial Z}\right]_-$ for negative part. This leads to the following update rules:

$$F = F \circledast \frac{XG^T}{FGG^T},$$

$$G = G \circledast \frac{F^T X}{F^T FG}.$$

Here $\circledast$ and $/$ stand for entrywise multiplication and division, respectively.

When the data matrix is unconstrained (i.e., it may have mixed signs), [Ding et al., 2010b] introduced Semi-NMF - a factorization in which we restrict G to be non-negative while placing no restriction on the signs of F. Multiplicative update rules for Semi-NMF have the following form:

$$F = XG(G^T G)^{-1},$$

$$G = G \circledast \sqrt{\frac{(X^T F)^+ + G(F^T F)^-}{(X^T F)^- + G(F^T F)^+}},$$

where $A^+ = \frac{1}{2}(|A| + A)$ and $A^- = \frac{1}{2}(|A| - A)$.

2.3 Convex NMF and Kernel NMF

We usually suppose that "good" features should have a low distortion w.r.t. the initial data as in this case they are assumed to capture the general patterns of the underlying

distribution. To this end, the Convex NMF (C-NMF) was proposed in [Ding et al., 2010b]. To develop C-NMF , we consider the factorization of the following form:

$$X \simeq FG^T = XWG^T, X \in \mathbb{R}^{n \times m}, W \in \mathbb{R}^{m \times k}, G \in \mathbb{R}^{m \times k}$$

$$X, W, G \geq 0,$$

where the column vectors of U lie within the column space of X:

$$F = XW.$$

In this formulation, the authors force basis vectors to represent linear combinations of initial data points weighted based on the columns of W. At the same time, adding a new factor in the NMF model increases the sparsity of the obtained solution.

The natural generalization of C-NMF is Kernel NMF (K-NMF) [Zhang and Chen, 2006]. To "kernelize" C-NMF, we consider a mapping ϕ which maps each vector x_i to a higher dimensional feature space, such that:

$$\phi : X \to \phi(X) = (\phi(x_1), \phi(x_1), ..., \phi(x_n)) \in \mathbb{R}^{n \times m}.$$

We obtain the factorization of the following form:

$$\phi(X) \simeq \phi(X)WG^T, \phi(X) \in \mathbb{R}^{n \times m}, W \in \mathbb{R}^{m \times k}, G \in \mathbb{R}^{m \times k}.$$

Each kernel can be described by its Gram matrix. We call a Gram matrix of a given kernel k some symmetric positive-semidefinite matrix K. Subsequently the kernel is an inner-product function defined as:

$$K = \phi(X)\phi(X)^T,$$

$$\phi(X)\phi(X)^T \simeq \phi(X)\phi(X)^T WG^T,$$

$$\phi(X) \in \mathbb{R}^{n \times m}, W \in \mathbb{R}^{m \times k}, G \in \mathbb{R}^{m \times k}.$$

Finally, K-NMF is is defined as follows:

$$K \simeq KWG^T, K \in \mathbb{R}^{n \times n}, W \in \mathbb{R}^{n \times k}, G \in \mathbb{R}^{n \times k}.$$

An important advantage of K-NMF compared to Standard NMF and C-NMF is that it can deal not only with attribute-value data but also relational data that can be beneficial if the clusters are well-separable in a nonlinear Hilbert space.

Multiplicative update rules for both C-NMF and K-NMF[1] have the following form:

$$W = W \circledast \sqrt{\frac{Y^+G + Y^-WG^TG}{Y^-G + Y^+WG^TG}},$$

$$G = G \circledast \sqrt{\frac{Y^+W + GW^TY^-W}{Y^-W + GW^TY^+W}},$$

where $Y = XX^T$, $Y^+ = \frac{1}{2}(|Y| + Y)$ and $Y^- = \frac{1}{2}(|Y| - Y)$.

2.4 Uni- and Bi-Orthogonal NMF

Different kinds of constraints can be imposed on cluster's properties in order to achieve better clustering results. One of the most common constraints that is used for clustering is orthogonality of subspaces of clusters. Indeed, imposing orthogonality on the subspaces of clusters means that we try to find clusters that are as different as possible. In our case, orthogonality constraints imposed on matrices obtained with NMF is considered to be useful as it results in unique factorization and has a good clustering interpretation.

The idea of Uni- and Bi-Orthogonal NMF was first described in [Ding et al., 2010b] where it was claimed to increase the quality of clustering and provide an unique non-negative matrix factorization (which is rare for this type of matrix factorizations). In [Ding et al., 2010b], authors proposed a novel approach for solving NMF problem with orthogonality constraints and showed that their update rules have a non-increasing property even though there was no robust proof of convergence. In [Mirzal, 2010], authors imposed orthogonality on matrices of Tri-NMF by adding supplementary terms directly into the cost-function instead of solving it as a constrained optimization prob-

[1] Matrix X in case of K-NMF is replaced with a Gram matrix K.

lem (that is the case for [Ding et al., 2010b]). Their approach has a robust convergence proof and it is mainly inspired by [Lin, 2007] but with its further generalization for matrices that have auxiliary constraints with mutually dependency between columns and/or rows.

The Bi-Orthogonal NMF (BONMF) seeks the following decomposition:

$$X \simeq FSG^T,$$

$$X \in \mathbb{R}^{n \times m}, F \in \mathbb{R}^{n \times k}, S \in \mathbb{R}^{k \times l}, G \in \mathbb{R}^{m \times l},$$

$$F^T F = I, G^T G = I, X, F, S, G \geq 0.$$

The multiplicative update rules for matrices F, G and S have the following form:

$$F = F \circledast \frac{XGS^T}{FF^T XGS^T},$$

$$S = G \circledast \frac{F^T XG}{F^T FSG^T G},$$

$$G = G \circledast \frac{X^T FS}{GG^T X^T FS}.$$

The Uni-Orthogonal NMF (UONMF) imposes orthogonality constraint on either columns of F or rows of G. It is clear that this variation is just a special case of BONMF with $S = I$.

The authors of Orthogonal NMF mentioned that the full orthogonality of matrices F and G cannot be achieved using their algorithm because it uses an approximate solution for non diagonal elements of the Lagrange multipliers matrix. So, their solution of this optimization problem does not result in a set of fully orthonormalized vectors.

2.5 Symmetric NMF

The Symmetric NMF (Sym-NMF) [Ding and He, 2005] of the similarity matrix A is formulated as following optimization problem:

$$A \simeq GG^T, A \in \mathbb{R}^{n \times n}, G \in \mathbb{R}^{n \times k}$$

where A is a similarity matrix calculated based on an arbitrary similarity measure, n is a number of objects, k is the number of clusters requested. Compared to NMF, Sym-NMF is more flexible in terms of choosing similarities for the data points. Any similarity measure that well describes the inherent cluster structure can be chosen. In fact, the formulation of NMF can be related to Sym-NMF when $A = X^T X$. This means that NMF implicitly chooses inner products as the similarity measure, which might not be suitable to distinguish different clusters.

Multiplicative update rule for matrix G in Sym-NMF has the following form:

$$G = G \circledast \left(0.5 J_{nk} + 0.5 \frac{XG}{GG^T G}\right),$$

where J_{nk} is an all-ones matrix of size $n \times k$.

2.6 Multilayer NMF

In order to improve performance of the NMF, especially for illconditioned and badly scaled data and also to reduce risk of getting stuck in local minima of a cost function, a simple hierarchical and multistage procedure to perform a sequential decomposition of non-negative matrices was developed in [Cichocki and Zdunek, 2007].

In the first step, a basic decomposition

$$X \simeq F_1 G_1$$

is performed using any available NMF algorithm. In the second stage, the results obtained from the first stage are used to perform the similar decomposition:

$$G_1 \simeq F_2 G_2$$

using the same or different update rules, and so on. The decomposition takes into account only the components obtained at the previous step. The process can be repeated arbitrary many times until some stopping criteria is satisfied. In each step, gradual improvements of the performance are usually obtained. Thus, the Multilayer NMF

(MNMF) is of the following form:

$$X \simeq F_1 F_2 ... F_L G_L,$$

with the basis matrix defined as $F = F_1 F_2 ... F_L$ where $F_1 \in \mathbb{R}^{n \times k}$ and $\{F_i\}_{i=2...L} \in \mathbb{R}^{k \times k}$. Physically, this means that we build up a system that has many layers or cascade connection of L mixing subsystems.

In Appendix A, we present our original contribution on Multilayer NMF.

2.7 Non-increasing property of update rules

Proving that the objective function of a given NMF problem under the proposed update rules is non-increasing can be usually achieved by introducing an auxiliary function similar to Expectation-Maximization algorithm. We now give a definition of an auxiliary function.

Definition 1. $G(h, h')$ is an auxiliary function of $F(h)$ if the conditions

$$G(h, h') \geq F(h),\ G(h, h) = F(h)$$

are satisfied.

Auxiliary function is very useful as it allows to assure the non-increasing property of the update rules based on the following lemma.

Lemma 2.1. *If G is an auxiliary function, then F is non-increasing under the update*

$$h^{t+1} = \arg\min_h G(h; h^t).$$

Proof of this lemma can be found in [Lee and Seung, 1999]. The graphical representation of the concept of an auxiliary function is given in Figure 2.1.

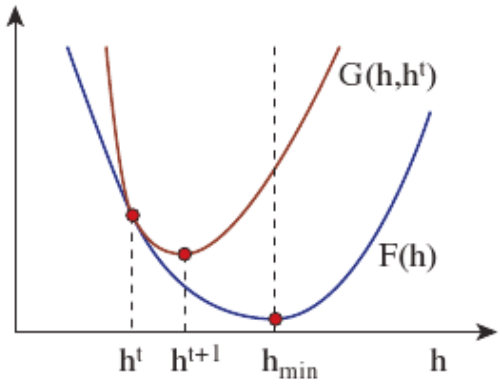


Figure 2.1: Graphical representation of Theorem 2.1 (Figure depicted from [Lee and Seung, 1999]).

2.8 Examples

We will now show the results of NMF in two cases: first, we apply it to a randomly generated data set; second, we use it for selected images from MNIST [LeCun and Cortes, 2010] data set.

Example 1. Let us consider a 7×5 matrix X:

$$X = \begin{pmatrix} 0.1394 & 0.8510 & 0.3727 & 0.4064 & 0.1499 & 0.5771 & 0.6410 \\ 1.4951 & 1.5249 & 1.8003 & 0.4375 & 0.5530 & 0.1808 & 0.5337 \\ 0.8679 & 0.4027 & 0.7757 & 1.5698 & 1.8955 & 1.9402 & 1.2940 \\ 0.3662 & 0.5031 & 0.3166 & 0.3039 & 0.3346 & 0.4824 & 0.5104 \\ 0.4071 & 0.7517 & 0.6138 & 1.9229 & 1.1510 & 1.9567 & 1.3429 \end{pmatrix}$$

where first three columns were generated based on a vector

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

and the last four based on a vector

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

by adding noise to them.

Applying Standard NMF to X gives us the following matrices F and G:

$$F = \begin{pmatrix} 0.2185 & 0.0546 \\ 0.2980 & 0.0518 \\ 0.2720 & 0.0576 \\ 0.0494 & 0.2158 \\ 0.0652 & 0.1783 \\ 0.0087 & 0.2633 \\ 0.0882 & 0.1785 \end{pmatrix}$$

$$G = \begin{pmatrix} 1.3920 & 5.9936 & 0.8937 & 1.1800 & 0.7422 \\ 1.7456 & 0.5316 & 7.8521 & 1.6372 & 7.4038 \end{pmatrix}$$

It is clear that first three rows of matrix G have bigger values in first column and the last four in the second one. It indicates that columns from 1 to 3 form one cluster and the rest another one.

Now let us consider F. We can normalize matrix F to see clearly that vectors found by Standard NMF are the basis vectors used to generate the corresponding clusters.

Example 2. Let us consider a 10×784 matrix X presented in Figure 2.2 which has rescaled 28×28 images of handwritten numbers 4 and 5 as its lines.

Figure 2.2: Images from MNIST data set

Applying Standard NMF to X gives us the following matrice G:

$$G = \begin{pmatrix} 0.2626 & 0.0001 \\ 0.1369 & 0.0946 \\ 0.4607 & 0.0001 \\ 0.0680 & 0.0689 \\ 0.0708 & 0.0931 \\ 0.0001 & 0.1539 \\ 0.0002 & 0.1481 \\ 0.0003 & 0.1428 \\ 0.0001 & 0.1315 \\ 0.0001 & 0.1670 \end{pmatrix}$$

We can see that this time the accuracy of clustering decreased because of two images of the number four that were misclassified. We can explain this by the fact that the quality of those two images is quite low and NMF was not able to distinguish 4 from 5 even though the values in fourth and fifth lines are pretty close.

Matrix F represents two images that can be considered as the basis vectors of this data set. Indeed, in Figure 2.3 we see two clear images of 4 and 5.

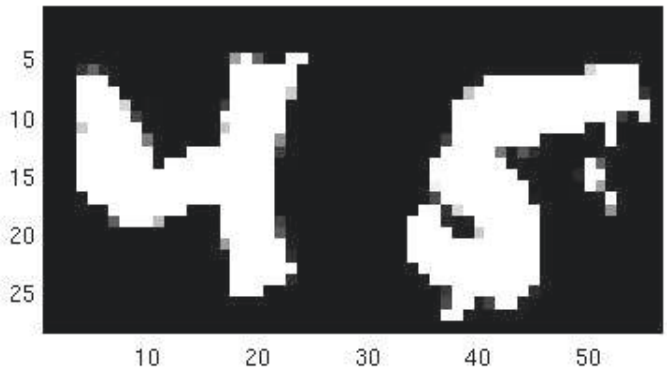


Figure 2.3: Basis vectors from matrix H

2.9 Conclusions

In this chapter, we presented a family of clustering methods based on NMF, the solutions to their corresponding optimization problems, their advantages and inconve-

niences. These algorithms enjoy local optima convergence which can be proved using a general approach based on the notion of the auxiliary functions. From the examples of NMF being applied to both artificial and real-world data sets, we can deduce the following:

- NMF can be efficient for dictionary learning due to its capability of reducing the dimensionality of the initial space while preserving the intrinsic nature of data;
- the presented methods give a vast choice of possibilities w.r.t. the eventual applications as they include linear, non-linear, hierarchical and spectral models;
- deriving multiplicative update rules is rather straightforward so as their implementation.

Chapter 3

Transfer learning

3.1 Introduction

Transfer learning is a widely known technique that was generally inspired by the ability of a human being to detect and to use previously gained knowledge in one area for efficient learning in another. In general, the definition of transfer learning was given in [Pan and Yang, 2010][1] as:

Definition 2. Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a target task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using knowledge gained from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

In this definition the notion of a *domain* is given by a pair of objects $\mathcal{D} = \{\mathcal{X}, P(X)\}$ where $\mathcal{X}$ represents the feature space and $P(X)$ stands for marginal distribution of $X = \{x_1, x_2, ..., x_n\} \in \mathcal{X}$. For a given domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, the *task* is defined as $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ where $\mathcal{Y}$ is a label space and $f(\cdot)$ is an objective predictive function usually written as a conditional probability of labels with respect to data instances, i.e., $f(x) = P(y|x)$. That being said, the condition $\mathcal{D}_S \neq \mathcal{D}_T$ implies either $\mathcal{X}_S \neq \mathcal{X}_T$ or $P_S(X) \neq P_T(X)$. The same thing for a task, $\mathcal{T}_S \neq \mathcal{T}_T$ implies either $\mathcal{Y}_S \neq \mathcal{Y}_T$ or $P_S(Y|X) \neq P_T(Y|X)$.

We follow the above mentioned survey and categorize transfer learning algorithms on two different levels. On the first level, we define three groups of algorithms as follows:

- supervised or inductive transfer learning (when labeled samples are available in target domain but there can be no labeled instances in the source one);
- semi-supervised or transductive transfer learning (labeled samples are available only for the source learning task);
- unsupervised transfer learning (no labeled data both in source and target learning tasks).

Then for each of the defined groups, we categorize transfer learning methods based on the way they proceed to perform the transfer of knowledge:

[1] In this section we follow the survey of [Pan and Yang, 2010] and complete it with recent contributions presented at top machine learning venues in last five years.

- instance-transfer approaches (reweightning of relevant labeled data in source domain to further use it in the target domain);
- feature-representation-transfer approaches (learning a shared feature representation for both domains in order to find invariant components);
- parameter-transfer approaches (imposing shared parameters or priors on source and target domain models to further induce a transferred target model influenced by source model through the discovered parameters);
- relational-knowledge-transfer approaches (matching relational knowledge between source and target domains and further relaxing the i.i.d. assumption in each domain).

Figure 3.1 presents an overview of major differences between presented settings of transfer learning.

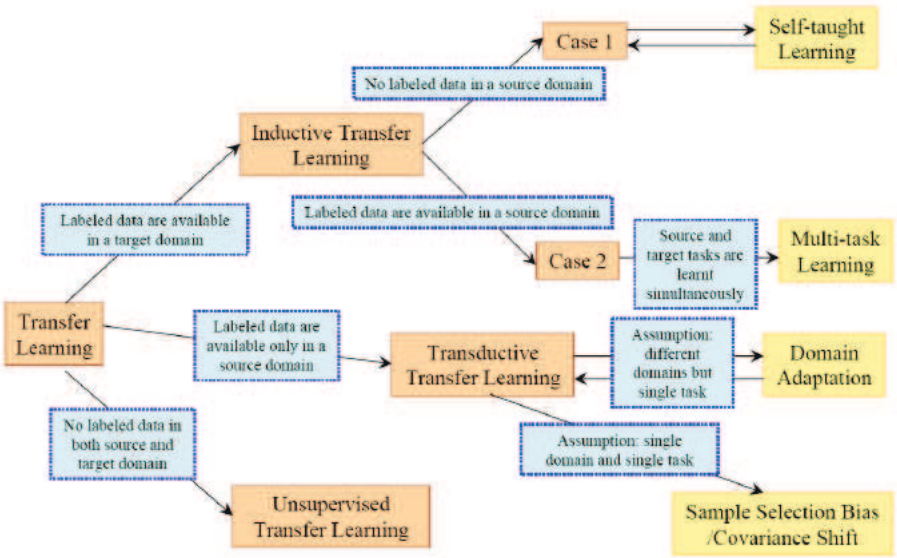


Figure 3.1: Different settings of transfer learning (Figure depicted from [Pan and Yang, 2010].)

We also note that tasks for both inductive and transductive settings include regression and classification (due to the presence of labels in source and/or target domains) while unsupervised setting is concentrated around clustering and dimensionality reduction (as there are no labels available).

3.2 Inductive transfer learning

We first give a definition of inductive transfer learning.

Definition 3. Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a target task $\mathcal{T}_T$, inductive transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using knowledge gained from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$.

This transfer learning setting always assumes that labeled data are necessarily available in the target domain as it is used to induce the target predictive function. On the other hand, according to Figure 3.1, there are two possible variations of inductive transfer learning that depend on the eventual absence/presence of labels in the source domain. The former case can be related to multi-task learning, while the latter is usually related to self-taught learning.

3.2.1 Transferring knowledge of instances

The most straightforward and intuitive way for a transfer learning approach to proceed is to use relevant instances from source domain directly in combination with target learning samples.

One of the first methods that addressed this problem was presented in [Wu and Dietterich, 2004] where the source domain auxiliary data was used in SVM framework to improve the classification accuracy of the target data samples. This was achieved by introducing an additional term to the cost function of SVM (in the form of a support vector or of an additional constraint) that takes into account weighted nearest neighbors corresponding to auxiliary data calculated with respect to the original data instances.

[Liao et al., 2005] proposed an active learning approach that adapts a classifier learned on labeled source data to partly unlabeled auxiliary data in the context of logistic regression. Their approach proceeds by training a classifier on weighted pairs of labeled instances where the corresponding weights represent the mismatch between unlabeled and labeled data.

Another way to transfer instances is to look at the conditional distributions $P(y_T|x_T)$ and $P(y_T|x_T)$ of the corresponding domains [Jiang and Zhai, 2007]. The proposed model maximizes the adapted log-likelihood of three different components: first one

is a weighted combination of labeled source instances multiplied by two coefficients that were estimated from the ratio of marginal distributions and from the mismatch of conditional distributions, respectively; second one is a combination of labeled target instances; the third one is a set of unlabeled target coefficients weighted using a bootstrap semi-supervised optimization.

Arguably, one of the most famous instance-based inductive transfer learning approaches is TrAdaBoost [Dai et al., 2007]. TrAdaBoost is basically inspired by a famous machine learning algorithm called AdaBoost [Freund and Schapire, 1996]. The main idea of the proposed approach is to train a base classifier on weighted source and target data in order to further evaluate its performance on target data only. The goal then is to update weights of the source instances based on their impact on the learning performance, i.e., decrease weights of missclassified instances to weaken their impact on the classification error and vice versa. The iterative procedure presented in the original work enjoys the existence of theoretical guarantees obtained for the generalization error on the combined data set.

Finally, some of the most recent approaches proposed to tackle this problem include [Lim et al., 2011] and [Haase et al., 2014]. The method presented in [Lim et al., 2011] makes use of the idea that examples from the same classes can be directly borrowed if they are similar. The borrowing procedure is performed by optimizing a standard binary classification loss-function multiplied by weights that define how many examples are borrowed. This cost function is further combined with a regularization term corresponding to sparse group lasso criterion that forces borrowed examples to be similar to the target data.

In [Haase et al., 2014] the authors used a convex combination of source and target errors where the former was weighted based on the ratio of the corresponding marginal distributions. This model is further used to obtain an orthonormal basis of the transferred Active Appearance Model. An interesting point that was implemented in this approach was to weight the source samples according to their innovation, i.e., to choose source samples that may provide information that is not covered by target data.

3.2.2 Transferring knowledge of feature representations

With the current success of representation learning [Bengio et al., 2013], transfer learning methods based on feature-representations witness a growing attention among researchers in machine learning community. In general, the main goal of a given feature-representation transfer approach consists in discovering a new set of features that reduce the discrepancy between the underlying source and target distributions while maintaining a low classification error over a set of tasks. Multiple strategies can be used in order to learn a new feature representation in inductive setting depending on the eventual absence or presence of labeled data in the source domain. If labeled data are absent, one may use unsupervised techniques to learn a shared feature representation. Otherwise, some well-known supervised methods based on the direct minimization of the joint loss-function can be applied.

Supervised algorithms for joint feature learning. One of the first attempts to construct shared features for different tasks was presented in [Jebara, 2004]. The proposed framework suggest to learn a set of features shared among different tasks and their corresponding discriminant functions by incorporating them into a SVM-like model. The final decision rule of the proposed approach depends on a parameter that allows to flow heterogeneously from learning all tasks separately to a single feature selection configuration that achieves good classification performance for all models.

Another interesting idea that was used for supervised feature construction is to find a linear mapping that projects data from all tasks to a shared low-dimensional representation [Argyriou et al., 2007]. The proposed approach minimizes a cost function that represents a combined sum of empirical errors corresponding to different tasks plus a regularization term over the coefficients. The goal of the optimization procedure is to learn a shared low-dimensional embedding for all tasks and a sparse parameter's matrix simultaneously. The cost function of this method takes the following form:

$$\arg\min_{U,A} \sum_{t\in\{T,S\}} \sum_{i=1}^{n_t} L\left(y_{t_i}, \left\langle a_t, U^T x_{t_i}\right\rangle\right) + \gamma \|A\|_{2,1}^2$$

$$s.t.\ U \in \mathbf{O}^D.$$

This optimization procedure tries to find an orthogonal matrix U and regression parameters A(a_t is a parameter vector related to task t) that lead to a low classifica-

tion error over all tasks. The regularization penalty represented as a mixed $\ell_{1,2}$ norm ensures that the obtained solution will be sparse and that the common features will be selected for a combination of tasks. Another paper from [Argyriou et al., 2008] presented a similar approach that makes use of a spectral regularization framework to discover a structural matrix of a set of tasks. The authors also pointed out some ideas on the equivalence between kernel learning and the proposed regularization framework. This connection was further fully developed in [Rückert and Kramer, 2008] where the same kind of reasoning was used to generate kernels that generalize well on the known source data sets to further use them on the new target one.

In [Lee et al., 2007] authors assumed that features across of all tasks have meta-features that describe both the properties of the feature and its potential relationship to the prediction problem. Their relevance is defined using hyperparameters (called meta-priors). These meta-priors are further transferred across different tasks that allows to improve the performance even in case if the tasks have non-overlapping features.

Some recent advances in feature-based transfer learning include [Zhong and Kwok, 2012] and [Guo and Xiao, 2012]. [Zhong and Kwok, 2012] addressed a common issue of many multi-task learning algorithms that lies in the assumption about the close relatedness of tasks. The proposed method discovers task relationships depending on the interactions among tasks and defines different task clusters for different features where the number of clusters may not be specified beforehand. Their cost function is given as follows:

$$\min_{U,V} \sum_{t=1}^{T} \|y^{(t)} - X^{(t)}(u_t + v_t)\|^2 + \lambda_1 \|U\|_{clust} + \lambda_2 \|U\|_F^2 + \lambda_3 \|V\|_F^2.$$

It basically consists of two terms: first one minimizes the empirical squared error over the tasks and discovers two matrices U and V representing a shared and a distinct part of a set of tasks; second one is a regularization terms over these matrices (where $\|U\|_{clust}$ is the sum of pairwise differences for elements in each row of U). Also, the proposed method is theoretically sound as it relies on the convex optimization problem with proven convergence guarantees.

In [Guo and Xiao, 2012] a similar idea was used in order to perform cross-language text classification. The proposed framework minimizes a classification error of each

classifier in each language while penalizing the distance between the subspace representations of parallel documents. The authors, then, applied their method successfully to a task of machine translation where documents in each language are translated into parallel documents in the other language to create two independent views of the text objects in different feature spaces.

Finally, an interesting theoretical study of inductive feature-based transfer learning based on sparse coding was presented in [Maurer et al., 2013]. The main assumption of this work is that the tasks parameters can be approximated by sparse linear combinations of the atoms of a dictionary on a high or infinite dimensional Hilbert space. The generalization bounds presented in this paper for both multi-task and transfer learning settings, allowed authors to derive a new algorithm based on sparse coding which achieves a considerably good results compared to other state-of-the-art methods in both settings with an increasing number of tasks.

Unsupervised algorithms for joint feature learning. Almost all feature-based transfer learning algorithms that construct features in an unsupervised manner were inspired by a method presented in [Raina et al., 2007]. The framework proposed in this paper is usually called *Self-taught learning* as it makes use of unlabeled data in source task to extract an overcomplete dictionary of basis vectors that will be further refined using available labeled instances. Formally, it is a two-stage algorithm that works as follows: at first stage it learns basis vectors $b = \{b_1, ..., b_s\}$ using unlabeled data $X_S^{(u)}$:

$$\min_{a,b} \sum_i \|x_{S_i}^{(u)} - \sum_j a_{S_i}^j b_j\|_2^2 + \beta \|a_{S_i}\|_1$$

$$s.t.\ \|b_j\|_2 \leq 1,\ \forall\, j \in 1, ..., s$$

then it uses them to obtain a new representation of labeled instances $X_T^{(l)}$ through the following optimization procedure

$$\hat{a}_{T_i}^{(l)} = \arg\min_{a_{T_i}} \|x_{T_i}^{(u)} - \sum_j a_{T_i}^j b_j\|_2^2 + \beta \|a_{T_i}\|_1.$$

Despite its superior performance on some benchmark data sets, it should be noted that a strong assumption about the similar modality of labeled and unlabeled samples needs to hold, otherwise the proposed method will most likely fail. Other methods that follow

the above mentioned algorithm include but not limited to [Raina, 2008] and [Lee et al., 2009].

One of the most recent results proposed for this paradigm was presented in [Wang et al., 2013]. This paper tries to overcome an important flow of the original approach in a very straightforward way by performing dictionary learning from both labeled and unlabeled data simultaneously. The presented method was evaluated on famous computer vision benchmarks and proved to be efficient when the size of the dictionary was chosen appropriately.

3.2.3 Other inductive transfer learning methods

Two other types of inductive transfer learning algorithms are parameter-based and relational transfer learning. In the following paragraph we will only make a brief overview of the above mentioned algorithms as they are out of the scope of this thesis.

Transferring knowledge of parameters. Parameter-based transfer learning methods usually assume that there exists a set of shared parameters or prior distribution of hyperparameters for individual models that can be used in multi-task setting to improve simultaneously the performance of both source and target tasks. This inductive transfer learning setting is quite similar to feature-based and instance-based frameworks as it essentially tries to use the obtained parameters to learn a new feature representation or to reweight instances from source domain to further use them in combination with target task data.

Arguably the most common way to transfer parameters is to use Gaussian Processes (GP) with a shared prior over multiple tasks. First work on this matter, presented in [Lawrence and Platt, 2004], considers learning parameters of a GP over multiple tasks with the same GP prior. In [Schwaighofer et al., 2005] GP approach is combined with hierarchical Bayesian framework. Finally, in [Bonilla et al., 2008] GP prior is used to model inter-tasks dependencies using a free-form covariance matrix.

Recently, [Srivastava and Salakhutdinov, 2013] proposed a new interesting approach for parameter-based transfer learning that uses tree-based priors combined with Deep Neural Networks (DNN). The proposed approach benefits from the discriminative power of DNN that leads to an improved classification performance when evaluated on benchmark data sets. Another interesting point of this this paper is that

the proposed framework works not only on fixed predefined trees but also can generate meaningful trees itself. Finally, we note that this paper is a direct extension of [Salakhutdinov et al., 2011a] and [Salakhutdinov et al., 2011b].

Another way to transfer parameters is to combine multiple models learned in source domain in order to use them further on target task. In [Gao et al., 2008] the combined model is defined through a weightning procedure where the corresponding weights are defined based on the predictive power of a given model w.r.t. the target task. Similar adaptive model-based approach was also presented in [Tommasi et al., 2010]. The authors proposed a SVM-like model that is able to select and weight appropriately prior knowledge coming from different tasks. Finally, the latter can be seen as an extension of [Evgeniou and Pontil, 2004] where SVM parameters of source and target tasks are assumed to be composed of a shared and a domain-specific parts. Then, the authors proposed a regularization framework that learns both shared and domain-specific parameters simultaneously.

Last point that we would like to discuss here is a theoretical analysis of parameter-based transfer learning. PAC-Bayesian analysis of parameter-based transfer learning in the lifelong learning setting was presented in [Pentina and Lampert, 2014]. Lifelong learning, first introduced in [Baxter, 2000], is a new promising research direction in machine learning that describes a situation where the goal is to transfer information to tasks for which no data have been observed so far. Results presented in this paper show that multitask risk is bounded by the empirical multitask risk plus two terms where the first one captures the divergence between the hyperprior and the hyperposterior distributions while the second one is a sum of divergences between i-th task hyperposterior and the hyperprior. This result is further applied to show how one can derive efficient algorithms for parameters transfer learning setting.

Transferring relational knowledge. The relational-knowledge transfer approach is an inductive transfer learning paradigm that arises when transfer learning problem is defined on relational domains and the corresponding data drawn from source and target tasks are not independent and identically distributed (i.i.d.) and thus can be represented by multiple relations.

First approach that applied transfer learning to relational domains was presented in [Mihalkova et al., 2007]. The authors used Markov Logic Networks (MLNs) to define mappings between predicates of source and target domains in order to use them

further to transfer clauses. Each clause has a corresponding weight that one needs to adapt before transferring source MLN to target domain. As an example, we may consider two domains where one represents individuals and their relationships in an academic department while the other one models cinema industry based on International Movie Database (IMDB). Obviously, predicate mappings can be established as directors play the same role as professors when related to actors and students, respectively. The proposed two-stage algorithm is known as TAMAR. Its further extension to single-entity-centered setting was presented in [Mihalkova and Mooney] where only one instance in target domain is available.

Another interesting approach that proceeds in a similar manner was proposed in [Van Haaren et al., 2015]. The difference, however, is that the proposed algorithm TODTLER uses both first-order and second-order logic, i.e., transfers second-order clauses from source domain to target domain by biasing learner in the latter towards models containing previously discovered regularities in the former. Authors stated that second-order clauses add depth to the proposed model and thus it falls into to the family of deep learning approaches. TODTLER is an improved version of DTM algorithm presented in [Davis and Domingos, 2009]. DTM is also based on second-order clauses that are transferred to target domain in form of cliques. Then, the proposed algorithm defines models involving these cliques and further tailors them based on the target domain alone.

Other applications of relational-transfer include planning using Web search [Zhuo and Yang, 2014] and automatic annotation [Jiang et al., 2015].

3.3 Transductive transfer learning

Term "transductive learning" [Arnold et al., 2007] in traditional machine learning usually describes a situation when all test data are required to be available during training time and that the obtained classifier cannot be applied to future data. In transfer learning, however, it stands for a different, yet similar, concept. We first give a definition of transductive transfer learning.

Definition 4. Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a target task $\mathcal{T}_T$, transductive transfer learning aims to improve the learning of

the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using knowledge gained from $\mathcal{D}_S$ and $\mathcal{D}_T$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{T}_S = \mathcal{T}_T$. Furthermore, some unlabeled data in target domain are available at training time.

In this definition, the equivalence of source and target tasks means that we can adapt the predictive function learned in source domain to classify the unlabeled target domain data. As it can be seen from the previous section, the obtained predictive function should take into account the distribution mismatch between source and target data in order to be efficient. In this setting, two different assumptions are possible: (1) the feature spaces of source and target domains are different, i.e. $\mathcal{X}_S \neq \mathcal{X}_T$; (2) feature spaces of both domain are the same while the marginal distributions are different, $P_S(X) \neq P_T(X)$. These two cases are usually referred as domain adaptation and sample selection bias, respectively. Domain Adaptation (DA) is a field associated with machine learning and transfer learning. This scenario arises when we aim at learning from a source data distribution a well performing model on a different (but related) target data distribution. For instance, one of the tasks of the common spam filtering problem consists in adapting a model from one user (the source distribution) to a new one who receives significantly different emails (the target distribution). Note that, when more than one source distribution is available we talk about multi-source domain adaptation.

In this section, we will briefly overview sample selection bias algorithms and give a more profound overview of domain adaptation as it includes one of the contributions of this thesis.

3.3.1 Transferring knowledge of instances

Most instance-based transfer learning algorithms in transductive setting transfer knowledge by learning a model that minimizes the weighted empirical risk of source domain in the following way:

$$\theta^* = \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{D}_S} \frac{P(\mathcal{D}_T)}{P(\mathcal{D}_S)} P(\mathcal{D}_S)\, l(x,y,\theta) \approx \arg\min_{\theta\in\Theta} \sum_{i=1}^{n_S} \frac{P_T(x_{T_i}, y_{T_i})}{P_S(x_{S_i}, y_{S_i})} l(x_{S_i}, y_{S_i}, \theta)$$

where empirical expected risk of target domain is approximated by the ratio of target and source domains joint probability distributions multiplied by loss-function on labeled source instances. Furthermore, we can use the assumption about the equivalence of conditional distributions $P(Y_T|X_T) = P(Y_S|X_S)$ to obtain $\frac{P_T(x_{T_i},y_{T_i})}{P_S(x_{S_i},y_{S_i})} = \frac{P(x_{S_i})}{P(x_{T_i})}$. Therefore, the major challenge of transductive transfer learning is to estimate this ratio and to use the corresponding weights in combination with empirical risk minimization of labeled source data. After that, the obtained model can be used directly on target task.

Term "sample selection bias" was known for quite a long time in econometrics while in machine learning it appeared first in [Zadrozny, 2004]. This paper formalized the problem of sample selection bias in machine learning terms, analyzed both empirically and theoretically how some classification techniques are affected by it and presented a new approach for sample selection bias correction. The proposed approach suggests using a costing procedure developed in a prior work [Zadrozny et al., 2003] to weight each example by the selection ratio. Finally, the proposed paper presented an approach that can be used to evaluate classification performance under sample selection bias.

[Bickel et al., 2007] introduced a discriminative approach that defines weights based on the probability that a given source domain sample will appear in the target domain distribution. These weights are learned by maximizing the posterior probability distribution given all available data to be further used for reweightning.

Another idea proposed in [Huang et al., 2007] is to assign weights to source data by matching the means of feature vectors between source and target data in a Reproducing Kernel Hilbert Space (RKHS). The proposed optimization problem has the following form:

$$\min_{\beta} \frac{1}{2}\beta^T K\beta - \kappa^T\beta$$

$$s.t.\ \beta_i \in [0, B]\ \ and\ |\sum_{i=1}^{n_s}\beta_i - n_s| \leq n_s\epsilon,$$

where the solution can be proved to have the following form $\beta_i = \frac{P(x_{S_i})}{P(x_{T_i})}$. Theoretical analysis for this estimator has been recently presented in [Yu and Szepesvri, 2012]. An important issue of this algorithm, however, is that it does not allow the selection of kernel parameters through cross-validation. This problem was addressed in [Sugiyama

et al., 2008] and a two-stage method based on Kullback-Leibler divergence minimization was proposed.

Finally, the most exhaustive study on instance reweightning in transductive transfer learning setting was presented in [Zhang et al., 2013]. This paper covers all three possible scenarios that can occur depending on the assumptions made, namely: (1) target shift when marginal distributions change but conditional distributions remain the same; (2) conditional shift which is the opposite of target shift; (3) generalized target shift which is combination of both target and conditional shifts. This paper uses Hilbert-Schmidt embeddings for marginal as well as conditional probability functions to propose algorithms for all three scenarios and provide theoretical guaranties for each of them. We note that in [Sun et al., 2011] authors also attempted to take into account both conditional and target shifts using a two-stage kernel-based method and that it can be seen as a special case of methods presented in [Zhang et al., 2013].

Other recent examples of methods proposed for sample selection bias include [Liu and Ziebart, 2014] and [Wen et al., 2014]. [Liu and Ziebart, 2014] defined robust bias-aware classifier as a solution to a two-player game with minimax log-loss. Then, authors proposed a parameter-based form solution that depends on $\frac{P(x_{S_i})}{P(x_{T_i})}$ estimation for importance sampling and compared both of them to source logistic regression. The proposed robust bias-aware classifier in some cases achieves a significantly better results than both source and reweighted target regressors.

In [Wen et al., 2014] authors related learning under marginal distributions shift to model misspecification. They showed that reweightning sometimes is simply not enough to adapt well in case if the underlying model was not chosen appropriately. The proposed method robust covariate shift adjustment (RCSA) allows to check whether reweightning procedure can be beneficial for a given model and, if it is the case, to find relevant features.

Finally, a theoretical study on domain adaptation and sample selection bias algorithms for regression was proposed in [Cortes and Mohri, 2014]. This paper presented new pointwise loss guarantees based on the discrepancy of the empirical source and target distributions in a RKHS for the general class of kernel-based regularization methods. Furthermore, authors derived algorithms that can be used on large-scale data sets in high-dimensional space due to the existence of efficient solvers for Semidefinite programming (SDP) problem that the original problem can be reduced to.

3.3.2 Transferring knowledge of feature representations

In transductive transfer learning, feature-representation transfer methods play an important role because of two reasons: (1) current success of representation learning proved that learning "good" features is a key component in the success of machine learning algorithms; (2) numerous unsupervised feature extracting techniques can be used as a basis to find a shared representation of features that reduces the discrepancy between source and target distributions. Last reason is of crucial importance as it allows to design algorithms that are usually referred as unsupervised domain adaptation. These algorithms do not rely on the presence of labeled data when it comes to learning a shared embedding but use it only to infer a predictive function in target aligned source domain.

Furthermore, an increasing interest in transductive transfer learning setting resulted in Unsupervised and Transfer Learning challenge (UTL) [Guyon et al., 2011] organized in 2010. Its main goal was to encourage scientists to learn "good" feature representations for cross-domain transfer learning. Overall, UTL challenge attracted 76 participants from some of the best research institutions in machine learning community and resulted in a vast number of new interesting approaches for feature-based transductive transfer learning.

Following [Margolis, 2011] we categorize feature-based transductive transfer learning methods into two categories:

- distribution similarity approaches;
- latent feature learning approaches.

First family of methods is related to transfer learning approaches that aim to discover a feature representation in which a distribution divergence measure is minimized by explicitly penalizing or excluding distinct features from both domains. Second type of feature-based approaches tries to find a latent feature space by using unsupervised methods applied simultaneously to unlabeled source and target data.

Distribution similarity approaches. One of the first works that addressed this problem was presented in [Aue and Gamon, 2005]. A simple approach for NLP task proposed in this paper suggests that one may just train a classifier in source domain based on features that are present in target domain while the absent features are simply

ignored. The proposed method was applied to sentiment classification task and showed a good performance in some cases but failed in others even when compared to "all data" setting (when a classifier is learned on both source and target data sets with no preprocessing). Another method that uses distinct feature elimination was described in [Margolis et al., 2010]. Both approaches are rather straightforward and their degraded performance in some cases can be explained by the loss of auxiliary knowledge in both domains when distinct features are eliminated.

A more sophisticated approach to align source and target data based on divergence minimization was presented in [Satpal and Sarawagi, 2007]. Authors proposed to penalize the distorted features across two domains in order to give more influence to relevant correlated features. This procedure is combined with maximization of the likelihood over source labeled data so that the cost function has the following form:

$$\arg\max_{w} \sum_{i \in \mathcal{D}_S} \sum_{k} w_k f_k(x_i, y_i) - log(z_w(x_i)) - \lambda \sum_{k} |w_k|^{\gamma} d(\mathbb{E}_S \{f_k(x,y)\}, \mathbb{E}_T \{f_k(x,y)\}$$

where the goal is to learn a weight vector w for features $f_k(x, y)$. The solution to this optimization problem involves two steps: (1) computing distances between feature means to update weights w_i; (2) fix the weights and update the feature means. A similar idea was used in [Arnold et al., 2007] but based on maximum entropy method [Berger et al., 1996]. The underlying idea is to learn a transformation function that aligns features across domains and thus results in the equivalence of joint probability distributions of features and labels in both domains. This method, however, is quite similar to instance-based methods as it basically scales features based on the ratio of the estimated source and target distributions.

As we could see in the previous section, Hilbert space embeddings can be very efficient in feature-based algorithms as they provide a possibility to align features using a nonlinear map from a rich Hilbert space. To this end, methods based on Maximum Mean Discrepancy (MMD) minimization [Chen et al., 2009; Pan et al., 2008, 2009] were proposed.

The approach in [Pan et al., 2008] is based on MMD minimization combined with maximum variance unfolding (MVU) presented in [Weinberger et al., 2004]. The goal of the latter is to find a kernel that corresponds to a low-dimensional manifold by maximizing the variance subject to fixed distances between neighbors. This allows to

obtain a kernel matrix that the authors used further with kernel PCA. The major issue of this approach is that it requires solving a SDP that makes it inapplicable on large data sets. In the follow-up work presented in [Pan et al., 2009] this issue has been overcome by replacing the initial computationally costly SDP with an eigenvalue decomposition. Contrary to approaches in [Pan et al., 2008] and [Pan et al., 2009] that use nonlinear projections and does not require labeled data to find a shared embedding, [Chen et al., 2009] introduced a method that learns an orthogonal linear projection to align the sample means of source and target data. A learning procedure then consists in direct minimization of the classification error over source domain samples using projected and original features. Main advantage of this approach comes from the linearity of the projection as it does not rely on solving a SDP. This, however, makes it less flexible as kernel functions provide a richer class of possible nonlinear feature maps.

Recently, a general framework for transductive feature-based transfer learning based on MMD was proposed in [Long et al., 2014a]. The proposed framework introduces two regularization terms that minimize the MMD distance between both marginal and conditional distributions of source and target domains. Provided that no labeled data are available in target domain, the authors used a classifier learned in source domain to pseudo-label target domain samples hoping that the obtained centroids will be close to the true class centroids. This framework was further incorporated into Regularized Least-Squares (RLS) and Support Vector Machine's cost functions to derive new transductive transfer learning algorithms. An important advantage of this regularization framework is that the corresponding cost function is convex and thus it enjoys a convergence to a global optima. Finally, in [Si et al., 2010] a regularization term based on the Bregman divergence between source and target distributions was introduced for transfer subspace learning. The proposed method, however, does not take into account the discrepancy between conditional distributions.

Latent feature learning approaches. Structural correspondence learning (SCL) presented in [Blitzer et al., 2006] is probably the most referenced latent feature linear projection approach. SCL is based on the semi-supervised technique called alternating structural minimization (ASO) introduced in [Ando and Zhang, 2005]. ASO makes use of auxiliary unlabeled data to learn a "predictive structure" on the hypothesis space across multiple tasks. To develop SCL, the authors' idea was to learn the correspondences between features that are distinct for source and target domains. The proposed

method has three stages: (1) it defines a set of pivot features that appear with a high frequency in both domains (in the follow-up work [Blitzer et al., 2007] it was replaced with a constraint that obliges features to have high mutual information with the label in source domain); (2) a linear classifier is then learned for each pivot feature based on the weighted original features; (3) finally, SVD is applied to the matrix of learned weights resulting in a reduced dimensional space formed by a prefixed number of top singular vectors. Although, SCL remains one of the most efficient approaches in practice with successful applications in cross-language machine translation [Prettenhofer and Stein, 2010] and speech classification [Margolis et al., 2010], entity recognition [Ciaramita and Chapelle, 2010] and conversation summarization [Sandu et al., 2010], choosing the optimal number of dimensions for SVD and weights to be used with a final classifier can present a problem if no labeled target data are available.

Some recent approaches proposed for latent feature learning include [Grauman, 2012], [Gong et al., 2013a], [Fernando et al., 2013] and [Long et al., 2014b]. In [Grauman, 2012] a method called Geodesic Flow Kernel (GFK) was proposed. GFK consists of three main steps: (1) first, it computes an optimal number of dimensions for the subspace of the embedding; (2) second, it constructs a geodesic path; (3) finally, it computes a geodesic kernel that is further used to learn a classifier with labeled data in source domain. To choose the optimal dimensionality of the embedding authors proposed a new subspace disagreement measure (SDM) that computes angles between principal components of source and target data with respect to principal components of the combined source and target data. A greedy strategy is then used to define the optimal number of aligned principal components. Geodesic flow kernel is then obtained in a closed form as inner-product calculated on the projection of initial features. The proposed projection function, in its turn, is a continuous function that parametrizes how the source data smoothly changes to the target data. Surprisingly, authors stated that if one calculates the geodesic kernel over all possible subspaces generated by the projection function on the geodesic path (i.e., for all values of the projection function), the obtained geodesic kernel will be invariant to all possible variations between source and target domains.

In [Gong et al., 2013a] a different strategy was used. The central idea of the proposed method is to define landmarks - labeled instances in source data that are more likely to be distributed in the same way in target domain. These landmarks were de-

fined by minimizing the MMD distance between source and target samples. After that, they were used to build auxiliary tasks that bring source and target domains closer in sense of Kullback-Leibler divergence. Finally, authors used these tasks as a basis to learn discriminative domain-invariant features for target domain by solving Multiple Kernel Learning (MKL) optimization problem. The proposed method has proved to be efficient and to outperform GFK on some benchmark data sets.

A very simple, yet robust, approach was introduced in [Fernando et al., 2013]. A key observation used in this paper was that one can simply align subspaces spanned by the principal components of source and target data (denoted by X_S and X_T) using the following cost function:

$$F(M) = \|X_S M - X_T\|_F^2 = \|X_S^T X_S M - X_S^T X_T\|_F^2 = \|M - X_S^T X_T\|_F^2$$

that leads to a solution $M^* = X_S^T X_T$. This solution was further used to define two key quantities, namely: the target aligned source coordinate system $X_a = X_S M^* = X_S X_S^T X_T$ and a similarity measure

$$Sim(y_S, y_T) = (y_S X_S M^*)(y_T X_T)^T = y_s X_S X_S^T X_T X_T^T y_T^T.$$

Authors presented a consistency theorem for the proposed similarity measure $Sim(y_S, y_T)$ that allowed them to find the optimal number of principal components to be chosen. A strong advantage of the proposed algorithm is its computational efficiency when compared to other kernel-based approaches as it basically involves only principal components calculation and simple matrix manipulations.

Finally, in [Long et al., 2014b] a new method called Transfer Joint Matching (TJM) was presented. The proposed method combines both instance reweightning, distribution divergence minimization and latent feature learning but as its primal goal is to find a projection of features to a low-dimensional embedding, we describe it here. The cost function of TJM consists of two terms: first term arises from the MMD minimization between source and target projected kernel principal components while the second one stands for regularization of the projection function over source and target data. The authors suggested that its superior performance on benchmark data sets can be explained by its capability to simultaneously match the feature distributions and reweight the

source instances in a principled dimensionality reduction procedure.

From theoretical point of view, unsupervised domain adaptation was investigated in [Ben-David et al., 2010a]. In the next chapter we will describe in detail the key contributions of this paper.

3.4 Unsupervised transfer learning

Similarly to previous sections, we start with a definition of unsupervised transfer learning.

Definition 5. Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a target task $\mathcal{T}_T$, unsupervised transfer learning aims to improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using knowledge gained from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$ and $\mathcal{Y}_S$ and $\mathcal{Y}_T$ are not observable.

To the best of our knowledge there are only a couple of algorithms that were proposed to solve this problem: self-taught clustering (STC) presented in [Dai et al., 2008a], transfer spectral clustering (TSC) [Jiang and Chung, 2012], Bregman multitask clustering (BMC)[Zhang and Zhang, 2011] and [Tran and d'Avila Garcez, 2013].

The main assumption of STC is that two tasks share a latent feature space that can be used as a "bridge" for transfer learning. The authors perform co-clustering on source and target data simultaneously, while the two co-clusters share the same feature set. The proposed method is based on mutual information maximization and its cost function has the following form:

$$J(\tilde{X_T}, \tilde{X_S}, \tilde{Z}) = I(X_T, Z) - I(\tilde{X_T}, Z) + \lambda \left[I(X_S, Z) - I(\tilde{X_S}, Z) \right],$$

where Z is a shared space of features, $\tilde{X_T}$, $\tilde{X_S}$ and $\tilde{Z}$ denote clustering solutions for X_T, X_S and Z.

TSC is a spectral method that makes use of both manifold information and co-clustering to transfer the knowledge among domains. The corresponding optimization

problem has the following form:

$$g(F^{(1)}, F^{(2)}, F^{(3)}) = \mathrm{tr}(F^{(1)^T} W_N^{(1)} F^{(1)}) + \mathrm{tr}(F^{(2)^T} W_N^{(2)} F^{(2)}) + \mathrm{tr}(F^{(3)^T} X_N^{(1)} F^{(1)}) + \mathrm{tr}(F^{(3)^T} X_N^{(2)} F^{(2)}).$$

where $W_N^{(i)}$ stand for the normalized nearest neighbors matrices of two tasks, $X_N^{(i)}$ for normalized data from the corresponding domains and $F_{i=1..3}^{(i)}$ are the embeddings for samples of the corresponding tasks and features, respectively.

Another approach that can be related to unsupervised transfer learning is [Zhang and Zhang, 2011]. The proposed method, however, is an instance of multi-task clustering rather then self-taught clustering. The optimization procedure presented in this work simultaneously minimizes two terms: first represents the sum of Bregman divergence between clusters and data of each task; second is a regularization term defined as the Bregman divergence between all pairs of partitions. The motivation for this cost function is two-fold - while the first term seeks a qualitative clustering for each task separately, second term takes into account the relationships between clusters of different tasks.

The most recent approach for unsupervised transfer learning was presented in [Tran and d'Avila Garcez, 2013]. This work uses Restricted Boltzman Machines (RBMs) to transfer subnetworks learned on source data set to the target domain. In order to avoid transferring irrelevant features, authors rank them with respect to target data and select only those features that have high weights. Experimental results presented in this paper for image recognition data sets showed that its performance can be superior when compared to a no-transfer sparse coding algorithm.

3.5 Transfer learning and NMF

In this section, we describe all transfer learning approaches that use, in one way or another, NMF methods described in the previous chapter. The methods that satisfy this criteria include [Ogino and Yoshida, 2011], [Long et al., 2012], [Markov and Matsui, 2012], [Chen and Zhang, 2013b], [Zhuang et al., 2013], [Yang et al., 2013] and [Wang et al., 2014].

In [Ogino and Yoshida, 2011] Topic Graph based NMF for Transfer Learning

(TNT) was introduced. The goal of this approach is to use learned feature vectors in the source domain to construct a graph structure called topic graph. This graph is utilized as a regularization term in the framework of NMF using the following cost function:

$$J = \|X_T - U_T V_T\|_F^2 + \nu tr(U_T L_S U_T^T) + \lambda tr(V_T^T L_T V_T^T),$$

where X_T is a target domain data, $L_{\{S,T\}} = D_{\{S,T\}} - U_{\{S,T\}}^T U_{\{S,T\}}$ is a graph Laplacian for X_S or X_T respectively and U_S is a matrix of basis vectors obtained by applying NMF to the source data. TNT is based on the previous work on NMF presented in [Cai et al., 2008] with the only difference that it adds one more regularization term preserving pairwise relation between two domains to the original cost function. The proposed method was evaluated on 20NewsGroups data set and showed a superior performance compared to baselines. Main weakness of this approach, however, lies in the assumption that both U_S and U_T span the same space and thus share the same features. A modified version of TNT was presented in [Yang et al., 2013]. The proposed approach differs from TNT only in the choice of the norm used for the regularization term - it replaces Frobenius norm in the cost function with $\ell_{2,1}$ norm.

In [Long et al., 2012] authors proposed a method called Dual Transfer Learning (DTL) that uses joint NMF framework to solve transductive transfer learning problem. Given a set of samples $\{X_i\}_{i=1..t}$ with corresponding labels $\{Y_i\}_{i=1..t}$ in source domain and a set of unlabeled samples $\{X_i\}_{i=t+1..n}$ in target domain, the goal is to minimize the following objective function:

$$\mathcal{L} = \sum_{i=1}^{n} \|X_i - [U, U_i]\, H V_i^T\|^2,$$

$$s.t.\ U, U_i, H, V_i \geq 0,$$

$$[U, U_i]^T\, \mathbf{1} = \mathbf{1},\ V_i \mathbf{1} = \mathbf{1}$$

where U represents common feature clusters for both domains, U_i represents domain-specific features and H stands for associations between them. Authors claimed that the proposed framework minimizes both marginal and conditional mismatch between source and target distributions even though no experimental tests were conducted to verify this. Similar to TNT, DTL was tested on 20NewsGroups and Reuters-21578 data

sets on cross-domain transfer learning problems and proved to be more efficient than other state-of-the-art NMF and co-clustering methods. In [Chen and Zhang, 2013b] an extension of DTL called Topical Correspondence Learning (TCL) was presented. The proposed method also tries to discover both domain-specific and common features by enforcing the same associations on them in the following way:

$$\mathcal{L} = \sum_{i=1}^{n} \|X_i - [\alpha U, (1-\alpha)U_i]\, HV_i^T\|^2,$$

$$s.t.\ U, U_i, H, V_i \geq 0,$$

$$[U, U_i]^T \mathbf{1} = \mathbf{1},\ V_i \mathbf{1} = \mathbf{1}$$

where

$$H = \begin{bmatrix} H & H \end{bmatrix}^T$$

and the hyperparameter $\alpha \in [0; 1]$ determines the probability of choosing a common or domain-specific term. This extension, however, is application-dependent as it was designed for the purpose of text classification. A further extension of DTL includes Triplex Transfer Learning (TTL) presented in [Zhuang et al., 2013]. TTL adds one more feature matrix to the cost function that represents distinct concepts and imposes unique association matrix for identical and alike concepts while association matrix for distinct concepts remains domain-dependent. The proposed objective function is given as follows:

$$\mathcal{L} = \sum_{i=1}^{n} \|X_i - \left[U^{identical}, U_i^{alike}, U_i^{distinct}\right] HV_i^T\|^2,$$

where

$$H = \begin{bmatrix} H^{identical} & H^{alike} & H_i^{distinct} \end{bmatrix}^T.$$

In [Markov and Matsui, 2012] a study of different self-taught learning algorithms was presented. Authors compared STL from [Raina et al., 2007] to PCA and a NMF-based STL approach that learns basis vectors using NMF instead of applying sparse coding. The experimental results showed that both STL and NMF-based STL have similar performance while PCA was far behind. Authors also pointed out that NMF does not allow to control sparsity of the basis vectors (while sparse coding does) that is, generally speaking, not true (for instance, NMF with sparseness constraints was

presented in [Hoyer and Dayan, 2004]).

Finally, the most recent result on transfer learning that makes use of NMF was presented in [Wang et al., 2014]. The cost function of the proposed method Domain Transfer Nonnegative Matrix Factorization (DomTrans-NMF) consists of three main terms: (1) first term is a simple NMF model applied to the combined source and target data set $X = X_S \bigcup X_T$ where $|X_S| = N_S$ and $|X_T| = N_T$; (2) second term corresponds to MMD minimization of partition matrices of source and target domains; (3) third term is a loss function of predicted labels w.r.t. the real ones. Overall, it leads to the following optimization problem:

$$\min_{U,V,\mathbf{w},b} \|X - UV\|_F^2 + \alpha\|V\boldsymbol{\pi}\|_2^2 + \beta\left\{\frac{1}{2}\|\mathbf{w}\|_2^2 + \gamma\|\left[(\mathbf{w}^T V + b\mathbf{1}) - \mathbf{y}\right]\boldsymbol{\iota}\|_2^2\right\}$$

$$s.t.\ U, V \geq 0,$$

where $(\mathbf{w}, b)$ are the parameters of a linear classifier $h(\mathbf{v}) = \mathbf{w}^T\mathbf{v} + b$, $\mathbf{y}$ is a vector of labels, $\boldsymbol{\iota}$ is an indicator if a given sample x_i is labeled and $\pi_i = \frac{1}{N_S}$ if x_i belongs to the source domain and $-\frac{1}{N_T}$ otherwise. DomTrans-NMF was further applied to the data from Brain-Interface competition that contains Electroencephalography (EEG) of 9 different persons classified into 4 different classes based on the motor imagery tasks. These data were collected during two days and the experimental setup proposed by the authors was to consider each day as a single domain. The results obtained using DomTrans-NMF were comparable to a kernel-based domain adaptation method presented in [Duan et al., 2012]. Overall, DomTrans-NMF was the first approach that combined direct distance-minimization between the source and target distributions and NMF.

3.6 Data sets

Some popular data sets used to evaluate transfer learning methods include: 20Newsgroup data set, Reuters data set, sentiment classification data set from [Blitzer et al., 2007], Office[Saenko et al., 2010]/Caltech[Gopalan et al., 2011] etc.

Most of the results in this thesis are evaluated on the famous Office/Caltech data set that has already became a benchmark for transfer learning algorithms. Office/Caltech

contains the 10 overlapping categories between the Office dataset and Caltech256 dataset and consists of four domains:

- Amazon (A) - images from online merchants (958 images with 800 features from 10 classes);
- Webcam (W) - set of low-quality images by a web camera (295 images with 800 features from 10 classes);
- DSLR (D) - high-quality images by a digital SLR camera (157 images with 800 features from 10 classes);
- Caltech (C) - famous data set for object recognition (1123 images with 800 features from 10 classes).

Figure 3.2 shows an example of keyboard and backpack images from Office/Caltech data set.

Figure 3.2: Examples of keyboard and backpack images from Amazon, Caltech, DSLR and Webcam data sets.

This set of domains leads to 12 domain adaptation problems, i.e., C $\rightarrow$ A, C $\rightarrow$ D, C $\rightarrow$ W, ..., D $\rightarrow$ W. Office/Caltech data set uses 20 source examples per category if source is Amazon, otherwise 8 examples per source and 3 labeled examples per target category. Following the typical preprocessing steps, all images were transformed to grayscale and resized to have the same size. K-means clustering was then applied to SURF descriptors in order to generate a codebook of size 800 from a subset of Amazon data set. Finally, all the features were standardized by z-score.

3.7 Conclusions

In this chapter, we presented different transfer learning methods both in inductive, transductive as well as unsupervised settings. We notice that transfer learning is a well-studied technique with numerous real-world applications that include natural language processing, text and image classification, automatic annotation, machine translation etc.

From the above presented overview, we make the following conclusions:

- historically, first methods proposed to perform the transfer of knowledge were based on instance reweightning; however, the situation has changed and nowadays representation-based algorithms become more and more common;
- inductive transfer learning has been studied extensively while the transductive setting (especially what is referred to as domain adaptation) is now arguably the most popular research direction among transfer learning scientists;
- unsupervised transfer learning is the less covered setting in transfer learning, so far. This, in its turn, makes it a topic of an ongoing interest for further researches.
- works on NMF-based transfer learning approaches started to appear lately and proved to be efficient in both inductive and transductive settings. The vast majority of the proposed methods, however, addressed the problem of text classification by adapting Tri-NMF with shared and domain-specific components.

Chapter 4

Kernel Alignment for Unsupervised Transfer Learning

4.1 Introduction

Transfer learning is considered to be useful only when both source and target domains have some semantic relationships. In [Rosenstein and Dietterich, 2005] the authors show that sometimes transfer learning can hurt performance if tasks are too dissimilar. The proposed approach changes the variance of the hyperprior depending on the similarity of parameters of source and target models and then computes a posterior distributions using hierarchical Naive Bayes. The evaluations of this approach are presented in Figure 4.1[1]. On the other hand, in [Mahmud and Ray, 2008] the authors

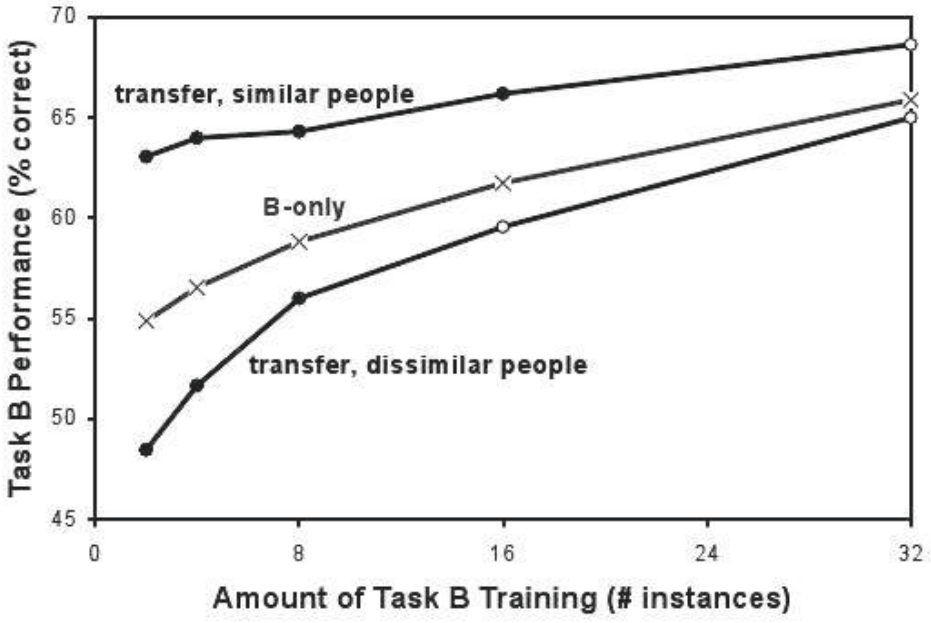


Figure 4.1: Transfer learning performance on data from a meeting acceptance task

used algorithmic information theory approach in order to perform supervised transfer learning between two tasks that have a very tenuous connection. For instance, their algorithm was successfully applied on Mushroom - German Credit data sets - two domains that have nothing in common. Authors called this new setting of transfer learning "universal transfer learning".

[1]Figure depicted from Rosenstein and Dietterich [2005].

Both of these works consider one of the biggest questions of transfer learning that is “how to avoid the ”negative transfer”?” - situation where the performance on a target task is decreased due to the use of irrelevant data. Indeed, all research on the negative transfer identify this phenomena with a low level of correlation between tasks even though there were no theoretical results proving this.

Contrary to domain adaptation theory where the classifier is expected to minimize the combined error of both source and target tasks, we will try to improve the performance in target task only. This idea was already applied in [Cao et al., 2010]. In practice, it means that we do not want to minimize explicitly the distance between distributions as in this case we fall into “transfer all” scheme where both tasks can be considered as a single task. Instead, we study what is the optimal alignment between two data sets that leads to an improved performance.

In this chapter, we propose a new unsupervised transfer learning algorithm based on kernel target alignment maximization with application to computer vision problem. To the best of our knowledge, kernel target alignment has never been applied in this context and thus the proposed method presents a novel contribution.

The rest of this chapter is organized as follows: in section 2 we briefly introduce basic notations and describe the approaches used later, in section 3 we introduce our unsupervised transfer learning algorithm. We present theoretical analysis of our approach in section 4. In section 5 the proposed approach will be evaluated. Finally, we will point out some ideas about the future extensions of our method in section 6.

4.2 Preliminary knowledge

In this section, we describe some basic notations and techniques that are used later. We start by introducing the Kernel Target Alignment measure.

4.2.1 Kernel Alignment

Kernel Target Alignment (KTA) is a measure of similarity between two Gram matrices, proposed in [Cristianini and Kandola, 2002] and defined as follows:

$$\hat{A}(K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}}.$$

Frobenius inner product is defined as:

$$\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^{m} k_1(x_i, x_j) k_2(x_i, x_j)$$

where k_1 and k_2 are two kernels, K_1 and K_2 are two corresponding Gram matrices.

As we can see, it essentially measures a cosine between two kernel matrices.

4.2.2 Clustering evaluation criteria

There are two classes of clustering evaluation metrics: internal and external clustering evaluation indexes. Speaking about unsupervised clustering, we can only use internal metrics because they are based on the information intrinsic to the data alone. Among them, the most referenced in literature are the following ones: the Bayesian information criteria, Calinski-Harabasz index, Davies-Bouldin index(DBI), Silhouette index, Dunn index and NIVA index. To estimate the effectiveness of clustering we will use one of the most effective (according to [Rendon et al., 2011]) clustering indexes, the Davies-Bouldin index. This internal evaluation scheme is calculated as follows:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{j:i \neq j} \left(\frac{d(x_i) + d(x_j)}{d(x_i, x_j)} \right)$$

where k denotes the number of clusters, i and j are cluster labels, $d(x_i)$ and $d(x_j)$ are distances to cluster centroids within clusters i and j, $d(x_i, x_j)$ is a measure of separation between clusters i and j. This index aims to identify sets of clusters that are compact and well separated. Smaller value of DBI indicates a "better" clustering solution.

4.3 Our approach

In this section, we describe our method for unsupervised transfer learning, we present an optimization problem related to it and its complexity analysis.

4.3.1 Motivation

The central idea that we will use to overcome the difference between weakly-related tasks is mainly inspired by a very popular approach used in neuroscience called Representation Similarity Analysis (RSA) [Kriegeskorte et al., 2008]. This method suggests that a proper comparison between different activity patterns in human's brain can be encoded and further compared using dissimilarity matrices. For a given brain region, authors interpret activity pattern associated with each experimental condition as a representation. Then, they obtain a representational dissimilarity matrix by comparing activity patterns with respect to all pairs of observations. This approach allows to relate activity patterns between different modalities of brain-activity measurement (e.g., fMRI and invasive or scalp electrophysiology), and between subjects and species. We follow this approach by replacing the dissimilarity matrices of brain activity patterns of different modalities by kernels defined on source and target task samples. Then, we reduce the distance between two distributions by learning a new representation of data for target task in a Reproducing Kernel Hilbert Space (RKHS). This new representation is further factorized using K-NMF in order to find weights of similarities in the transformed instance space. Finally, we use these weights as a "bridge" for transfer learning on the target task.

4.3.2 Kernel target alignment optimization

Let us consider two tasks $\mathcal{T}_S$ and $\mathcal{T}_T$ where the corresponding data samples are given by matrices $X_S = \{x_{s_1}, x_{s_2}, ..., x_{s_n}\} \in \mathcal{R}^m$ and $X_T = \{x_{t_1}, x_{t_2}, ..., x_{t_n}\} \in \mathcal{R}^m$. For the sake of convenience, we will consider data sets X_S and X_T with the same number of instances. This inconvenience can be overcome in two ways: by sub-sampling the bigger data set or by using any kind of a bootstrap to increase the size of the smaller data set.

We start by calculating Gram matrices K_S and K_T for both source and target tasks, for example, using a Gaussian kernel function. Calculating $\hat{A}(K_S, K_T)$ gives us an idea on how correlated the initial kernels are. Small value of $\hat{A}(K_S, K_T)$ means that transfer learning will most likely fail as source and target task distributions are too different. In order to find an intermediate kernel K_{ST} that plays the role of an embedding for both tasks, we apply the kernel alignment optimization to the calculated kernels K_S,K_T that

consists in maximizing unnormalized kernel alignment over α_i:

$$\max \ \langle K_S, K_{ST} \rangle_F$$

$$K_{ST} = \sum_{n=1}^{k} \alpha_n K_n(x_{t_i}, x_{t_j})$$

$$\forall n, \alpha_n \geq 0.$$

Normalization in the cost function is omitted compared to the original definition of kernel alignment in section 2 due to the computational convenience as suggested in [Neumann et al., 2005]. Matrix K_{ST} represents a linear combination of kernel matrices K_n (any arbitrary set of kernel functions can be used) calculated based on X_T. There are several methods which can be used to solve this optimization problem. In our work we use the one that was described in [Cristianini and Kandola, 2002]. The others can be found in [Ramona and David, 2012] and in [Pothin and Richard, 2006]. The proposed optimization problem can be rewritten in the following form:

$$\max \ -\boldsymbol{\alpha}^T (K + \lambda I)\boldsymbol{\alpha} + \mathbf{f}^T \boldsymbol{\alpha}$$

$$s.t. \ \alpha_n \geq 0, \ \forall n = 1..k,$$

where $K(i,j) = \langle K_i, K_j \rangle_F$ and $f(i) = \langle K_i, K_S \rangle_F$. In its current form, the maximization procedure presents a quadratic programming (QP) problem and can be solved using any off-shelf QP solver. For each kernel K_{ST} obtained in the process of alignment optimization, we look for a set of vectors W_{ST} which arises from the K-NMF of K_{ST}:

$$K_{ST} \simeq K_{ST} W_{ST} H_{ST}^T.$$

This matrix is of a particular interest as it represents the weights of similarities that lead to a good reconstruction of K_{ST} in a nonlinear RKHS. Due to the alignment optimization procedure, it naturally consists of adapted weights of an embedding between two tasks. The information contained in W_{ST} can be used further with C-NMF for the target task in order to find more efficient basis vectors that are weighted based on a "good" nonlinear reconstruction of transformed instances. The criteria that we use to evaluate if the obtained reconstruction is "good" or not is DBI. We recall that this

index shows if the clusters are dense and well-separated.

More formally, we look for a matrix W^*_{ST} that minimizes the DBI with respect to target kernel K_T:

$$W^*_{ST} = \arg\min_{W_{ST}} DBI(K_T).$$

We call this matrix the "bridge matrix". Given that K_{ST} was calculated as a linear combination of kernels of X_T and was brought closer in sense of alignment to K_S, W_{ST} naturally incorporate information about geometrical structure of X_S that can help to find better basis vectors in X_T.

4.3.3 Transfer process using the "bridge matrix"

Next step is to perform C-NMF of X_T with the matrix of weights fixed to W^*_{ST}. We use C-NMF as it allows us to reinforce the impact of X_T on the partition matrix H_T.

$$X_T \simeq X_T W^*_{ST} H_T^T.$$

We call this factorization : the Bridge Convex NMF (BC-NMF). Finally, our approach is summarized in Algorithm 1.

Algorithm 1: Bridge Convex NMF (BC-NMF)

input : X_S - source domain data set, X_T - target domain data set, r - number of clusters, n_{iter} - number of iterations

output: $H_{ST}*$ - partition matrix, W^*_{ST} - "bridge matrix"

Initialize K_S, K_T;

$K_S \leftarrow kernel(X_S, X_S, \sigma)$;
$K_T \leftarrow kernel(X_T, X_T, \sigma)$;
$\hat{A}_{init} \leftarrow \hat{A}(K_S, K_T)$;

for $i \leftarrow 1$ **to** n_{iter} **do**
    $K_{ST} \leftarrow alignment\ optimization(K_S, K_T)$;
    $W_{ST} \leftarrow K - NMF(K_{ST}, r)$;

$H^*_{ST} \leftarrow CNMF(X_T, W^*_{ST}, r)$;

4.3.4 Complexity

At each iteration of our algorithm, we perform a K-NMF which makes it quite time consuming when the number of instances is large. On the other hand, it does not depend on the number of features that makes its usage attractive for tasks from high-dimensional spaces. The complexity of K-NMF is of order $n^3 + 2m(2n^2k + nk^2) + mnk^2$ for a Gram matrix $K \in \mathbb{R}^{n \times n}$, where m is a number of iterations used for K-NMF to converge (usually, $m \approx 100$), k - is a desired number of clusters. Then, this expressions should be multiplied by t - the number of iterations needed to optimize the alignment between two kernels. Finally, we obtain the following order of complexity: $t(n^3 + 2m(2n^2k + nk^2) + mnk^2)$.

It should be noted that in real-life tasks the quantity of data in source domain is often greater than in the target one. In order to decrease the computational effort of BC-NMF we propose to proceed a data treatment in the parallel fashion. We split data into several parts and obtain an optimal result for each of them. After that, we use any arbitrary consensus approach (for example, Consensus NMF described in [Li et al., 2007]) to calculate the final result which is close to all the partitions obtained.

4.4 Theoretical analysis

In this section, we present the relationships between KTA and two quantities commonly used in transfer learning and domain adaptation problems, namely: Hilbert Schmidt Independence Criterion (HSIC) [Gretton et al., 2005] and Quadratic Mutual Information(QMI).

4.4.1 Hilbert-Schmidt independence criterion

For readers' convenience, we give the definitions of a mean map and its empirical estimate from Chapter 6 here.

Definition 6. Let $k : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ be a kernel in the RKHS $\mathcal{H}_k$ and $\phi(x) = k(x, \cdot)$. Then, the following mapping

$$\mu[p] = \mathbb{E}_{x \sim p}[\phi(x)]$$

is called a mean map. Its empirical value is given by the following estimate:

$$\mu[X] = \frac{1}{m}\sum_{i=1}^{m}\phi(x_i),$$

where we $X = \{x_1, ..., x_m\}$ is drawn i.i.d. from p.

If $\mathbb{E}_x[k(x,x)] < \infty$ then $\mu[p]$ is an element of RKHS $\mathcal{H}_k$. According to Moore-Aronszajn theorem, the reproducing property of $\mathcal{H}_k$ allows us to rewrite every function $f \in \mathcal{H}_k$ in the following form: $\langle \mu[p], f \rangle_{\mathcal{H}_k} = \mathbb{E}_x[f(x)]$. We now give the definition of HSIC.

Definition 7. Let $k(x,x')$ and $l(y,y')$ be bounded kernels with associated feature maps $\phi : \mathcal{X} \to \mathcal{F}$, $\psi : \mathcal{Y} \to \mathcal{G}$ and let (x,y) and (x',y') be independent pairs drawn from the joint distribution p_{xy}. Then HSIC is defined as follows:

$$HSIC(p_{xy}, \mathcal{F}, \mathcal{G}) = \|\mathcal{C}_{xy}\|^2 = \mathbb{E}_{x,x',y,y'}\left[k(x,x')l(y,y')\right] + \mathbb{E}_{x,x'}\left[k(x,x')\right]\mathbb{E}_{y,y'}\left[l(y,y')\right] + \mathbb{E}_{x,y}\left[\mathbb{E}_{x'}\left[k(x,x')\right]\mathbb{E}_{y'}\left[l(y,y')\right]\right],$$

where $\mathcal{C}_{xy} = \mathbb{E}_{x,y}\left[(k(x,\cdot) - \mu[p]) \otimes (k(y,\cdot) - \mu[q])\right]$ is cross-covariance operator.

Its biased estimate can be calculated from a finite sample using following equation:

$$\widehat{HSIC} = \frac{1}{m^2}tr(KHLH),$$

where $K_{ij} = k(x_i, x_j)$, $L_{ij} = l(y_i, y_j)$ and $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$ is a centering matrix projecting data to a space orthogonal to the vector $\mathbf{1}$.

From this we can see that KTA coincide with the biased estimate of HSIC when centered kernels are used. It shows that KTA is a suitable choice for transfer learning algorithms as its maximization increases iteratively the dependence between source and target distributions. Furthermore, cross-covariance operator has already proved to be efficient when applied in domain adaptation problem for target and conditional shift correction [Zhang et al., 2013].

4.4.2 Quadratic mutual information

Another important point is the equivalence between KTA and Information-Theoretic Learning (ITL) estimators [Principe, 2010]. We define the inner-product between two pdfs p and q as a bivariate function on the set of square intergrable probability density functions:

$$\mathcal{V}(p,q) = \int p(x)q(x)dx.$$

It is easy to show that $\mathcal{V}(p,q)$ is symmetric and non-negative definite and thus according to Moore-Aronszajn theorem, there exists a unique RKHS $\mathcal{H}_v$ associated with $\mathcal{V}(p,q)$. We further define Quadratic Mutual Information (QMI):

$$QMI(x,y) = \iint (p(x,y) - p(x)p(y))^2 dxdy.$$

In order to establish a connection between KTA and QMI, we can use the equivalence between $\mathcal{H}_v$ and $\mathcal{H}_k$ established in [Principe, 2010] through Parzen window estimation [Parzen, 1962]. Parzen window estimator of given probability density functions $p(x)$,$p(y)$ and $p(x,y)$ is defined as follows:

$$\hat{p}(x) = \frac{1}{m}\sum_{i=1}^{m} k_x(x - x_i),\ \hat{p}(y) = \frac{1}{m}\sum_{i=1}^{m} k_y(y - y_i),$$

$$\hat{p}(x,y) = \frac{1}{m}\sum_{i=1}^{m} k_x(x - x_i)k_y(y - y_i).$$

This leads to the following result:

$$\widehat{QMI}(x,y) = \|\hat{p}(x,y) - \hat{p}(x)\hat{p}(y)\|^2 = \frac{1}{m^2}tr(KHLH),$$

where kernel matrices K and L are calculated with respect to Parzen window kernels used for estimation. Once again, we see that KTA with centered kernels is equal to QMI estimation when the Gram matrices K and L are defined as inner-products of Parzen window kernels.

We also note that STC [Dai et al., 2008a] presented in Chapter 3 is based on mutual information maximization. The latter was used to perform co-clustering of target

and auxiliary data with respect to a shared set of features. Another example where mutual information was used for domain adaptation is [Gong et al., 2013b]. Thus, we may conclude that the established relationships allow us to assume that KTA can be effective when used for transfer learning.

4.5 Experimental results

In this section we evaluate our approach and analyze its behavior on Office/Caltech data set.

4.5.1 Baselines and setting

We choose the following baselines to evaluate the performance of our approach:

- C-NMF on target data only;
- K-NMF using each kernel from the set of base kernels used for KTA maximization ("Kernel alone");
- Transfer Spectral Clustering (TSC);
- Bridge Convex-NMF (BC-NMF).

Using C-NMF we can directly factorize matrix X_T as:

$$X_T \simeq X_T W_T H_T^T$$

and consider matrix H_T as an initial partition which could be obtained without taking into account the knowledge from the source task. Accuracy obtained on this partition gives us the "No transfer" value. This particular choice of the baseline can be explained by the fact that our approach is, basically, C-NMF but with a weight matrix W_T learned using kernel alignment optimization. Thus, if we are able to increase the accuracy of classification compared to this baseline it will be only due to the efficiency of our approach.

On the other hand, we also give the maximum value of accuracy achieved for a set of kernels that we use in the optimization of KTA. We chose the following kernel functions: (1) Gaussian kernels with bandwidth varying between 2^{-20} to 2^{20} with multiplicative step-size of 2; homogeneous polynomial kernels with the degree varying from 1 to 3. We call this "kernel alone" value as it presents the result of applying K-NMF to a given kernel without taking into account the auxiliary knowledge. Source task kernel was calculated using linear kernel.

Finally, we compare out method to TSC[1] that according to the experimental results presented in [Jiang and Chung, 2012] outperforms both STC and Bregman multitask clustering (BMC). To define the number of nearest neighbors needed to construct the source and target graphs in TSC, we perform cross-validation for $k \in [5; 100]$ and report the best achieved accuracy value. As suggested in the original paper, we set $\lambda = 3$ and the step length $t = 1$.

The performance of chosen algorithms is evaluated following next criteria:

$$Accuracy = \frac{|\mathbf{x} : \mathbf{x} \in \mathcal{D} \wedge \hat{y}(\mathbf{x}) = y(v)|}{|\mathbf{x} : \mathbf{x} \in \mathcal{D}|},$$

where $\mathcal{D}$ is a data set, and $y(\mathbf{x})$ is the truth label of $\mathbf{x}$ and $\hat{y}(\mathbf{x})$ is the predicted label of $\mathbf{x}$.

4.5.2 Results

In Table 4.1 we can see the results of experimental tests of our approach for transfer between two different domains where bold and underlined numbers stand for the best and second best results respectively. From the results, we can see that our algorithm BC-NMF significantly outperforms TSC in 10 transfer learning scenarios. Furthermore, in some cases TSC achieves lower accuracy values than the "kernel alone" setting. This can be explained by the fact that clusters of the corresponding tasks are not well separable in the initial feature space and thus a nonlinear projection of features to a new RKHS can be beneficial. We also note that using a single kernel from the set of base kernels does not lead to good performance when compared to BC-NMF, while the learned combination of base kernels improves the overall classification accuracy

[1] We used Matlab implementation of TSC provided to us by the authors of the original paper.

Table 4.1: Purity values on Office/Caltech data set obtained using BC-NMF

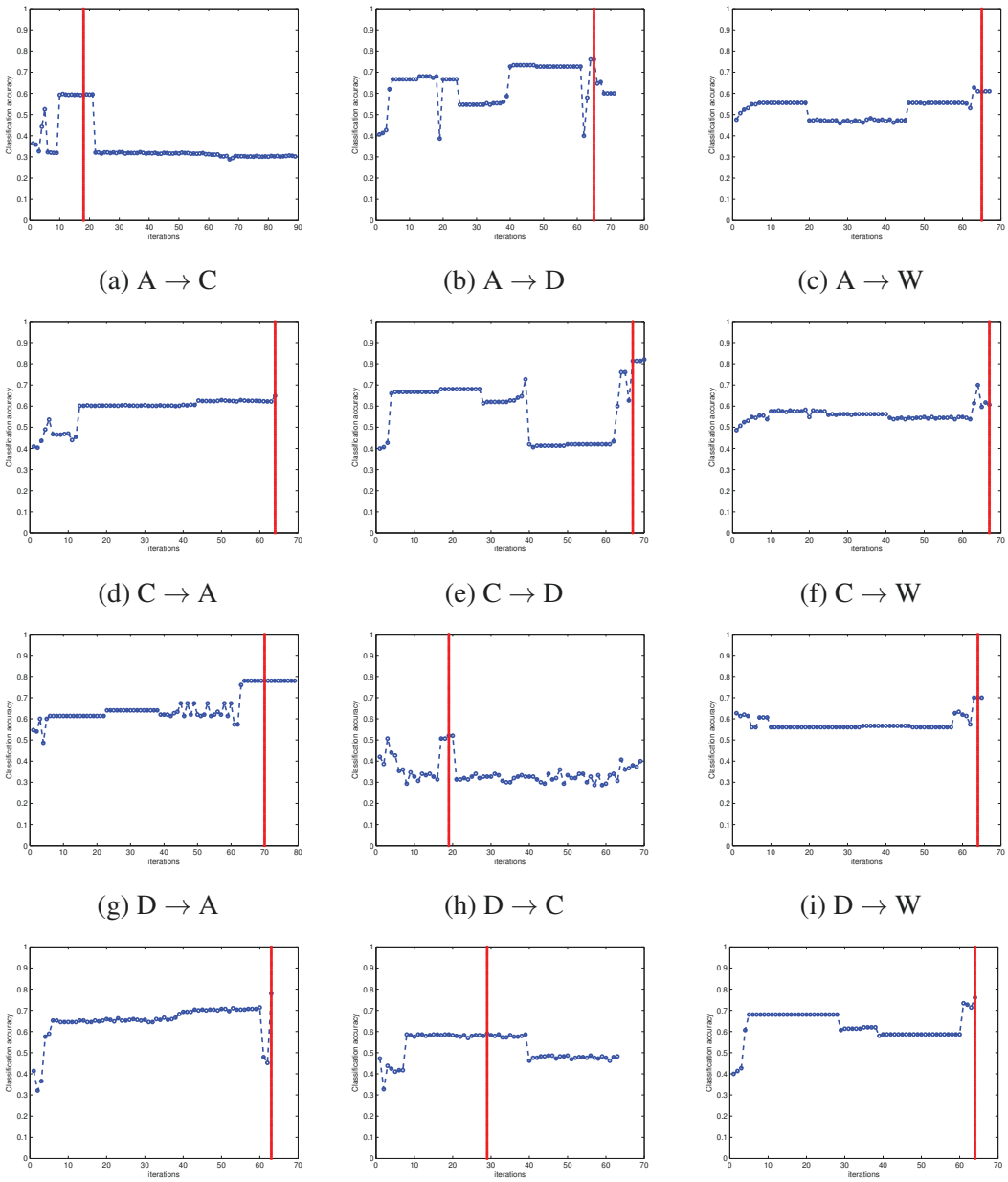
Domain pair	**C-NMF**	**Kernel alone**	**TSC**	**BC-NMF**
C → A	33.24	40.34	43.32	**64.88**
C → W	46.78	56.00	52.54	**60.69**
C → D	46.5	47.33	54.14	**81.33**
A → C	24.89	35.33	46.03	**59.29**
A → W	46.78	56.00	53.22	**60.69**
A → D	46.5	47.33	51.59	**76.0**
W → C	24.89	35.33	**62.71**	58.97
W → A	33.24	40.34	61.36	**77.93**
W → D	46.5	47.33	59.66	**76.0**
D → C	24.89	35.33	**54.14**	52.0
D → A	33.24	40.34	54.78	**78.0**
D → W	46.78	56.00	55.59	**70.0**

considerably. Finally, comparing the obtained results with C-NMF applied to target data only clearly shows that the improved performance is due to the transfer as the only difference between BC-NMF and C-NMF lies in the learned weight matrix W.

In conclusion, we analyze two cases where TSC achieves better clustering results than BC-NMF. We remark that in these two cases Caltech10 plays the role of the target domain. We further notice that the overall performance of both C-NMF and "kernel alone" approaches on Caltech10 is rather weak compared to their performance on Amazon, DSLR and Webcam tasks. We recall that both C-NMF and K-NMF assume that the basis vectors lie in the column space of their instance space while it is not necessarily true. However, if the source task data set is large enough, our approach is still able to improve the performance using the auxiliary knowledge (i.e., A → C) while when it is not the case (i.e., W → C, D → C) BC-NMF may need a larger variety of base kernels to learn a good weight matrix W or more instances from the source data set.

Figure 4.2 presents the learning curves of BC-NMF on each task. We plotted the red bar to indicate where the optimal weight matrix W was obtained. It can be noticed that the proposed strategy to choose W_{ST} does not always lead to the best possible

results but still performs reasonably well.



(a) A → C (b) A → D (c) A → W

(d) C → A (e) C → D (f) C → W

(g) D → A (h) D → C (i) D → W

(j) W → A (k) W → C (l) W → D

Figure 4.2: Algorithm performance on 12 transfer learning scenarios. Each line describes the learning curve of BC-NMF on the corresponding task's pair while the red bar shows where the optimal weight matrix W_{ST} was obtained based on DB index.

4.6 Conclusions and future work

In this chapter, we presented a new method for unsupervised transfer learning. We use kernel alignment optimization in order to minimize the distance between the distributions of source and target tasks. We apply K-NMF to the intermediate kernels obtained during this procedure and look for a weight matrix that reconstructs well the similarity based representation of data. Once this matrix is found, we use it in C-NMF on the target task to obtain the final partition. Our approach was evaluated on benchmark computer vision data sets and demonstrated a significant improvement when compared to some state-of-art methods. We also showed how KTA maximization can be related to HSIC and QMI optimization. The established relationships allow us to conclude that the use of KTA for transfer learning is justified from both theoretical and practical points of view. One of the inconvenients of our approach is that it is quite time consuming. Nevertheless, this issue can be overcome as discussed in section 3.

In future, we will extend our work in the multiple directions. First of all, we will start by creating a multi-task version of our method. This can be done in the same fashion but with the only difference: firstly, we search for an optimal Gram matrix for each pair of tasks, then we will use the simultaneous non-negative matrix factorization [Badea, 2008] to find the common "bridge matrix" that captures the knowledge from all tasks. Multi-task version of our algorithm can be very important because it could show us the participation of each task in overall improvement. Secondly, it would be useful to derive bounds for classification error. This problem, however, is complicated as there is no statistical theory that can be used in unsupervised setting in the same way how it can be done for supervised and semi-supervised learning.

Chapter 5

Non-negative Embedding for Fully Unsupervised Domain Adaptation

5.1 Introduction

In this chapter, we would like to present an approach that makes use of the fact that some data are intrinsically non-negative and to show that preserving this non-negativity can be beneficial for classification while learning new feature representations. Indeed, it makes sense as the majority of transfer learning algorithms are usually applied to image classification and object recognition data sets where data represented by color frequencies are naturally non-negative. Our work is related to a couple of unsupervised domain adaptation methods where the goal is to find an intermediate representation of data of a source domain that can be used further in combination with a target domain. A common approach to do that is to look for a new projection of data in the corresponding space. To this end, the application of PCA was widely investigated and used in order to find a common space where the divergence between marginal distributions of two domains is minimized [Chen et al., 2009; Pan et al., 2009]. According to the theory of domain adaptation, classification error of the target task is bounded by the divergence between distributions of each domain so this idea is theoretically justified. Subspace approaches were also widely used for domain adaptation and transfer learning in, for example, [Fernando et al., 2013; Grauman, 2012].

Our approach differs from the above mentioned works in two principal ways. First one is that we seek to find a non-negative embedding of basis vectors for two domains so that we could benefit from the fact that some data is intrinsically non-negative. To this end, our approach is similar to methods presented in [Chen and Zhang, 2013a; Long et al., 2012; Zhuang et al., 2013]. The main difference, however, is that they use NMF techniques for matching between objects of two domains and are usually applied for text-classification via Tri-NMF. Our approach instead learns a shared dictionary and its application is not limited to text classification. Second main difference is that we do not assume that we have labels in source domain - we use a shared set of basis vectors simultaneously with performing clustering of the target domain data. This type of setting is usually called "self-taught clustering" [Dai et al., 2008b] and is an instance of unsupervised transfer learning. In our work, we would like to show that this paradigm can be used as a complementary for unsupervised domain adaptation approaches.

5.2 Unsupervised domain adaptation via non-negative embedding

We assume that we have two sets of unlabeled data $X_S \in \mathbb{R}^{m \times n_s}$ and $X_T \in \mathbb{R}^{m \times n_t}$ that correspond to source task data and target task data respectively. We denote their marginal distributions by $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$. First step of our approach consists in retrieving non-negative basis of each task by applying Projective NMF to X_S and X_T.

5.2.1 Projective NMF

Orthogonal Projective NMF (OPNMF) introduced in [Yang and Oja, 2010] minimizes the following cost function:

$$\min_U \; J = \|X - UU^T X\|_F^2$$

$$s.t.\; U^T U = I, U \geq 0,$$

where

- $X \in \mathbb{R}^{m \times n}$ is an input data matrix
- columns of $U \in \mathbb{R}^{m \times k}$ can be considered as basis vectors
- k is the desired number of basis vectors

As shown in [Yang and Oja, 2010], Orthogonal Projective NMF solves PCA problem with non-negative constraints when Oja's rule is applied during the optimization procedure.

At first step we will apply OPNMF to matrices X_S, X_T and we will fix $k = m$:

$$X_S \simeq U_S U_S^T X_S, X_S \in \mathbb{R}_+^{m \times n_s}, U_S \in \mathbb{R}_+^{m \times d^*},$$

$$X_T \simeq U_T U_T^T X_T, X_T \in \mathbb{R}_+^{m \times n_t}, U_T \in \mathbb{R}_+^{m \times d^*}.$$

The resulting matrices U_S and U_T are m non-negative eigenvectors of X_S and X_T. We cannot use source eigenvectors for target task directly as we are interested in features that are aligned with target task basis vectors. To choose them we will use subspace

disagreement measure (SDM) presented in [Grauman, 2012] to define d^* using non-negative principal components.

To compute SDM, we combine the data sets into one data set X_{S+T} and compute its subspace U_{S+T} using OPNMF. Intuitively, if the two data sets are similar, then all three subspaces should not be too far away from each other on the Grassmannian. The SDM captures this notion and is defined in terms of the principal angles:

$$D(d) = \frac{1}{2}[\sin \alpha_d + \sin \beta_d],$$

where α_d denotes the d-th principal angle between the U_S and U_{S+T} and β_d between U_T and U_{S+T}.

Then optimal number of basis vectors d is defined as follows:

$$d^* = \min\{d | D(d) = 1\}.$$

After applying this procedure to U_S we obtain a matrix $U_{S_{d^*}}$ which we will used as an aligned subspace to generate a non-negative embedding.

5.2.2 Non-negative embedding generation

Using the notations defined above let us consider the following cost function:

$$\min \ J_{une} = \|U_{S_{d^*}} - U_* H_S^T\|_F^2 + \|U_T - U_* H_T^T\|_F^2 + \|X_T - U_* H_*^T\|_F^2$$

$$s.t. \ U_*, H_S, H_T, H_* \geq 0,$$

$$H_S^T H_S = I, H_T^T H_T = I, H_*^T H_* = I.$$

Here first two terms share the same factor U_* - a matrix of basis vectors that can be seen as a shared subspace of $U_{S_{d^*}}$ and U_T. Third term is just a standard NMF applied to matrix X_T with prototype matrix fixed to U_*. Final result is given by matrix H_*.

Following the idea from [Fernando et al., 2013] we will now define a similarity function based on the transition matrix between $U_{S_{d^*}}$ and U_T. We will further prove that this function is consistent.

First, we observe that $U_* = U_T H_T$. By plugging this expression into the first term

we have $U_{S_{d^*}} = U_T H_T H_S^T$. It means that the transition matrix M^* that aligns $U_{S_{d^*}}$ with U_T is defined as follows:

$$U_{S_{d^*}} = U_T M_*.$$

Consequently, the similarity function for two elements $x_s \in X_S$ and $x_t \in X_T$ is given by $S(x_s, x_t) = x_s U_{S_{d^*}} M_* (x_t U_T)^T = x_s U_* U_*^T x_t^T$.

Now let us define $\tilde{C}_n$ to be the covariance matrix of a sample D of size n drawn i.i.d. from a given distribution and $\tilde{C}$ its expected value over that distribution. Let k_+^X be a non-negative rank of a matrix X, i.e. $rank_+(X) = k_+^X$. We will now present two theorems that we will use further to prove the consistency theorem of similarity function S.

Theorem 5.1. *[Zwald and Blanchard, 2005] Let B be s.t. for any vector x, $\|x\| \leq B$, let $X_{\tilde{C}}^d$ and $X_{\tilde{C}_n}^d$ be the orthogonal projectors of the subspaces spanned by first d eigenvectors of $\tilde{C}$ and $\tilde{C}_n$. Let $\lambda_1 > \lambda_2 > ... > \lambda_d > \lambda_{d+1} \geq 0$ be the first d+1 eigenvalues of $\tilde{C}$ then for any $n \geq (\frac{4B}{\lambda_d - \lambda_{d+1}} \left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right))^2$ with probability at least $1 - \delta$ we have:*

$$\|X_{\tilde{C}}^d - X_{\tilde{C}_n}^d\| \leq \frac{4B}{\sqrt{n}(\lambda_d - \lambda_{d+1})} \left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right).$$

Theorem 5.2. *[Moitra, 2012] For a given matrix X and $\delta > 0$ there exists a nearly optimal algorithm under the Exponential Time Hypothesis [Impagliazzo and Paturi, 2001] that returns factors $\tilde{U}$ and $\tilde{H}$ that are δ close to U and H where $X = UH$ is a non-negative matrix factorization of rank r where $k_+^X \leq r$.*

Using these two theorems we will now prove the following lemma:

Lemma 5.3. *Let B be s.t. for any vector x, $\|x\| \leq B$, let $U_{\tilde{C}}^d$ and $U_{\tilde{C}_n}^d$ be the orthogonal projectors of the subspaces spanned by first d eigenvectors of $\tilde{C}$ and $\tilde{C}_n$. Let H be a matrix arising from the non-negative matrix factorization of $U_{\tilde{C}_n}^d$. Let $\lambda_1 > \lambda_2 > ... > \lambda_d > \lambda_{d+1} \geq 0$ be the first d+1 eigenvalues of $\tilde{C}$ then for any*

$n \geq (\frac{4B}{\lambda_d - \lambda_{d+1}} \left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right)^2$ *with probability at least* $1 - \delta$ *we have:*

$$\|U^d_{\tilde{C}}\tilde{H} - U^d_{\tilde{C}_n}H\| \leq \delta\sqrt{m} + \sqrt{d}\frac{4B}{\sqrt{n}(\lambda_d - \lambda_{d+1})}\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right).$$

Proof.

$$\begin{aligned}\|U^d_{\tilde{C}}\tilde{H} - U^d_{\tilde{C}_n}H\| = \\ &= \|U^d_{\tilde{C}}\tilde{H} - U^d_{\tilde{C}_n}H + U^d_{\tilde{C}_n}\tilde{H} - U^d_{\tilde{C}_n}\tilde{H}\| = \\ &\leq \|U^d_{\tilde{C}_n}\|\|\tilde{H} - H\| + \|\tilde{H}\|\|U^d_{\tilde{C}} - U^d_{\tilde{C}_n}\| \\ &\leq \delta\sqrt{m} + \sqrt{d}\frac{4B}{\sqrt{n}(\lambda_d - \lambda_{d+1})}\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right).\end{aligned}$$

□

$\|U^d_{\tilde{C}_n}\|$ is bounded by $\sqrt{m}$ as the eigenvectors are normalized, $\|\tilde{H}\|$ is bounded by $\sqrt{d}$ due to the constraints in the cost function. Other two terms are bounded using Theorem 1 and 2.

The theorem for the consistency of $S(x_s, x_t)$ is stated as follows:

Theorem 5.4. *Let* $U^d_{S_n}$ *and* $U^m_{T_n}$ *be the d- and m- dimensional projection operators built from the source and target samples of size* n_S *and* n_T*. Let* U^d_S *(resp.* U^m_T*) the expected value of* $U^d_{S_n}$ *(resp.* $U^m_{T_n}$*) associated with d+1 (resp. m+1) eigenvalues* $\lambda^S_1 > \lambda^S_2 > ... > \lambda^S_d > \lambda^S_{d+1} \geq 0$ *(resp.* $\lambda^T_1 > \lambda^T_2 > ... > \lambda^T_m > \lambda^T_{m+1} \geq 0$*). Let* H_S *(resp.* H_T*) be a non-negative matrix arising from the non-negative matrix factorization of* U^d_S *(resp.* U^m_T*). Then with probability at least* $1 - \delta$ *we have:*

$$\|U^d_S M^* U^m_T - U^d_{S_n} M^*_n U^m_{T_n}\| \leq \delta(m\sqrt{d} + d\sqrt{d}) + \frac{4dB\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right)(\sqrt{m} + \sqrt{d})}{\sqrt{n_S}(\lambda^S_d - \lambda^S_{d+1})\sqrt{n_T}(\lambda^T_m - \lambda^T_{m+1})},$$

where $M^*_n = H_S H'_T$.

Proof.

$$\begin{aligned}\|U_S^d M^* U_T^m - U_{S_n}^d M_n^* U_{T_n}^m\| &= \|U_S^d \tilde{H}_S \tilde{H_T}^T U_T^m - U_{S_n}^d H_S H_T^T U_{T_n}^m + U_S^d \tilde{H}_S H_T^T U_{T_n}^m - U_S^d \tilde{H}_S H_T^T U_{T_n}^m\| \\ &\leq \|U_S^d \tilde{H}_S\| \|\tilde{H_T}^T U_T^m - H_T^T U_{T_n}^m\| + \|H_T^T U_{T_n}^m\| \|U_S^d \tilde{H}_S - U_{S_n}^d H_S\| \\ &\leq \delta(m\sqrt{d} + d\sqrt{d}) + \frac{4dB\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right)(\sqrt{m} + \sqrt{d})}{\sqrt{n_S}(\lambda_d^S - \lambda_{d+1}^S)\sqrt{n_T}(\lambda_m^T - \lambda_{m+1}^T)}.\end{aligned}$$

□

5.3 Multiplicative update rules

In order to derive the multiplicative update rules for our method, we use the approach presented in Chapter 2.

5.3.1 Fully unsupervised non-negative embedding (UNE)

For the cost function from section 3

$$\min\ J_{une} = \|U_{S_{d^*}} - U_* H_S^T\|_F^2 + \|U_T - U_* H_T^T\|_F^2 + \|X_T - U_* H_*^T\|_F^2$$

and taking into account orthonormalization and non-negativity constraints

$$\begin{aligned}&U_*, H_S, H_T, H_* \geq 0 \\ &H_S^T H_S = I, H_T^T H_T = I, H_*^T H_* = I\end{aligned}$$

we obtain the following update rules:

$$
\begin{aligned}
H_T &= H_T \circledast \frac{U_*^T U_{S_{d^*}}^T + H_T}{U_*^T U_* H_T + H_T H_T^T H_T}, \\
H_S &= H_S \circledast \frac{U_* U_T^T + H_S}{U_*^T U_* H_S + H_S H_S^T H_S}, \\
U_* &= U_* \circledast \frac{U_T H_T^T + U_{S_{d^*}} H_S^T + X_T H_*^T}{U_* H_T H_T^T + U_* H_S H_S^T + U_* H_* H_*^T}, \\
H_* &= H_* \circledast \frac{U_*^T X_T + H_*}{U_*^T U_* H_* + H_* H_*^T H_*}.
\end{aligned}
$$

This optimization problem is not convex in all arguments thus presented update rules usually converge to a local minima. The non-increasing property of these update rules can be proved using an auxiliary function defined as follows: $G(h, h')$ is an auxiliary function of $F(h)$ if $G(h, h') \geq F(h)$ and $G(h, h) = F(h)$.

We do not give a proof here as it represents a typical result from NMF optimization theory and is rather technical once the auxiliary function for a given optimization problem was found.

Finally, our approach is summarized in Algorithm 2.

Algorithm 2: Non-negative embedding generation for fully unsupervised domain adaptation

input : X_S - source domain data set, X_T - target domain data set, n - number of clusters, n_{iter} - number of iterations

output: H_* - partition matrix for X_T, U^* - non-negative embedding

Initialize U_S, U_T, H_S, H_T, U^*, H_*;

$U_S \leftarrow OPNMF(X_S, n)$;
$U_T \leftarrow OPNMF(X_T, n)$;
$d^* \leftarrow SDM(U_S, U_T)$;
$U_S^{d^*} \leftarrow U_S(1 : d^*, :)$;

for $i \leftarrow 1$ **to** n_{iter} **do**

$$H_T = H_T \circledast \frac{U_*^T U_{S_{d^*}}^T + H_T}{U_*^T U_* H_T + H_T H_T^T H_T};$$
$$H_S = H_S \circledast \frac{U_* U_T^T + H_S}{U_*^T U_* H_S + H_S H_S^T H_S};$$
$$U_* = U_* \circledast \frac{U_T H_T^T + U_{S_{d^*}} H_S^T + X_T H_*^T}{U_* H_T H_T^T + U_* H_S H_S^T + U_* H_* H_*^T};$$
$$H_* = H_* \circledast \frac{U_*^T X_T + H_*}{U_*^T U_* H_* + H_* H_*^T H_*};$$

$Sim(x_s, x_t) \leftarrow x_s U_* U_*^T x_t$;

5.4 Experimental results

In this section we will evaluate our approach and compare it to some currently published state-of-art approaches using Office/Caltech data set[1].

5.4.1 Baseline methods

Following a recently published paper on visual domain adaptation [Long et al., 2014b] we will use the same baseline methods to evaluate our method, namely:

- 1-Nearest Neighbor classifier (NN);

[1]We note that z-score normalization can lead to negative values when applied to initial data sets while we suppose that preserving non-negativity can be important for classification. Contrary to the description in Chapter 3, we does not use the z-score normalization for our approach.

- Principal component analysis (PCA);
- Joint feature selection and subspace learning (FSSL) [Gu et al., 2011];
- Transfer Component Analysis (TCA);
- Geodesic flow kernel (GFK);
- Transfer joint matching (TJM);
- Subspace alignment (SA).

The most common approach used to compare domain adaptation algorithms is to learn a simple NN classifier using target aligned source representation with a small amount of labels to further use it for classification of the unlabeled target task. For fully unsupervised version of our approach (UNE) we do not use any labels at all.

We will use accuracy to evaluate the performance of chosen algorithms. It is defined as:

$$Accuracy = \frac{|\mathbf{x} : \mathbf{x} \in \mathcal{D} \wedge \hat{y}(\mathbf{x}) = y(v)|}{|\mathbf{x} : \mathbf{x} \in \mathcal{D}|},$$

where $\mathcal{D}$ is a test data set, and $y(\mathbf{x})$ is the truth label of $\mathbf{x}$ and $\hat{y}(\mathbf{x})$ is the predicted label of $\mathbf{x}$.

5.4.2 Classification results

The classification accuracies obtained on Office/Caltech data set are presented in Table 5.1. We can see that our fully unsupervised approach outperforms all other in 4 domain adaptation scenarios. It is particularly interesting to observe that it performs exceptionally well when the source data set is big enough (A,C) and the target one is small (W,D).

On the other hand, UNE fails to reach a good performance compared to other baselines for two scenarios when both source and target domains have a very small amount of instances per class (W,D). It has been already observed before [Gong et al., 2013b] that selecting landmarks requires a sufficient amount of data even in supervised setting. When working in a fully unsupervised manner, learning a discriminant dictionary in the absence of a sufficient number of data becomes barely possible. In this case, discovering patterns in data requires at least some supervision. In all other scenarios, the

Table 5.1: Purity values on Office/Caltech data set obtained using UNE

Domain pair	NN	PCA	FSSL	TCA	GFK	TJM	SA	UNE
C → A	23.70	36.95	35.88	45.82	41.02	**46.76**	39.0	33.86
C → W	25.76	32.54	32.32	30.51	40.68	38.98	36.8	**47.32**
C → D	25.48	38.22	37.53	35.67	38.85	44.59	39.6	**48.15**
A → C	26.00	34.73	33.91	40.07	40.25	**39.45**	35.3	24.86
A → W	29.83	35.59	34.35	35.25	38.98	42.03	38.6	**46.37**
A → D	25.48	27.39	26.37	34.39	36.31	45.22	37.6	**50.19**
W → C	19.86	26.36	25.85	29.92	30.72	30.19	**32.3**	24.74
W → A	22.96	29.35	29.53	28.81	29.75	29.96	**37.4**	33.88
W → D	59.24	77.07	76.79	85.99	80.89	**89.17**	80.3	49.04
D → C	26.27	29.65	27.89	32.06	30.28	31.43	**32.4**	24.15
D → A	28.50	32.05	30.61	31.42	32.05	32.78	**38.0**	34.18
D → W	63.39	75.93	74.99	**86.44**	75.59	85.42	83.6	48.34

performance of the proposed approach is close to the best baseline. According to domain adaptation theory from Chapter 6, the distance between distributions has to be minimized for domain adaptation algorithm to succeed. We plot the evolution of the MMD distance between projected data from source and target task in Figure 5.1.

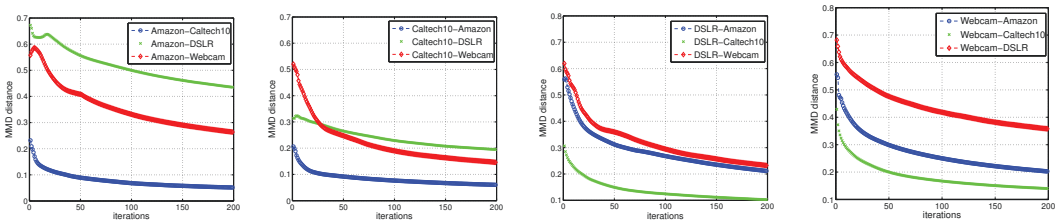


Figure 5.1: MMD distance on 12 cross-domain visual adaptation scenarios where the source task is **A**, **C**, **D**, **W** from left to right.

As we can see, our approach minimizes properly the distance between distributions. This justifies its use from the theoretical point of view.

Finally, we can observe that our all three best approaches can be seen as a comple-

mentary to each other as none of them totally outperforms the others.

5.5 Conclusions and future work

In this chapter, we presented a new approach for fully unsupervised domain adaptation. We create a non-negative embedding as a shared feature's space of two aligned sets of non-negative basis vectors. This embedding is then used as a prototype matrix for NMF clustering. The proposed approach is very simple, intuitive and easy to implement. We evaluated our method on a famous visual domain adaptation benchmark and observed that despite its simplicity it can outperform state-of-the-art methods in 4 different scenarios while being close to the best baseline in 6 others. We also presented a consistency theorem of the similarity function built using the proposed non-negative embedding.

To evaluate our approach from theoretical point of view, we prove a theorem that relates the source and target domain errors using kernel embeddings of distributions functions. We show that our approach agrees well with the theory and iteratively minimizes the distance between distributions. Furthermore, the latter can be estimated efficiently using unbiased linear time estimator.

In future our approach can be extended in multiple directions. First of all, it would be interesting to develop a multi-task version of our approach. This can be done by by replacing the first term in cost function with a sum of factorizations that correspond to different source domains. In this case, the question of how to weight this terms arises as the distance between their distributions with respect to target domain distribution may vary a lot.

Chapter 6

Generalization Bounds for Domain Adaptation using Hilbert-Schmidt Embeddings

6.1 Introduction

From a theoretical point of view, for the first time the domain adaptation problem was investigated in [Ben-David et al., 2007]. The authors of this paper focus on the domain adaptation problem following the uniform convergence theory and consider the 0-1 loss function in the setting of binary classification. They define the hypothesis and empirical errors as risk functions to further derive bounds that relate them using the $\mathcal{A}$-divergence [Kifer et al., 2004]. [Ben-David et al., 2007] show that the key factors that define the potential success of domain adaptation is the divergence between tasks' distributions and the existence of the ideal joint hypothesis on both source and target domains. We present the main results of this work along with the restrictions in Section 2. The analysis presented in the original paper was further extended in [Blitzer et al., 2008] who give uniform convergence bounds for algorithms that minimize a convex combination of source and target tasks' errors [1].

In the follow-up work [Ben-David et al., 2010b], the impossibility theorems for domain adaptation problem based on 0-1 loss were proved and illustrated using a handful of examples. The main domain adaptation assumptions studied in this paper are: (1) the source and target distributions are close; (2) there exist a hypothesis with low error on both of them; (3) the labeling function does not change between the training and test data. They concluded that neither of the assumption combinations (1) and (3) nor (2) and (3) suffice for successful domain adaptation.

Another important paper that focuses on domain adaptation theoretical guarantees is [Mansour et al., 2009]. This paper extended the bounds of [Ben-David et al., 2010a] to a broader class of convex loss functions and introduced new regularization-based domain adaptation algorithms based on the discrepancy distance. Authors showed that the proposed discrepancy distance can be estimated from finite samples and that it can be used to derive data-dependent Rademacher complexity learning bounds. In [Cortes and Mohri, 2014], the discrepancy distance was used to prove pointwise loss guarantees for kernel-based regularization algorithms, including kernel ridge regression, support vector machines (SVMs), or support vector regression (SVR). These guarantees were further used to design efficient empirical discrepancy minimization algorithms

[1] We will further refer only to the paper [Ben-David et al., 2010a] as it presents an extended and full version of the preliminary results published in both [Ben-David et al., 2007] and [Blitzer et al., 2008].

for large-scale problems that can be casted as a SDP problem.

Finally, a new family of generalization bounds based on the property of robust algorithms was introduced in [Mansour and Schain, 2014]. The notion of "algorithmic robustness" was first presented in [Xu and Mannor, 2010] in order to measure the sensitivity of a given algorithm to changes in training data. This new approach for generalization bounds was used to design new robust domain adaptation SVM-based algorithms for classification and regression.

Multi-task learning is an another field of machine learning related to domain adaptation. The most exhaustive study on this subject is presented in [Crammer et al., 2008]. The main difference of multi-task learning from domain adaptation is that it assumes the same distribution over the sources and the presence of labeled data in the target domain.

In this chapter, we propose new generalization bounds for domain adaptation from the kernel methods perspective and point out some of their benefits by comparing them with previous bounds. Our motivation is three-fold: (1) kernel methods allow to overcome the divergence between two distributions by learning a feature map that projects data to a shared latent space; (2) there is a natural distance that can be defined between kernel embeddings of probability distributions that enjoys the existence of an efficient estimator and is directly linked to the optimal transport problem; (3) kernel methods applied in domain adaptation have already shown a significant success in many real-world applications.

6.2 Related work

In this section we describe two main approaches to derive domain adaptation learning bounds that are closely related to our work. These are:

- generalization bounds using $\mathcal{H}\Delta\mathcal{H}$ distance from [Ben-David et al., 2010a];
- generalization bounds based on discrepancy distance from [Mansour et al., 2009].

6.2.1 Domain adaptation based on $\mathcal{H}\Delta\mathcal{H}$ distance

In [Ben-David et al., 2010a] the problem of domain adaptation is formalized as follows: we define a domain as a pair consisting of a distribution $\mathcal{D}$ on inputs $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to [0, 1]$, which can have a fractional (expected) value when labeling occurs nondeterministically. Initially, we consider two domains, a source domain and a target domain. We denote by $\langle \mathcal{D}_{\mathcal{S}}, f_S \rangle$ the source domain and $\langle \mathcal{D}_{\mathcal{T}}, f_T \rangle$ the target domain. A hypothesis is a function $h : \mathcal{X} \to \{0, 1\}$.

Definition 8. The probability according to the distribution $\mathcal{D}_{\mathcal{S}}$ that a hypothesis h disagrees with a labeling function f (which can also be a hypothesis) is defined as

$$\epsilon_S(h, f) = \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{S}}}[|h(x) - f(x)|].$$

We can see that the source error is an expectation of disagreement between source and hypothesis labeling functions. The following theorem gives the bound that relates the source and target tasks' error functions.

Theorem 6.1. *[Ben-David et al., 2010a] For a hypothesis h,*

$$\epsilon_T(h, f_T) \leq \epsilon_S(h, f_S) + d_1(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}) + \min\{\mathbb{E}_{\mathcal{D}_{\mathcal{S}}}[|f_S(x) - f_T(x)|], \mathbb{E}_{\mathcal{D}_{\mathcal{T}}}[|f_T(x) - f_S(x)|]\},$$

where $d_1(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})$ is a total variation distance between distributions $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$.

The authors then define the symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$ as a set of hypotheses

$$g \in \mathcal{H}\Delta\mathcal{H} \iff g(x) = h(x) \oplus h'(x)$$

for some $h, h' \in \mathcal{H}$, where $\oplus$ stands for XOR operation.

The following theorem gives a bound on the target task error using the divergence measure defined above.

Theorem 6.2. *[Ben-David et al., 2010a] Let $\mathcal{H}$ be a hypothesis space of Vapnik-Chervonenkis (VC) dimension d. If $\mathcal{U}_{\mathcal{S}}$, $\mathcal{U}_{\mathcal{T}}$ are unlabeled samples of size m' each, drawn independently from $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$ respectively, then for any $\delta \in (0, 1)$ with prob-*

ability at least $1-\delta$ *(over the choice of samples), for every* $h \in \mathcal{H}$ *:*

$$\epsilon_T(h, f_T) \leq \epsilon_S(h, f_S) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{2}{\delta})}{m'}} + \lambda,$$

where λ *is the combined error of the ideal hypothesis* $h*$ *that minimizes* $\epsilon_S(h) + \epsilon_T(h)$*.*

In order to prove this theorem the authors used the fact that $|\epsilon_T(h, h') - \epsilon_S(h, h')| \leq \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})$.

Last important result that we would like to cite here is a theorem that relates the minimizer of the combined error defined as a convex combination of source and target errors to the target task minimizer.

Theorem 6.3. *[Ben-David et al., 2010a] Let* $\mathcal{H}$ *be a hypothesis space of VC dimension* d*. If* $\mathcal{U}_S$*,* $\mathcal{U}_T$ *are unlabeled samples of size* m' *each, drawn independently from* $\mathcal{D}_{\mathcal{S}}$ *and* $\mathcal{D}_{\mathcal{T}}$ *respectively. Let* S *be a labeled sample of size* m *generated by drawing* βm *points from* $\mathcal{D}_{\mathcal{T}}$ *(*$\beta \in [0, 1]$*) and* $(1-\beta)m$ *points from* $\mathcal{D}_{\mathcal{S}}$ *and labeling them according to* f_S *and* f_T*, respectively. If* $\hat{h} \in \mathcal{H}$ *is the empirical minimizer of* $\hat{\epsilon}_\alpha(h)$ *on* S *and* $h_T^* = \min_{h\in\mathcal{H}} \epsilon_T(h)$ *then for any* $\delta \in (0, 1)$*, with probability at least* $1-\delta$ *(over the choice of samples),*

$$\epsilon_T(\hat{h}, f_T) \leq \epsilon_T(h_T^*, f_T) + c_1 + c_2,$$

where

$$c_1 = 4\sqrt{\frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta}}\sqrt{\frac{2d\log(2(m+1)) + 2\log(\frac{8}{\delta})}{m}},$$

$$c_2 = 2(1-\alpha)\left(\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{8}{\delta})}{m'}} + \lambda\right).$$

6.2.2 Domain adaptation based on discrepancy distance

Similar to the previous subsection, we present the main results of [Mansour et al., 2009]. We start with the definition of discrepancy distance.

Definition 9. Let H be a set of function mapping X to Y and let $L : Y \times Y \to \mathbb{R}_+$ define a loss function over Y. The discrepancy distance $disc_L$ between two distributions $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ over X is defined by

$$disc_L(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T}) = \max_{h,h' \in H} \left| \mathbb{E}_{\mathcal{D}_\mathcal{S}}[L(h'(x), h(x))] - \mathbb{E}_{\mathcal{D}_\mathcal{T}}[L(h'(x), h(x))] \right|,$$

where Y is a label set.

We note that for the 0-1 classification loss, the discrepancy distance coincides with $\mathcal{H}\Delta\mathcal{H}$ divergence and suffers from the same computational restrictions as the latter. Using this definition, the analogue of Theorem 6.2 can be proved.

Theorem 6.4. *Assume that the loss function L is symmetric and obeys the triangle inequality. Then, for any hypothesis $h \in H$ the following holds*

$$\epsilon_T(h, f_T) \leq \epsilon_T(h^*_T, f_T) + \epsilon_S(h, h^*_S) + disc_L(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T}) + \epsilon_S(h^*_T, h^*_S),$$

where $h^*_S = \min_{h \in \mathcal{H}} \epsilon_S(h)$.

As pointed out by the authors, the proposed bound is not directly comparable to Theorem 3, nevertheless, the comparison made in this paper showed they can be more tight in some plausible scenarios. The important difference between two theorems lies in the way how they estimate the corresponding distance. While Theorem 6.2 relies on Sauer's lemma to bound the true $\mathcal{A}$-divergence by its empirical counterpart, $disc_L$ is estimated using the Rademacher classification bounds. We now give a corollary that shows how the discrepancy distance can be estimated from finite samples.

Corollary 6.5. *Let $\mathcal{H}$ be a hypothesis set bounded by some M for some loss function L_q: $L_q(h, h') \leq M$, for all $h, h' \in H$. If $\mathcal{U}_S$, $\mathcal{U}_T$ are samples of size m and n drawn independently from $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ respectively. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ (over the choice of samples),*

$$disc_L(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T}) \leq disc_L(\mathcal{U}_\mathcal{S}, \mathcal{U}_\mathcal{T}) + 4q \left(\hat{\mathfrak{R}}_m(H) + \hat{\mathfrak{R}}_n(H) \right) + 3M \left(\sqrt{\frac{\log(\frac{4}{\delta})}{2m}} + \sqrt{\frac{\log(\frac{4}{\delta})}{2n}} \right),$$

where $\hat{\mathfrak{R}}_m(H)$ and $\hat{\mathfrak{R}}_n(H)$ denote empirical Rademacher complexity of H over samples $\mathcal{U}_S$ and $\mathcal{U}_T$ respectively.

These theorems from were further used to provide guarantees for kernel-based regularization algorithms that allow to minimize the discrepancy distance.

6.2.3 Our contributions

From the results seen above, we can outline two main restrictions of domain adaptation theory that were mentioned in [Ben-David et al., 2010a]:

- all key theorems in [Ben-David et al., 2010a] assume that the hypothesis space is of VC dimension.
- there is no calculation guarantees for the estimation of the divergence in $\mathcal{H}\Delta\mathcal{H}$ space.

We now discuss more in detail each of these two statements. First one comes from the fact that authors followed the Vapnik-Chervonenkis theory [Vapnik, 1995] and used the Sauer's lemma in order to bound the difference between true and empirical $\mathcal{A}$-divergence (Theorem 3.4, [Kifer et al., 2004]). Sauer's lemma gives an upper bound for growth function of a given hypothesis class $\mathcal{H}$ when the hypothesis space is of Vapnik-Chervonenkis dimension. Our idea is to overcome this issue by bounding the Rademacher complexity of the RKHS space directly using kernel embeddings of probability distributions without applying the Sauer's lemma.

Second issue lies in the definition of the empirical estimate proposed in the original work as minimizing the error for most reasonable hypothesis classes is an intractable problem (Lemma 2, [Ben-David et al., 2010a]). One may want to have an efficient estimate with a proved computational guarantees to calculate the divergence between two distributions. We show how the empirical estimator of $d_{\mathcal{H}\Delta\mathcal{H}}$ can be replaced with an unbiased empirical estimate of the maximum mean discrepancy (MMD) distance that can be computed in quadratic time.

The results from [Mansour et al., 2009] overcome one of the limitations of [Ben-David et al., 2010a] by extending the original theorems to a larger class of loss functions, however, the problem of efficient estimation of the discrepancy distance remained unaddressed. Furthermore, the results in both [Mansour et al., 2009] and [Cortes and Mohri, 2014] do not present the generalization bounds for the case of

combined error of source and target tasks similar to the ones proposed in [Ben-David et al., 2010a].

Our work is similar to [Mansour et al., 2009] in that it also uses Rademacher complexity learning bounds to estimate the distance from finite samples. On the other hand, our work is similar to [Cortes and Mohri, 2014] in that it assumes that the hypothesis space is a subset of a RKHS. While the scope of these two papers is to provide guarantees for feature- and kernel-based regularization algorithms, our results aim at proving the generalization bounds for domain adaptation that combines the advantages of Rademacher based bounds and completes them by explicitly introducing a natural distance which quadratic proxima can be estimated efficiently in linear time.

The rest of the chapter is organized as follows: in section 3, with present the optimal transportation problem, its dual and show how the functional derived from the latter can be embedded into Hilbert space; in section 4, we introduce some basic notations and properties related to kernel embeddings of distribution functions. Then, we show how the error function can be written in terms of the inner-product of the corresponding Hilbert space. In section 5, we present generalization bounds for the source and target error functions. Section 6 evaluates our results on a benchmark computer vision data set. Finally, we conclude with some ideas about the future research directions in section 7.

6.3 Optimal transportation

In this section, we present the formalization of the Monge-Kantorovich optimization problem. We further introduce its dual and rewrite it using a kernel formulation that allows us to embed the optimization criterion into a tractable space.

6.3.1 Monge-Kantorovich problem

Following [Courty et al., 2014], we consider two domains $\Omega_1 = \Omega_2 = \mathbb{R}^d$. Let $\mathcal{P}(\Omega_i)$ be the set of all probability measures over Ω_i. Let $p = \mathcal{P}(\Omega_1)$ and $q = \mathcal{P}(\Omega_2)$ be two probability measures. Then an application $\boldsymbol{T} : \Omega_1 \rightarrow \Omega_2$ is said to be a transport if $\boldsymbol{T}\#p = q$ where

$$\boldsymbol{T}\#p(y) = p(\boldsymbol{T}^{-1}(\boldsymbol{y})) \; \forall \boldsymbol{y} \in \Omega_2.$$

and $\#$ is the image measure(or push-forward). The cost associated to this transport is

$$C(\boldsymbol{T}) = \int_{\Omega_1} c(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))dp(\boldsymbol{x}),$$

where the cost function $c : \Omega_1 \times \Omega_2 \to \mathbb{R}^+$ can be seen as the energy required to move a mass $p(\boldsymbol{x})$ from $\boldsymbol{x}$ to $\boldsymbol{y}$. Altogether, this leads to a definition of the Monge optimal transportation problem where the optimal transport $\boldsymbol{T}_0$ is given as a solution of the following problem:

$$\boldsymbol{T}_0 = \arg\min_{\boldsymbol{T}} \int_{\Omega_1} c(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))dp(\boldsymbol{x}),\ s.t.\ \boldsymbol{T}\#p = q.$$

The graphical illustration of the Monge problem is presented in Figure 6.1[1].

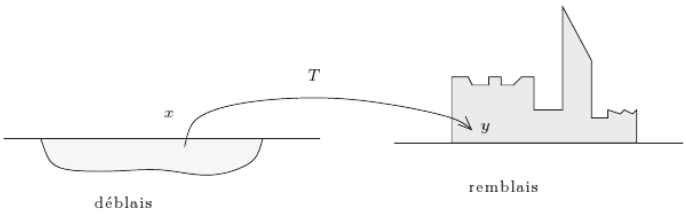


Figure 6.1: Graphical representation of the optimal transportation problem

6.3.2 Dual problem

However, nonlinearity of the objective functional and a lack of convexity for its domain make Monge's problem difficult to solve. Kantorovich in [Kantorovich, 1942] addressed these issues by giving a relaxation of the original Monge optimal transportation problem using the notion of a probabilistic coupling γ defined as a joint probability measure over $\Omega_1 \times \Omega_2$. The Monge-Kantorovich problem reads as follows:

$$\arg\min_{\gamma} \int_{\Omega_1 \times \Omega_2} c(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))d\gamma(\boldsymbol{x}, \boldsymbol{y}),\ s.t.\ \boldsymbol{P}^{\Omega_1}\#\gamma = p, \boldsymbol{P}^{\Omega_2}\#\gamma = q,$$

[1]Figure depicted from Villani [2009].

where $\boldsymbol{P}^{\Omega_i}$ is the projection over Ω_i. It admits a unique solution γ_0 and allows to define the Wasserstein distance between p and q as follows:

$$W(p,q) = \inf_{\gamma} \int_{\Omega_1 \times \Omega_2} c(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x})) d\gamma(\boldsymbol{x}, \boldsymbol{y}), \ s.t. \ \boldsymbol{P}^{\Omega_1} \# \gamma = p, \boldsymbol{P}^{\Omega_2} \# \gamma = q.$$

Using the Kantorovich duality principle, one can prove the following theorem [Dudley, 2002]:

Theorem 6.6. *Let $p, q \in \mathcal{P}^1(\Omega_i)$ where Ω_i is separable. Then the Wasserstein distance can be expressed as follows:*

$$W(p,q) = \|p - q\|_L^* = \sup_{\|f\|_L \leq 1} \left| \int f d(p-q) \right|,$$

where

$$\|f\|_L = \sup_{x \neq y \in \Omega_i} \frac{|f(x) - f(y)|}{d(x,y)}$$

is the Liptschitz semi-norm for real-valued continuous f on Ω_i and some metric $d(\cdot,\cdot)$ on $\Omega_1 \times \Omega_2$.

6.3.3 $W(p,q)$ in RKHS

From this results, one cant see that this metric is restricted to the class of function $\mathcal{F} = \{f : \|f\|_L \leq 1\}$. Following the ideas from [Gao and Galvao, 2014], one may construct a Hilbert space where this class of function is embedded in a unit ball $\mathcal{B}_{\mathcal{H}_k}$ of a Hilbert space $\mathcal{H}_k$ with an associated kernel k, i.e.:

$$\mathcal{F} = \{f : \|f\|_L \leq 1\} = \{f : \|f\|_{\mathcal{H}_k} \leq 1\} = \mathcal{B}_{\mathcal{H}_k}.$$

The following theorem defines a new distance $W_{\mathcal{H}_k}$.

Theorem 6.7. *[Gao and Galvao, 2014] If the kernel k is square-root integrable w.r.t. both p and q then*

$$W_{\mathcal{H}_k}(p,q) = \left\| \int k(\cdot, x) dp - \int k(\cdot, x) dq \right\|_{\mathcal{H}_k},$$

where $\mathcal{H}_k$ induces kernel $k \in \mathcal{B}_{\mathcal{H}_k}$.

Finally, an important result that relates $W_{\mathcal{H}_k}$ and W is given in the following Corollary.

Corollary 6.8. *If $0 \leq k(x_i, x_j) \leq K$ and $d(x, y) = \|k(\cdot, x) - k(\cdot, y)\|_{\mathcal{H}_k}$ then*

$$W_{\mathcal{H}_k} \leq W \leq \sqrt{W_{\mathcal{H}_k}^2 + 2K}.$$

This result shows that $W_{\mathcal{H}_k}$ is comparable with W in the probability metric space.

6.4 Domain adaptation model based on feature maps

In this section, we briefly introduce kernel embeddings of distribution functions and show how Definition 8 can be restated when operating in a RKHS. We start with a definition of a mean map and its empirical estimate.

Definition 10. Let $k : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ be a kernel in the RKHS $\mathcal{H}_k$ and $\phi(x) = k(x, \cdot)$. Then, the following mapping

$$\mu[P_X] = \mathbb{E}_x[\phi(x)]$$

is called a mean map. Its empirical value is given by the following estimate:

$$\mu[X] = \frac{1}{m} \sum_{i=1}^{m} \phi(x_i),$$

where we $X = \{x_1, ..., x_m\}$ is drawn i.i.d. from P_X.

If $\mathbb{E}_x[k(x, x)] < \infty$ then $\mu[P_X]$ is an element of RKHS $\mathcal{H}_k$. According to the Moore-Aronszajn theorem, the reproducing property of $\mathcal{H}_k$ allows us to rewrite every function $f \in \mathcal{H}_k$ in the following form: $\langle \mu[P_X], f \rangle = \mathbb{E}_x[f(x)]$.

We now give a definition of the maximum mean discrepancy (MMD) between two distributions.

Definition 11. Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$ and let p and q be two probability Borel measures. Then we define $d_{MMD(p,q)}$ as:

$$d_{MMD(p,q)} = \sup_{f \in \mathcal{F}} [\mathbb{E}_{x \sim p}[f(x)] - \mathbb{E}_{y \sim q}[f(y)]].$$

This expression can be further simplified by observing that in RHKS every function f can be written as $f(x) = \langle \phi(x), f \rangle_{\mathcal{H}_k}$. Finally, if $\|f\|_{\mathcal{H}_k} \leq 1$[1], we have:

$$d_{MMD(p,q)} = \|\mu_{x \sim p}[\phi(x)] - \mu_{y \sim q}[\phi(y)]\|_{\mathcal{H}_k}.$$

At this point it becomes obvious that MMD distance coincidences with $W_{\mathcal{H}_k}(p, q)$ defined in the previous section. This fact is quite important as it shows that the usage of MMD as a discrepancy measure between domains distribution functions for the generalization bounds is relevant as it appears naturally as a solution of the optimization problem directly linked to the domain adaptation problem.

A typical approach used by the majority of domain adaptation algorithms is to try to find a function that maps both tasks to a shared latent space where a classifier learned on the source data is assumed to have a good performance when applied to the target data.

We now assume that $l_{h,f} : x \to l(h(x), f(x))$ is a convex loss-function defined $\forall h, f \in \mathcal{F}$. We further assume that l obeys the triangle inequality. Similar to Definition 1, we say that $h(x)$ correspond to the hypothesis and $f(x)$ to the true labeling functions, respectively. Considering that $h, f \in \mathcal{F}$, the loss function l is a non-linear mapping of the RKHS $\mathcal{H}_k$ for the family of losses $l(h(x), f(x)) = |h(x) - f(x)|^q$[2]. Using results from [Saitoh, 1997], one may show that $l_{h,f}$ also belongs to the RKHS $\mathcal{H}_{k^q}$ admitting the reproducing kernel k^q and that its norm obeys the following inequality:

$$||l_{h,f}||^2_{\mathcal{H}_{k^q}} \leq ||h - f||^{2q}_{\mathcal{H}_k}.$$

This result gives us two important properties of $l_{f,h}$ that we use further:

[1] For the sake of convenience, we further consider only functions $f \in \mathcal{F}$ where $\mathcal{F}$ is a unit ball in the RKHS $\mathcal{H}_k$. Nevertheless, all the results presented in this paper can be easily generalized to the case where $\|f\|_{\mathcal{H}_k} \leq C$.

[2] If $h, f \in \mathcal{F}$ then $h - f \in \mathcal{F}$ implying that $l(h(x), f(x)) = |h(x) - f(x)|^q$ is a nonlinear transform for $h - f \in \mathcal{F}$.

- $l_{h,f}$ belongs to the RKHS that allows us to use the reproducing property;
- $||l_{h,f}||_{\mathcal{H}_{kq}}$ is bounded.

Thus, the error function defined above can be also expressed in terms of the inner product in the corresponding Hilbert space, i.e[1]:

$$\epsilon_S(h, f_S) = \mathbb{E}_{x\sim D_S}[l(h(x), f_S(x))] = \mathbb{E}_{x\sim D_S}[\langle \phi(x), l\rangle_{\mathcal{H}}].$$

We define the target error in the same manner:

$$\epsilon_T(h, f_T) = \mathbb{E}_{y\sim D_T}[l(h(y), f_T(y))] = \mathbb{E}_{y\sim D_T}[\langle \phi(y), l\rangle_{\mathcal{H}}].$$

When the source and target error functions are defined with respect to h and $f_{S,T}$, we use the shorthand $\epsilon_S(h, f_S) = \epsilon_S(h)$ and $\epsilon_T(h, f_T) = \epsilon_T(h)$. We also note that f and h are not restricted to be binary-valued functions.

From practical point of view, we observe that numerous domain adaptation and transfer learning approaches are based on MMD minimization [Chen et al., 2009; Geng et al., 2011; Huang et al., 2006; Pan et al., 2008, 2009]. Furthermore, conditional kernel embeddings were used for target and conditional shift correction [Zhang et al., 2013] and vector-valued regression [Grünewälder et al., 2012]. Thus, the use of this metric in domain adaptation theory is justified and appears to be natural.

6.5 Generalization bounds using MMD distance between kernel embeddings

In this section, we introduce generalization bounds for the source and target error when the divergence between tasks' distributions is measured by the MMD distance. We start with a lemma that relates the source and target error in terms of the introduced discrepancy measure for an arbitrary pair of hypothesis. Then, we show how target error can be bounded by the empirical estimate of the MMD plus the complexity term.

[1]For simplicity, we will further write $\mathcal{H}$ meaning $\mathcal{H}_{kq}$ and l meaning $l_{f,h}$.

6.5.1 A bound relating the source and target error

Using the definitions introduced before we can prove the following lemma.

Lemma 6.9. *If* $\|l\|_{\mathcal{H}} \leq 1$ *then for every* h, h' *the following holds:*

$$\epsilon_T(h, h') \leq \epsilon_S(h, h') + d_{MMD(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})}.$$

Proof.

$$\begin{aligned}
\epsilon_T(h, h') &= \epsilon_T(h, h') + \epsilon_S(h, h') - \epsilon_S(h, h') \\
&= \epsilon_S(h, h') + \mathbb{E}_{y \sim D_T}[\langle \phi(y), l \rangle_{\mathcal{H}}] - \mathbb{E}_{x \sim D_S}[\langle \phi(x), l \rangle_{\mathcal{H}}] \\
&= \epsilon_S(h, h') + \langle \mathbb{E}_{y \sim D_T}[\phi(y)] - \mathbb{E}_{x \sim D_S}[\phi(x)], l \rangle_{\mathcal{H}} \\
&\leq \epsilon_S(h, h') + d_{MMD(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})}.
\end{aligned}$$

□

In this proof, the second line is obtained using the definition of the source and target errors while the last inequality is due to the fact that $\|l\|_{\mathcal{H}} \leq 1$. This lemma is similar to Lemma 3 from [Ben-David et al., 2010a]. Using it and the result that relates the true and the empirical MMD distance [Song, 2008], we can prove the following theorem.

Theorem 6.10. *Let* $\|l\| \leq 1$ *in a RKHS* $\mathcal{H}$, $\mathcal{U}_{\mathcal{S}}$ *and* $\mathcal{U}_{\mathcal{T}}$ *are two samples of size* m *drawn i.i.d. from* $\mathcal{D}_{\mathcal{S}}$ *and* $\mathcal{D}_{\mathcal{T}}$ *respectively then with probability at least* $1 - \delta$ *for all* h *the following holds:*

$$\epsilon_T(h) \leq \epsilon_S(h) + \hat{d}_{MMD(\mathcal{U}_{\mathcal{S}}, \mathcal{U}_{\mathcal{T}})} + \frac{2}{m}\mathbb{E}_{\mathcal{D}_{\mathcal{S}}}\left[\sqrt{tr(K_{\mathcal{S}})}\right] + \frac{2}{m}\mathbb{E}_{\mathcal{D}_{\mathcal{T}}}\left[\sqrt{tr(K_{\mathcal{T}})}\right] + 2\sqrt{\frac{\log(\frac{2}{\delta})}{2m}} + \lambda,$$

where $\hat{d}_{MMD(\mathcal{U}_{\mathcal{S}}, \mathcal{U}_{\mathcal{T}})}$ *is an empirical counterpart of* $d_{MMD(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})}$, K_S *and* K_T *are the kernel matrices calculated on* $\mathcal{U}_{\mathcal{S}}$ *and* $\mathcal{U}_{\mathcal{T}}$ *respectively and* λ *is the combined error of the ideal hypothesis* h^* *that minimizes the combined error of* $\epsilon_S(h) + \epsilon_T(h)$.

Proof.

$$
\begin{aligned}
\epsilon_T(h) &\leq \epsilon_T(h^*) + \epsilon_T(h^*, h) \\
&\leq \epsilon_T(h^*) + \epsilon_S(h, h^*) + \epsilon_T(h^*, h) - \epsilon_S(h, h^*) \\
&\leq \epsilon_T(h^*) + \epsilon_S(h, h^*) + d_{MMD(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T})} \\
&\leq \epsilon_T(h^*) + \epsilon_S(h) + \epsilon_S(h^*) + d_{MMD(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T})} \\
&= \epsilon_S(h) + d_{MMD(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T})} + \lambda \\
&\leq \epsilon_S(h) + \hat{d}_{MMD(\mathcal{U}_\mathcal{S}, \mathcal{U}_\mathcal{T})} + \frac{2}{m}\mathbb{E}_{\mathcal{D}_\mathcal{S}}\left[\sqrt{tr(K_\mathcal{S})}\right] + \frac{2}{m}\mathbb{E}_{\mathcal{D}_\mathcal{T}}\left[\sqrt{tr(K_\mathcal{T})}\right] + 2\sqrt{\frac{\log(\frac{2}{\delta})}{2m}} + \lambda.
\end{aligned}
$$

□

We can see that this theorem is similar to Theorem 6.2. The main difference, however, is that the complexity term does not depend on the Vapnik-Chervonenkis dimension. In our case, the loss function between two errors is bounded by the empirical MMD between distributions and two terms that correspond to the empirical Rademacher complexities of $\mathcal{H}$ w.r.t. the source and target samples. In both Theorem 3 and 11, λ plays the role of the combined error of the ideal hypothesis. Its presence in the bound comes from the use of the triangle inequality for classification error.

This result is particularly useful as $\hat{d}_{MMD(\mathcal{U}_\mathcal{S}, \mathcal{U}_\mathcal{T})}$ can be approximated by $\hat{d}^2_{MMD(\mathcal{U}_\mathcal{S}, \mathcal{U}_\mathcal{T})}$ that, in its turn, can be calculated using the following equation:

$$
\hat{d}^2_{MMD(\mathcal{U}_\mathcal{S}, \mathcal{U}_\mathcal{T})} = \frac{1}{m(m-1)} \sum_{i \neq j} h((x_i, x_j), (y_i, y_j))
$$

where for $x \in \mathcal{U}_\mathcal{S}$ and $y \in \mathcal{U}_\mathcal{T}$ we have

$$
h((x_i, x_j), (y_i, y_j)) = \tfrac{1}{m^2} \textstyle\sum_{i,j=1}^{m} k(x_i, x_j) - \tfrac{2}{m^2} \sum_{i,j=1}^{m} k(x_i, y_j) + \tfrac{1}{m^2} \sum_{i,j=1}^{2} k(y_i, y_j).
$$

The following lemma gives a computation guarantee for the unbiased estimator of $\hat{d}^2_{MMD(\mathcal{U}_\mathcal{S}, \mathcal{U}_\mathcal{T})}$.

Lemma 6.11. *Gretton et al. [2012]*

For $m_2 = m/2$ the estimator

$$\hat{d}^2_{MMD(\mathcal{U}_\mathcal{S},\mathcal{U}_\mathcal{T})} = \frac{1}{m_2}\sum_{i=1}^{m_2} h((x_{2i-1}, y_{2i-1})(x_{2i}, y_{2i}))$$

can be computed in linear time and it is an unbiased estimator of $d^2_{MMD(\mathcal{D}_\mathcal{S},\mathcal{D}_\mathcal{T})}$.

We also note that the obtained bound can be further simplified if one uses, for instance, Gaussian, exponential or Laplacian kernels to calculate matrices $K_\mathcal{S}$ and $K_\mathcal{T}$. In this case $tr(K_\mathcal{S}) = tr(K_\mathcal{T}) = m$.

Finally, it can be seen that the bound from Theorem 6.10 has the same terms as Theorem 6.2 while the MMD distance is estimated as in Theorem 6.5.

6.5.2 A learning bound for combined error

In domain adaptation, one often wants to find a trade-off between minimizing the source and target errors depending on the number of instances available in each domain and their mutual correlation. Let us assume that we possess βn labeled instances drawn independently from $\mathcal{D}_\mathcal{T}$ and $(1-\beta)n$ labeled instances drawn independently from $\mathcal{D}_\mathcal{S}$. In this case, the empirical combined error is defined as a convex combination of errors on source and target training data

$$\hat{\epsilon}_\alpha(h) = \alpha\hat{\epsilon}_T(h) + (1-\alpha)\hat{\epsilon}_S(h),$$

where $\alpha \in [0, 1]$.

The use of the combined error is motivated by the fact that if the number of instances in target sample is small compared to the number of instances in source domain (which is usually the case in domain adaptation), minimizing target error may not be appropriate. Instead, one may want to find an appropriate value of α that ensures the minimum of $\hat{\epsilon}_\alpha(h)$ with respect to a given hypothesis h.

We now follow [Ben-David et al., 2010a] and prove a lemma that bounds the difference between target error $\hat{\epsilon}_T$ and weighted error $\hat{\epsilon}_\alpha$. Next, we use the concentration results for MMD estimators to bound the difference between empirical and true combined error functions.

Lemma 6.12. *With the assumptions from Lemma 1 the following holds:*

$$|\epsilon_\alpha(h) - \epsilon_T(h)| \leq (1-\alpha)(d_{MMD}(\mathcal{D}_T, \mathcal{D}_S) + \lambda).$$

Proof.

$$\begin{aligned} |\epsilon_\alpha(h) - \epsilon_T(h)| &= |\alpha\epsilon_T(h) + (1-\alpha)\epsilon_S(h) - \epsilon_T(h)| \\ &= |(1-\alpha)(\epsilon_S(h) - \epsilon_T(h))| \\ &\leq (1-\alpha)(d_{MMD}(\mathcal{D}_T, \mathcal{D}_S) + \lambda). \end{aligned}$$

□

Here, the second line reads from the definition of the combined error while the final result is obtained by adding and subtracting $\epsilon_S(h, h*)$ and $\epsilon_T(h, h*)$ and applying the triangle inequality the the resulting terms. This result shows that the level of confidence, that we have in using the source data, defines the potential difference between the weighted combined error and the target error. This conclusion agrees with the idea that when $\alpha \to 1$ the distance between distributions does not define the potential success of the domain adaptation as we can not rely on source data to improve performance in the target domain.

We now proceed to the concentration inequality for the true and empirical combined error. The following lemma is a slight modification of Theorem 14 from [Gretton et al., 2012].

Lemma 6.13. *Let $\|l\| \leq 1$ in a RKHS $\mathcal{H}$. Let D be a sample of size n corresponding to the combined error where βn points are drawn from $\mathcal{D}_\mathcal{T}$ and $(1-\beta)n$ from $\mathcal{D}_\mathcal{S}$. Then with probability at least $1-\delta$ for all h with $0 \leq k(x_i, x_j) \leq K$ the following holds:*

$$\begin{aligned} P\left\{|\hat{\epsilon}_\alpha(h) - \epsilon_\alpha(h)| > 2\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right) + \epsilon\right\} \\ \leq \exp\left\{\frac{-\epsilon^2 n}{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)}\right\}. \end{aligned}$$

Proof. First, we use McDiarmid's theorem in order to obtain the right side of the inequality by defining the maximum changes of magnitude when one of the sample vectors has been changed. We first rewrite the difference between the empirical and true combined error in the following way

$$\begin{aligned}
|\hat{\epsilon}_\alpha(h) - \epsilon_\alpha(h)| &= |\alpha\epsilon_T(h) - (\alpha-1)\epsilon_S(h) - \alpha\hat{\epsilon}_T(h) + (\alpha-1)\hat{\epsilon}_S(h)| \\
&= |\alpha\mathbb{E}_{\mathcal{D}_\mathcal{T}}(l) - (\alpha-1)\mathbb{E}_{\mathcal{D}_\mathcal{T}}(l) - \frac{\alpha}{n\beta}\sum_{i=1}^{\beta n} l(h(x_i), f_S(x_i)) \\
&+ \frac{(\alpha-1)}{n(1-\beta)}\sum_{i=1}^{n(1-\beta)} l(h(y_i), f_T(y_i))| \\
&\leq \sup_{l\in\mathcal{F}} |\alpha\mathbb{E}_{\mathcal{D}_\mathcal{T}}(l) - (\alpha-1)\mathbb{E}_{\mathcal{D}_\mathcal{S}}(l) - \frac{\alpha}{n\beta}\sum_{i=1}^{\beta n} l(h(x_i), f_S(x_i)) \\
&+ \frac{(\alpha-1)}{n(1-\beta)}\sum_{i=1}^{n(1-\beta)} l(h(y_i), f_T(y_i))|.
\end{aligned}$$

Changing either x_i or y_i in this expression changes its value by at most $\frac{2\alpha\sqrt{K}}{\beta n}$ and $\frac{2(1-\alpha)\sqrt{K}}{(1-\beta)n}$, respectively. This gives us the denominator of the exponent

$$\beta n\left(\frac{2\alpha\sqrt{K}}{\beta n}\right)^2 + (1-\beta)n\left(\frac{2(1-\alpha)\sqrt{K}}{(1-\beta)n}\right)^2 = \frac{4K}{n}\left(\frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{(1-\beta)}\right).$$

Then, we bound the expectation of the difference between the true and empirical combined errors by the sum of Rademacher averages over the samples. Denoting by X' an i.i.d sample of size βn drawn independently of X (and likewise for Y'), and using the symmetrization technique we have

$$\begin{aligned}
&\mathbb{E}_{X,Y}\sup_{l\in\mathcal{H}} |\alpha\mathbb{E}_{\mathcal{D}_\mathcal{T}}(l) - (\alpha-1)\mathbb{E}_{\mathcal{D}_\mathcal{S}}(l) - \frac{\alpha}{n\beta}\sum_{i=1}^{\beta n} l(h(x_i), f_S(x_i)) + \frac{(\alpha-1)}{n(1-\beta)}\sum_{i=1}^{n(1-\beta)} l(h(y_i), f_T(y_i))| \\
&\leq \mathbb{E}_{X,Y}\sup_{l\in\mathcal{H}} |\mathbb{E}_{X'}\left(\frac{\alpha}{n\beta}\sum_{i=1}^{\beta n} l(h(x_i'), f_S(x_i'))\right) - (\alpha-1)\mathbb{E}_{Y'}\left(\frac{(\alpha-1)}{n(1-\beta)}\sum_{i=1}^{\beta n} l(h(y_i'), f_T(y_i'))\right)
\end{aligned}$$

$$
\begin{aligned}
&- \frac{\alpha}{n\beta}\sum_{i=1}^{\beta n} l(h(x_i), f_S(x_i)) + \frac{(\alpha-1)}{n(1-\beta)}\sum_{i=1}^{(1-\beta)n} l(h(y_i), f_T(y_i))| \\
&\leq \mathbb{E}_{X,X',Y,Y'} \sup_{l\in\mathcal{H}} |\frac{\alpha}{n\beta}\sum_{i=1}^{\beta n}\sigma_i(l(h(x_i'), f_S(x_i')) - l(h(x_i), f_S(x_i))) \\
&+ \frac{1-\alpha}{n(1-\beta)}\sum_{i=1}^{\beta n}\sigma_i(l(h(y_i'), f_T(y_i')) - l(h(y_i), f_T(y_i)))| \\
&\leq 2\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right).
\end{aligned}
$$

Finally, the Rademacher averages, in their turn, are bounded using a theorem from [Bartlett and Mendelson, 2003]. This gives us the desired result

$$
\begin{aligned}
P\left\{|\hat{\epsilon}_\alpha(h) - \epsilon_\alpha(h)| > 2\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right) + \epsilon\right\} \\
\leq \exp\left\{\frac{-\epsilon^2 n}{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)}\right\}.
\end{aligned}
$$

□

We are now ready to prove the analogue of Theorem 6.3.

Theorem 6.14. *Let $\|l\| \leq 1$ in a RKHS $\mathcal{H}$ and let $\mathcal{U}_\mathcal{S}$ and $\mathcal{U}_\mathcal{T}$ be two samples of size m drawn i.i.d. from $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ respectively. Let D be a sample of size n corresponding to the combined error where βn points are drawn from $\mathcal{D}_\mathcal{T}$ and $(1-\beta)n$ from $\mathcal{D}_\mathcal{S}$. If $\hat{h}$ is the empirical minimizer of $\hat{\epsilon}_\alpha(h)$ and $h^* = \min_h \epsilon_T(h)$ then for any $\delta \in (0,1)$ with probability at least $1-\delta$ (over the choice of samples),*

$$
\epsilon_T(\hat{h}) \leq \epsilon_T(h_T^*) + c_1 + 2(1-\alpha)c_2,
$$

where

$$
c_1 = 2\sqrt{\frac{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\log(2/\delta)}{n}} + 2\left(\sqrt{\frac{\alpha}{\beta}} + \sqrt{\frac{1-\alpha}{1-\beta}}\right)\sqrt{K/n},
$$

$$c_2 = \hat{d}_{MMD(\mathcal{U}_{\mathcal{S}},\mathcal{U}_{\mathcal{T}})} + \frac{2}{m}\mathbb{E}_{\mathcal{D}_{\mathcal{S}}}\left[\sqrt{tr(K_{\mathcal{D}_{\mathcal{S}}})}\right] + \frac{2}{m}\mathbb{E}_{\mathcal{D}_{\mathcal{T}}}\left[\sqrt{tr(K_{\mathcal{D}_{\mathcal{T}}})}\right] + 2\sqrt{\frac{\log(\frac{2}{\delta})}{2m}} + \lambda.$$

Proof.

$$\begin{aligned}
\epsilon_T(\hat{h}) &\leq \epsilon_\alpha(\hat{h}) + (1-\alpha)(d_{MMD}(\mathcal{D}_T, \mathcal{D}_S) + \lambda) \\
&\leq \hat{\epsilon}_\alpha(\hat{h}) + \sqrt{\frac{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\log(2/\delta)}{n}} \\
&\quad + 2\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right) + (1-\alpha)(d_{MMD}(\mathcal{D}_T, \mathcal{D}_S) + \lambda) \\
&\leq \hat{\epsilon}_\alpha(h_T^*) + \sqrt{\frac{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\log(2/\delta)}{m}} \\
&\quad + 2\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right) + (1-\alpha)(d_{MMD}(\mathcal{D}_T, \mathcal{D}_S) + \lambda) \\
&\leq \epsilon_\alpha(h_T^*) + 2\sqrt{\frac{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\log(2/\delta)}{n}} \\
&\quad + 4\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right) + (1-\alpha)(d_{MMD}(\mathcal{D}_T, \mathcal{D}_S) + \lambda) \\
&\leq \epsilon_T(h_T^*) + 2\sqrt{\frac{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\log(2/\delta)}{n}} \\
&\quad + 4\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right) + 2(1-\alpha)(d_{MMD}(\mathcal{D}_T, \mathcal{D}_S) + \lambda) \\
&\leq \epsilon_T(h_T^*) + 2\sqrt{\frac{2K\left(\frac{(1-\alpha)^2}{1-\beta} + \frac{\alpha^2}{\beta}\right)\log(2/\delta)}{n}} + 4\sqrt{K/n}\left(\frac{\alpha}{n\beta\sqrt{\beta}} + \frac{(1-\alpha)}{n(1-\beta)\sqrt{1-\beta}}\right) \\
&\quad + 2(1-\alpha)(\hat{d}_{MMD}(\mathcal{U}_T, \mathcal{U}_S) + \frac{2}{m}\mathbb{E}_{\mathcal{D}_{\mathcal{S}}}\left[\sqrt{tr(K_{\mathcal{S}})}\right] + \frac{2}{m}\mathbb{E}_{\mathcal{D}_{\mathcal{T}}}\left[\sqrt{tr(K_{\mathcal{T}})}\right] + 2\sqrt{\frac{\log(\frac{2}{\delta})}{2m}} + \lambda).
\end{aligned}$$

□

The proof follows the standard theory of uniform convergence for empirical risk minimizers where lines 1 and 5 are obtained using Lemma 12, lines 2 and 4 are ob-

tained using Lemma 13, line 3 follows from the definition of $\hat{h}$ and h_T^* and line 6 is a consequence of Theorem 6.10.

Several observations can be made from this theorem. First of all, the main quantities that define the potential success of domain adaptation according to [Ben-David et al., 2010a] (i.e., the distance between the distributions and the combined error of the joint ideal hypothesis) are preserved in the bound. This is an important point that indicates that two results are not contradictory or supplementary. Second, rewriting the approximation of the bound as a function of α and omitting additive constants can lead to a similar result as in Theorem 6.3. This observation may point out the existence of a strong connection between them.

We examine below the relationships between the results from Section 2 and the ones presented above.

6.5.3 Analysis of the bounds

As it was stated above, the main theoretical difference between the two types of bounds is that VC-dimension appears in the original work and does not appear in ours. In order to understand the difference between the proposed and original bounds, we need to outdraw the classical scheme that is used to prove the concentration inequalities in the learning theory. It usually consists of the following steps:

1. Using McDiarmid's or Hoeffding inequality to obtain the concentration bound of the form:

$$P^m\{|\epsilon(h) - \hat{\epsilon}(h)| > t\} \leq c_1 \exp\{\frac{-c_2 m t^2}{c_3}\},$$

where c_1, c_2, c_3 are some constants.

2. Use sample symmetrization in order to bound the desired probability by the probability of an event based on two samples:

$$P^m(Q) \leq 2P^{2m}(R) \leq 4\Pi_{\mathcal{H}}(2m) \exp\{\frac{-c_2 m t^2}{c_3}\},$$

where $Q = \{z \in \mathcal{Z}^m : |\epsilon(h) - \hat{\epsilon}(h)| > t\}$, $R = \{(r, s) \in \mathcal{Z}^m \times \mathcal{Z}^m : |\epsilon(h) - \hat{\epsilon}(h)| > t/2\}$ and $\Pi_{\mathcal{H}}(2m)$ is a growth function.

3. Bound the growth function using Sauer's lemma for a hypothesis space $\mathcal{H}$ of

VC-dimension d:

$$\Pi_{\mathcal{H}}(n) \leq n^d.$$

Proof of Theorem 6.2 follows this scheme when the bound for the empirical estimator of the $\mathcal{H}\Delta\mathcal{H}$ divergence is obtained. Authors used the fact that the VC-dimension of $\mathcal{H}\Delta\mathcal{H}$ is at most twice the VC-dimension of $\mathcal{H}$ and this, in combination with Sauer's lemma, gives the desired result. Theorem 6.3, in its turn, uses the same kind of reasoning to derive a bound for the difference between the true and empirical combined error (Lemma 5, [Ben-David et al., 2010a]).

Theorem 6.10, however, was proved in a different manner. To obtain the bound for the empirical MMD distance it uses the following estimate: for a RKHS $\mathcal{H}$ where $\|l\|_{\mathcal{H}} \leq 1$ for every $l \in \mathcal{H}$, we have

$$\mathcal{R}_m(\mathcal{H}) \leq \frac{2}{m}\mathbb{E}_X\sqrt{k(x_i, x_j)}.$$

where $\mathcal{R}_m(\mathcal{H}) = \mathbb{E}_X\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{H}}|\frac{2}{m}\sum_{i=1}^m \sigma_i f(x_i)|\right]$ is the Rademacher average of space $\mathcal{H}$. This estimate introduces two terms to the bound of Theorem 6.10 that correspond to the Rademacher average of the space $\mathcal{H}$ with respect to the distributions $\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}$. Lemma 13 also uses this estimate to derive the bound for the true and empirical combined error.

The strong connection between both results can be shown using the following observation: if we replace $\mathcal{H}$ with an arbitrary space $\mathcal{F}$ of binary-valued functions (e.g., 0-1 loss functions) with finite VC-dimension d_{VC}, it can be shown that $\mathcal{R}_m(\mathcal{F}) \leq c\sqrt{\frac{d_{VC}}{m}}$ for some constant c. In this case, the bounds recover the VC bounds of [Ben-David et al., 2010a].

6.6 Experimental results

In this section, we evaluate the proposed bounds and compare them to the existing results using Office/Caltech data set described in Chapter 3.

We analyze the results obtained in this chapter in two different settings. First, we evaluate a computational efficiency of the empirical estimate of $\mathcal{H}\Delta\mathcal{H}$ divergence[1] and

[1]We recall that discrepancy distance from [Mansour et al., 2009] coincides with $\mathcal{H}\Delta\mathcal{H}$ divergence

compare it to the computational cost of MMD distance estimation. Then, we study the behavior of both divergence measures when calculated during the optimization procedure of one of the state-of-art domain adaptation approaches presented in [Shi and Sha, 2012]. This particular choice of method can be explained by two reasons: (1) we do not want to fall into a favorable setting by choosing an approach that uses kernels to find a suitable projection of data; (2) it follows an iterative procedure so that we can trace the evolution of both divergence measures and compare them to the real classification error values.

6.6.1 Run-time performance comparison

$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$ is usually calculated by learning a linear classifier to discriminate between source and target samples pseudo-labeled with 0 and 1. To this end, we apply Naive Bayes (NB), Linear Discriminant Analysis (LDA) and linear Support Vector Machines (SVM) to learn a hypothesis that distinguishes between source and target domains. MMD, in its turn, is calculated using an off-shelf routine provided in a toolbox ITE [Szabó, 2014]. We vary the number of instances from Amazon and Caltech domains by gradually increasing it from 50 to 950 with step equal to 50. The resulting plot for four estimators is presented in Figure 6.2. We can see that NB-based estimation of $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ is faster than LDA- and SVM-based estimations but still can not match the performance of $\hat{d}_{MMD}$ estimation. The results of computational cost obtained on 12 domain adaptation problems from Office/Caltech data set are presented in Table 6.1. We can see from the results that NB-, LDA- and SVM-based estimation of $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ is much slower when compared to $\hat{d}_{MMD}$ estimation calculated according to Lemma 4.11.

6.6.2 Divergence analysis

The Information-Theoretical Learning of Discriminative Clusters (ITLDC) [Shi and Sha, 2012] is a state-of-art domain adaptation method that proved to be efficient on Office/Caltech data set. The main idea of this approach is to learn iteratively a linear transformation matrix L that identifies an embedding where both source and target do-

in case of 0-1 loss.

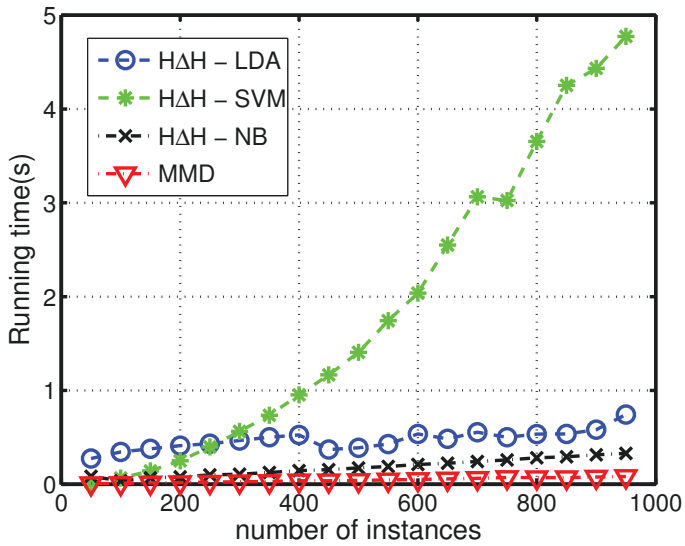


Figure 6.2: Running time as a function of samples' size on A vs. C.

Table 6.1: Run-time for $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ and $\hat{d}_{MMD}$ estimation on Office/Caltech data set.

Domain pair	$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(SVM)$	$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(LDA)$	$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(NB)$	$\hat{d}_{MMD}$
C $\rightarrow$ A	5.517	0.7651	0.3709	0.0664
C $\rightarrow$ W	1.8156	0.4924	0.2508	0.0255
C $\rightarrow$ D	1.4703	0.4745	0.2466	0.0177
A $\rightarrow$ C	5.6698	0.6803	0.3728	0.0562
A $\rightarrow$ W	1.2914	0.4855	0.2241	0.0292
A $\rightarrow$ D	0.8896	0.4747	0.2001	0.0185
W $\rightarrow$ C	1.8157	0.4944	0.2523	0.0286
W $\rightarrow$ A	1.2314	0.4588	0.2262	0.0269
W $\rightarrow$ D	0.2436	0.4386	0.0881	0.0127
D $\rightarrow$ C	1.5029	0.4752	0.2481	0.0205
D $\rightarrow$ A	0.9539	0.4643	0.1992	0.0173
D $\rightarrow$ W	0.2459	0.4365	0.0888	0.0125
Average time	1.89	0.51	0.23	**0.028**

mains are similarly distributed. Simultaneously, it optimizes the information-theoretic metric that assumed to mimic the behavior of the missclafication error in the target domain. Finally, when the resulting linear mapping is obtained, we project both source and target data to a shared space using L. We apply this approach to Amazon $\leftrightarrow$ Caltech pair of tasks in both directions. Intuitively, we expect that the classification performance in target domain may increase if the divergence between domains is properly minimized. Otherwise, according to Theorem 6.2 and Theorem 6.10, the classification accuracy in target domain will most likely decrease. In order to verify this, at each iteration of ITLDC[1] we learn a 1-NN classifier using projected source data and apply it to a transformed target data. In Figures 6.3 and 6.4, we present $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$, $\hat{d}_{MMD}$ and the classification error values obtained using 1-NN classifier for for A $\rightarrow$ C (top row) and C $\rightarrow$ A (bottom row) adaptation problems. The results for the rest of Office/Caltech data set's pairs can be found in Appendix B.

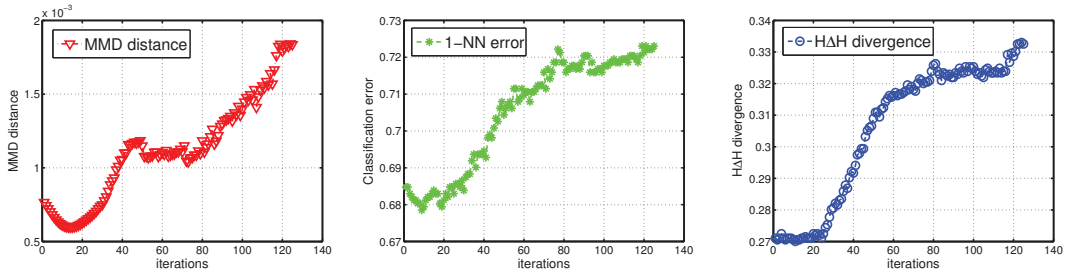


Figure 6.3: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Amazon/Caltech pair of tasks using ITLDC

Several observations follow from these figures. For A $\rightarrow$ C adaptation problem, the classification error decreases in the beginning and then slowly increases until the convergence of the optimization procedure. At this point, we may expect the divergence between two domains to have the same behavior. $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ fails to capture the initial increase as it remains nearly constant in the beginning and then follows correctly the shape of the classification error plot. On the other hand, the estimate of $\hat{d}_{MMD}$ follows the behavior of the classification accuracy almost implicitly. It decreases when the error increases and vice versa. Second example, when we swap the source and target domains, shows that both $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$, $\hat{d}_{MMD}$ capture well the general trend of the

[1]The obtained accuracy values may not coincide with the results from the original paper.

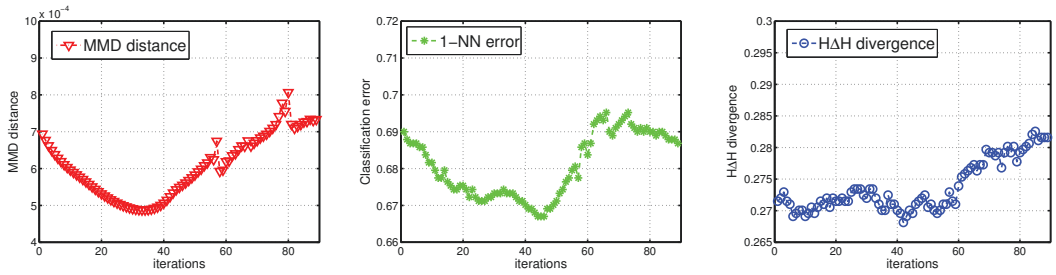


Figure 6.4: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Caltech/Amazon pair of tasks using ITLDC

classification accuracy. Both of them respond well to the changes in the behavior of the classification error. It confirms that the proposed generalization bounds are similar in shape to the true errors.

6.7 Conclusions and future work

The theory of learning from different domains, presented lately in [Ben-David et al., 2010a], studies exhaustively the domain adaptation problem using the results from statistical learning theory. The generalization bounds proved in this work show explicitly under which conditions a given domain adaptation approach is most likely to succeed.

In this chapter, we presented the generalization bounds for domain adaptation based on the notion of kernel embeddings in a Reproducing Kernel Hilbert Space. The idea underlying the proposed framework was particularly motivated by the fact that the minimization of the distance between kernel embeddings of distribution functions has already proved to be efficient and was exploited by numerous domain adaptation algorithms. Even though the obtained generalization bounds are close to the original ones, the latter does not assume that the hypothesis space is of VC-dimension. We showed that the complexity terms arising in generalization inequalities can be bounded in terms of the expectation over the trace of kernel matrix. We also presented the conditions under which our bounds may recover the standard VC-theory results. From practical point of view, the proposed formal estimator of $\mathcal{H}\Delta\mathcal{H}$ is computationally intractable problem while MMD enjoys the existence of a linear time unbiased estimator for its quadratic counterpart. The empirical evaluations support the validity of the presented

results and show their efficiency.

In future, our work can be continued in multiple directions. The most important of them is to investigate a potential dependency between α and the divergence between distributions by the means of a statistical test that would allow to define analytically the weights of source and target error in the combined objective function. This can be possibly done using the techniques from two-sample tests that minimize the probability of the empirical distance between two samples falling below a given threshold. Indeed, in domain adaptation problem, one essentially tries to minimize the Type II error that is the probability of wrongly accepting that two distributions are equal when they are not. Furthermore, the application of kernel two-sample tests may give rise to a new family of domain adaptation algorithms that find a projection of source and target tasks' data to a RKHS with the associated kernel that minimizes the Type II error.

A somewhat different interesting future direction for this contribution may be to make use of the proposed bounds in online learning setting where the relevance of arriving samples to a given target data set should be estimated as quick as possible. The same also holds for active learning setting. One may consider a situation where the number of labeled samples in source domain is drastically small compared to the number of data points in target domain. In such scenario, it can be more appropriate to choose only those data points from traget task that can be labeled reliably using a classifier trained on labeled source data sample. In both of the above mentioned cases, the computational efficiency of MMD distance can play a crucial role when compared to $\mathcal{H}\Delta\mathcal{H}$ estimation.

Finally, the proposed results can be naturally extended to the multi-source framework.

Chapter 7

Conclusions and future perspectives

In this thesis we were working on two different subjects that are: Non-negative matrix factorization and Transfer learning. The goal of the thesis is to adapt NMF to use it for unsupervised transfer learning.

7.1 Conclusions

Regarding NMF, we proposed an approach that studies the behavior of Multilayer NMF using Hoyer's projection operator. The main idea introduced in this contribution is that the hierarchical factors arising from Multilayer NMF can be obtained using the projections of the initial set of basis vectors with an appropriate level of sparsity. While Multilayer NMF presents an efficient way of obtaining these factors, its computational cost remains quite high due to the multiple factorizations performed for each layer. We showed that it can be reduced significantly if one replaces each new factor with a sparse projection of the basis vectors of the previous layer. Furthermore, we presented the theoretical analysis that explains the reasons why the proposed procedure can be more suitable for preserving low distortion of the obtained basis vectors with respect to the initial data. Finally, experimental results on several image data sets confirmed the suggestions that were studied.

In the second part, we introduced two new approaches for unsupervised transfer learning and a new view of the domain adaptation theory based on Hilbert-Schmidt embeddings.

The domain adaptation theory presented in this thesis, closes the gap between the

previous works on this subject by simultaneously achieving three goals: (1) it extends the original work to a broader class of convex loss functions; (2) it eliminates VC assumptions imposed on the hypothesis class and thus results in more data-dependent bounds; (3) it explicitly introduces the MMD distance that enjoys the existence of an efficient estimator. The experimental results presented in this contribution showed that the proposed distance follows the behavior of the true classification error while it can be computed almost ten times quicker than the best estimator for the $\mathcal{H}\Delta\mathcal{H}$ divergence. Even though there were prior works that assumed the hypothesis space to be a subset of a RKHS, their goal was to derive generalization bounds for kernel-based regularization algorithms while our contributions cover the original generalization bounds for both simple and combined objective functions.

We also proposed a nonlinear approach for unsupervised transfer learning based on kernel target alignment optimization. We used kernel alignment optimization in order to minimize the distance between the distributions of source and target tasks. We applied K-NMF to the intermediate kernels obtained during this procedure and looked for a weight matrix that reconstructs well the similarity-based representation of data. Once this matrix is found, we used it in C-NMF on the target task to obtain the final partition. Our approach was evaluated on benchmark computer vision data sets and demonstrated a significant improvement when compared to some state-of-the-art unsupervised transfer learning methods. We also showed how KTA maximization can be related to HSIC and QMI optimization. The established relationships allowed us to conclude that the use of KTA for transfer learning is justified from both theoretical and practical points of view.

Finally, we presented a linear approach that is based on the assumption that preserving the non-negativity of the embedding for both source and target tasks can be beneficial if the initial data are intrinsically non-negative. The presented approach NNE consists of two stages: (1) first, NNE discovers non-negative principal components that are further used to choose the basis vectors that are well-aligned between the source and target tasks; (2) then, it finds a non-negative embedding that is used further to obtain a partition of the target task through a simple and intuitive optimization procedure. From the theoretical point of view, we showed that the similarity measure calculated based on the obtained embedding is consistent and converges uniformly to the true similarity function. We evaluated our approach on a famous Office/Caltech

data and reported a comparable performance to the state-of-the-art domain adaptation methods.

7.2 Future perspectives

Possible future perspectives of this thesis are many. The analysis of Multilayer NMF with Hoyer's operator can be improved by designing a new paradigm that relates sparsity to discriminative power of features and thus allows to provide an analytic solution for the level of sparsity of each layer. With the current success of deep architectures, another interesting direction is to apply the proposed analysis to Deep Neural Networks. This can possibly become an interesting alternative to dropout techniques.

The theoretical analysis presented in chapter 4 can be improved by the means of a statistical test that would allow to define analytically the weights of source and target error in the combined objective function. This can be possibly done using the techniques from two-sample tests that minimize the probability of the empirical distance between two samples falling below a given threshold. Indeed, in domain adaptation problem, one essentially tries to minimize the Type II error that is the probability of wrongly accepting that two distributions are equal when they are not. Furthermore, the application of kernel two-sample tests may give rise to a new family of domain adaptation algorithms that find a projection of source and target tasks' data to a RKHS with the associated kernel that minimizes the Type II error. A somewhat different interesting future direction for this contribution is to make use of the proposed bounds in online learning setting where the relevance of arriving samples to a given target data set should be estimated as quickly as possible.

BC-NMF can be extended to work in the multi-task setting using an optimal shared Gram matrix obtained for a set of kernels corresponding to different source domains. As mentioned in Chapter 4, it can be done using the simultaneous NMF model. On the other hand, one may also try to incorporate the consensus NMF[Li et al., 2007] to the results obtained using the optimal Gram matrices of each source task. Another important future direction for this approach is to propose a method that adjusts kernel parameters in combination with weights adaptation. If found, the proposed algorithm would be able to reduce significantly the computational cots of kernel alignment opti-

mization and the number of kernels used.

The proposed NNE approach can be extended to a multi-task version using the weighted sum of terms corresponding to different tasks. In this case, we essentially encounter the problem mentioned above - how to define the weights in the combined objective function so that they minimize the overall classification error. This question, however, remains unanswered as the theory of self-taught clustering has not been developed so far due to the fact that one can not use typical approaches (e.g., VC theory bounds and Rademacher complexity based bounds) relying on the presence of labeled data. Nevertheless, unsupervised learning in general and unsupervised transfer learning in particular become a topic of ongoing interest nowadays as supervised and semi-supervised learning settings are widely covered in terms of theoretical guarantees and arguably do not have much room for improvement in terms of classification performance.

Appendix A

1 Introduction

In this Appendix we study the Multilayer NMF - a model that can be seen as a pre-training step of Deep NMF model for learning hidden representations. We analyze the factors obtained using Multilayer NMF and show that the process of building layers can be seen as a repeated application of the Hoyer's projection operator applied sequentially to the factor of the second layer. We also provide the sparsity analysis for matrices obtained during the optimization procedure at each layer. We conclude that the overall sparsity decreases with the increasing number of layers despite the general assumption that Multilayer NMF is efficient due to the fact that it increases the sparsity of learned factors.

1.1 Background and related works

There are two well-known issues that arise when one uses NMF:

- the decomposition obtained using simple NMF model is not unique and thus it is numerically unstable;
- the simple NMF model with two factors is not deep enough to produce hierarchical features.

First problem can usually be solved in two ways: either by applying some kind of initialization that transforms the initial data and further use it as an input of the NMF model (for instance, see [Gillis, 2012]) or by adding a third factor in the NMF model (so called Tri-NMF [Ding et al., 2010a]) that disables the possibility of obtaining an infinite number of decompositions for a given matrix by multiplying each of the resulting factors by an arbitrary matrix on the right and its inverse on the left. The general study about uniqueness of NMF can be found in [Donoho and Stodden, 2004]. Other recent works discussing the necessary assumptions for NMF to be unique include [Theis et al., 2005] and [Huang et al., 2013].

On the other hand, second problem can be tackled only by using new, more complicated types of NMF that produce the hierarchy of factors. The following types of NMF were proposed in order to deal with this issue: Convolutive NMF, Overlapping NMF and Multilayer NMF [Cichocki et al., 2009].

Convolutive NMF is a natural generalization of NMF where the set of horizontally shifted versions of the initial matrix is used in the optimization procedure. This particular type of NMF can prove to be very efficient when working with audio signals whose frequencies vary in time. This model is, however, not hierarchical (it minimizes the sum of decompositions corresponding to shifted matrices) and quite application-dependent.

Overlapping NMF is pretty much the same as Convolutive NMF with the only difference that we process vertically shifted versions of the primary matrix. Obviously, it suffers from the same disadvantages.

In our work, we will consider the Multilayer NMF that was introduced in Chapter 2 as it deals with both issues of the simple NMF model. Considering current success of deep and representation learning in real-world applications, this type of NMF is the best candidate from the family of NMF methods that proved to learn hierarchical features [Trigeorgis et al., 2014] and enjoys a good numerical stability with the increasing number of layers.

Sparsity is an another key property of many machine learning algorithms. The effect of reducing the number of components was first observed by neuroscientists who were studying mammalian brain activity (see [Olshausen and Field, 2004]). There are two key concepts related to sparsity: sparse activity and sparse connectivity [Thom and Palm, 2013a]. The sparse activity means that only a small number of elements is active at any time. The small connectivity reveals the same concept applied to connections between elements. Both properties can be demonstrated using a simple NMF model with two factors. If the prototypes matrix is sparse, we try to reconstruct a given object by a reduced number of features. On the other hand, if partition matrix is sparse, we assume that there is a small number of activations of prototypes for a given object.

As sparsity plays an important role in learning features, it is worth mentioning that Multilayer NMF is considered to increase gradually the sparseness of obtained basis matrices even though there was no empirical or theoretical study about this.

1.2 Our contributions

In our work we can highlight two main contributions:

- We studied the Multilayer NMF using Hoyer's projection operator;

- We analyzed the evolution of factors' sparsity obtained at each layer.

The rest of this appendix is organized as follows: in section 2, we will briefly introduce basic notations of Standard and Multilayer NMF, in section 3 we present our analysis of Multilayer NMF using Hoyer's projection operator. We will summarize the results in section 4. Finally, we will conclude and point out some ideas about the future extensions of our analysis in section 5.

2 Preliminary knowledge

In this section, we describe some basic notations and techniques that are used later. We start by introducing the Deep NMF model.

2.1 Deep NMF

For the Deep NMF, we need to fine-tune the two factors in each layer, in order to reduce the total reconstruction error of the model, by employing alternating minimization of the following cost function:

$$C_{Deep}(X, W, H) = \frac{1}{2}\|X - W_1 W_2 ... W_L H_L\|^2 = \\ = tr[X^T X - 2X^T W_1 W_2 ... W_L H_L + H_L^T W_L^T W_{L-1}^T ... W_1^T W_1 W_2 ... W_L H_L].$$

Another approach used in Lyu and Wang [2013] proposes to optimize the whole sequence of factors in one single optimization procedure based on the solution to stochastic matrix sandwich problem. It further uses the Dirichlet sparsity regularizer to reinforce sparsity of the obtained matrices.

2.2 Hoyer's normalized sparsity measure

In [Hoyer and Dayan, 2004] a new sparseness measure based on the ratio of $\ell 1$ and $\ell 2$ norms was proposed in order to evaluate the sparseness of a given vector x. It is

defined as follows:

$$s : \mathbb{R}^n \setminus \{0\} \rightarrow [0;1],\ x \rightarrow \frac{\sqrt{n} - \frac{\|x\|_1}{\|x\|_2}}{\sqrt{n}-1}.$$

This sparseness measure satisfies all criteria of a relevant sparseness measure described in [Hurley and Rickard, 2009]. s is scale-invariant and differentiable on its entire domain [Thom and Palm, 2013a].

To exploit this sparseness measure, a sparseness-enhancing projection operator was proposed in order to be used with projected gradient descent algorithms. For a given degree of sparsity s^*, the operator finds the closest vector in the Euclidian sense that has a desired level of sparsity s^*, given any arbitrary vector. More formally, Hoyer's projection operator is defined as follows:

$$S_H(s1, s2) = \{s \in \mathbb{R}^n | \|s\|_1 = s_1, \|s\|_2 = s_2\}.$$

A variation of this operator when one wants to preserve nonnegativity is simply done by searching for feasible solutions only in the positive orthant $S_H(s1, s2) \cap \mathbb{R}^n_{\geq 0}$.

3 Analysis of Multilayer NMF

In this section we will study the Multilayer NMF using Hoyer's projection operator. We will use it in order to build a sequence of projected prototype matrices starting from the second layer of Multilayer NMF. The main idea that we want to investigate here is two-fold: (1) what happens at each layer of NMF in terms of features sparsity? and (2) what lies underneath the hierarchical learning procedure, i.e. what is the relationships between the features obtained at different layers? To answer these two questions, we propose a very simple procedure that aims at finding the closest projection of a given feature that has the same properties as a feature of the next level, given the previous level's set of features.

As it can be seen from the definition, this projection operator allows us to obtain a projection of the initial vector x with the desired level of sparsity simply by manipu-

lating with its $\ell 1$ and $\ell 2$ norms. Usually, however, we want the resulting vector to be normalized. In this case, the value of $\ell 2$ norm can be set to 1 and the value of $\ell 1$ norm can be easily derived from the above equation.

3.1 Proposed approach

In order to build a sequence of projected matrices, we start by performing first two iterations of the Multilayer NMF:

$$X \simeq W_1 H_1, H_1 \simeq W_2 H_2.$$

We cannot directly project first prototype matrix W_1 because second factor W_2 is supposed to have a different size, i.e., $W_1 \in \mathbb{R}^{n \times k}$, $W_2 \in \mathbb{R}^{k \times k}$. After two iterations, we obtain W_3 by projecting W_2 using Hoyer's operator with the following parameters: $\forall\, W_i \;\; \|W_i\|_1 = k_{W_i},\; \|W_i\|_2 = 1$ where

$$k_{W_i} = \sqrt{m} - \frac{\sqrt{m} - \frac{\|W_i\|_1}{\|x\|_2}}{\sqrt{m} - 1}(\sqrt{m} - 1).$$

More precisely, we fix the $\ell 2$ norm of all $\{W_i\}_{i=2..L}$ to 1 and we calculate the $\ell 1$ norm of the projection in terms of the sparsity of factors arising from Multilayer NMF.

Finally, the expression for our Projected Multilayer NMF (PMNMF) takes the following form:

$$W_{p_i} = S_H \left(\underbrace{\ldots}_{i-2} S_H(S_H(W_2, k_{W_2}, 1) \right).$$

We can see directly that the proposed approach has to be more efficient than the Multilayer NMF as it does not need to perform NMF to obtain the factors of each layer. We will further analyze its computational complexity in order to show that it is, indeed, the case.

The proposed algorithm is summarized in Algorithm 1.

Algorithm 3: Construction of a sequence of projected matrices

Data: X - initial data set, n - number of clusters, m - number of layers, n_b - number of features

Result: W_p - sequence of projected matrices

initialization;

$[W_a\ H] \leftarrow nmf(X, n)$;

for $i \leftarrow 1$ **to** m **do**

  $[W\ H_n] \leftarrow nmf(H, n)$;

  $W_a \leftarrow W_a * W$;

  $s \leftarrow sparsity(W_a)$;

  for $k = 1..n$ **do**

    $k_p \leftarrow \sqrt{n_b} - (\sqrt{n_b}) - 1) * s$;

    $W_p(k) \leftarrow S_H(W(:,k)/\|W(:,k)\|_2, k_p, 1)$;

3.2 Complexity analysis

We will now discuss more in detail the computational complexity of the Multilayer NMF when compared to the Projective Multilayer NMF. As presented above, we will consider the hierarchical system that consists of L layers. First, we note that the computational complexity of the original optimization procedure with multiplicative update rules for Standard NMF has the computational complexity $\mathcal{O}(tknm)$ for a given input matrix $X \in \mathbb{R}^{m \times n}$ where t denotes the number of iterations used to minimize the cost function (usually, $t \approx 100$). As first two iterations are the same for both approaches, we obtain the following computational complexity $\mathcal{O}(tknm + tk^2n)$. After that, the Multilayer NMF builds the following $(L-1)$ factors in the same manner that leads to the total computational complexity $\mathcal{O}(tknm + Ltk^2n)$. At this point, the Projective Multilayer NMF deviates from the original approach and calculates all $(L-2)$ factors left using Hoyer's projection operator. When using an appropriate procedure, it can be proved that the projection of a given vector $x \in \mathbb{R}^n$ can be calculated in linear time and in constant space [Thom and Palm, 2013b]. Thus, the complexity of the proposed approach is equal to $\mathcal{O}(tknm + tk^2n + k^2L)$.

Comparing the complexities of each method, we can conclude by saying that the proposed approach can be almost L times more efficient than the Multilayer NMF when Standard NMF is applied to obtain the factors. Furthermore, we considered the most optimistic scenario when Standard NMF is used at each layer which is usually not the case as the latter gives less homogeneous clustering results when compared to other variations of NMF with additional constraints on the factors. In this case, if Standard NMF is replaced with three factor NMF (for instance, Convex NMF or Tri-NMF [Ding et al., 2010a]) the computational complexity of Multilayer NMF will increase further.

3.3 Theoretical analysis

In general, it is hard to analyze NMF-based methods as the cost function that has to be minimized is non-convex (it is actually multi-convex, i.e., convex in each of the factors). Furthermore, there is no closed-form solution that allows to analyze analytically the properties of factors in terms of sparsity, orthogonality or their clustering abilities. We can see, however, that different layers of Multifactor NMF have a different meaning. First prototype matrix W_1 represents a learned dictionary that captures the essential coded information about the geometrical structure of the initial data. The matrices $\{W_i\}_i = 2..L$ are destined to refine the initial prototype matrix in order to make it more discriminant. To this end, we would like to try to find an answer to the following question: " under what conditions the initial coding obtained in W_1 can be preserved and refined using hidden layers?"

To answer this question, we will use the Johnson-Lindenstrauss lemma [Johnson et al., 1984]. This lemma is formulated as follows:

Lemma. *Let $\epsilon \in (0; \frac{1}{2})$ be a real number, and $X = \{x_1, x_2, ..., x_n\}$ be a set of n in space $\mathbb{R}^d$. Let k be an integer with $k \geq C\epsilon^{-2}log(n)$, where C is a sufficiently large constant. Then there exists a linear mapping $f : \mathbb{R}^m \to \mathbb{R}^k$ such that:*

$$(1 - \epsilon)\|x_i - x_j\|_2 \leq \|f(x_i) - f(x_j)\|_2 \leq (1 + \epsilon)\|x_i - x_j\|_2$$

for all $i, j = 1, 2, ..., n$.

Multiple solutions were proposed to obtain the linear mapping that satisfies the JL condition. The pioneer work [Johnson et al., 1984] builds transformation matrix by choosing randomly a k-dimensional subspace of X with orthogonal columns and then rescales it with respect to k and d. After that, [Indyk and Motwani, 1998] proved that it is possible to replace a random k-dimensional subspace with k random Gaussian vectors that are distributed iid. from $\mathcal{N}(0, 1)$. Finally, one of the most recent work on this subject shows that the linear transform can be represented by a sparse matrix where the level of sparsity can be defined beforehand [Ailon and Chazelle, 2006].

From a practical point of view, the linear mapping that reduces dimensionality of the initial set of vectors and preserves the distance for an arbitrary pair of points has found its application in numerous dimensionality reduction techniques. For instance, solving the ϵ-approximate nearest neighbor problem that aims at finding the closest Euclidian projection of a given vector x.

In this context, we can treat matrix $W_1 \in \mathbb{R}^{m\times k}$ arising from the first iteration of Multilayer NMF as a linear transformation that reduces the dimensionality of $X \in \mathbb{R}^{m\times n}$ to $H_1 \in \mathbb{R}^{m\times n}$. If we assume that W_1 fulfills the JL condition then we can rewrite the latter as follows:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|W_1 x_i - W_1 x_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2.$$

Indeed, we consider W as a good dictionary for clustering if the distortion between the initial data and the features is low. In this case, the distances between the data points are preserved and we can expect to have a good clustering performance when using H_1. If we further continue with Multilayer NMF, we obtain the following inequality for the kth layer:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\prod_{l=1}^{k} W_l x_i - \prod_{l=1}^{k} W_l x_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2,$$

where $\prod_{l=1}^{k} W_l \in \mathbb{R}^{m\times k}$ is a transformation matrix for X and H_k.

In the case of Projective Multilayer NMF, Hoyer's operator seeks to find a projec-

tion of a given vector x on D where

$$D = \left\{ s \in \mathbb{R}^d_{\geq 0} | \|s_1\|_2 = \lambda_1, \|s_2\|_2 = 1 \right\}.$$

We now rewrite the JL condition for Projective Multilayer NMF:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|W_1 W_2 \prod_{l=1}^{k} S_H(\underbrace{\ldots}_{l} W_2) x_i - W_1 W_2 \prod_{l=1}^{k} S_H(\underbrace{\ldots}_{l} W_2) x_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2.$$

The standard NMF that is used to obtain the factors of each layer does not impose additional constraints on the factors. It means that multiplying the initial learned dictionary W_1 with the factors W_2 to W_L will most likely increase the length of the initial feature vectors and thus increase the distortion between them and the initial data. Increasing distortion, in its turn, may decrease the clustering performance. This observation can also explain the fact that in both [Trigeorgis et al., 2014] and [Lyu and Wang, 2013] the best results where obtained for multifactor decomposition with 2 hidden layers.

Contrary to Multilayer NMF, matrices $\{W_i\}_{i=3..L}$ obtained using Hoyer's projection operator have the length of all their columns equal to one and the sparsity of the factors can be defined by changing the values of $\ell 1$ norm. In our opinion, this property can play a crucial role in building more complex models with a large number of hidden layers that fine-tune the prototype matrices while maintaining a low distortion with respect to the initial data.

4 Experimental results

In this section we will present the experimental results obtained using our approach for some famous benchmark data sets used in face and image recognition.

4.1 Data sets and evaluation criteria

We evaluate the proposed approach using some well-known image data sets such as:

- Yale data set (165 images of 15 individuals);
- ORL (400 images of 40 different individuals of size 32x32);
- MNIST (70000 images of handwritten digits of size 28x28):
- USPS (9298 images of 10 different digits of size 16x16);
- PIE (41368 images of 68 different individuals).

Figure A.1 shows example photos from Yale, ORL and PIE data sets.

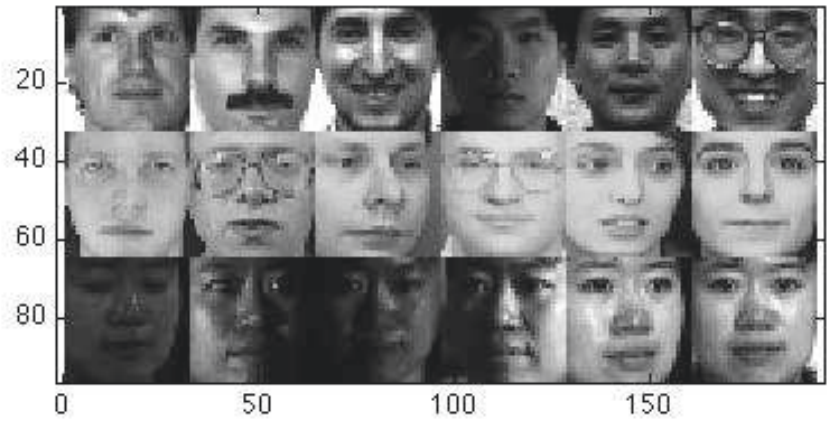


Figure A.1: Samples from Yale, ORL and PIE data sets

We performed 10 fold cross-validation for each of them and evaluated the results using purity [Rendon et al., 2011].

Purity is the standard measure of clustering quality in supervised setting. Given a particular cluster S_r of size n_r, the purity of this cluster is defined to be:

$$Pu(S_r) = \frac{1}{n_r} \max_i n_r^i.$$

Larger purity values indicate better clustering solutions.

4.2 Analysis of Multilayer NMF using Projected Multilayer NMF

The analysis that we present consists of three main parts:

- we analyze purity values obtained at each level for both MNMF and PMNMF in order to see if the obtained features lead to the same results;
- we compare projected factors to real factors of Multilayer NMF by calculating the $\ell 1$ norm as $\ell 2$ norm is fixed to 1 for all projections. We recall that bigger $\ell 1$ values of factors stand for lower sparsity values;
- we present sparsity values for both sequences of matrices and show that it decreases from layer to layer.

The results for all five data sets are presented in Figure A.2-Figure A.6.

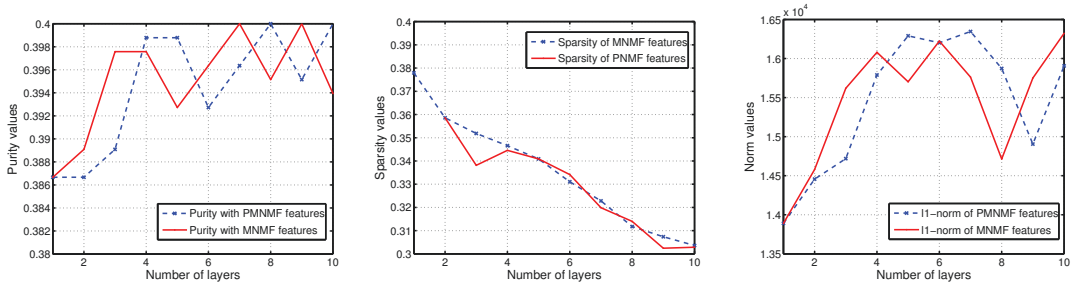


Figure A.2: Results on purity, sparsity and $\ell 1$ norm of features on Yale data set

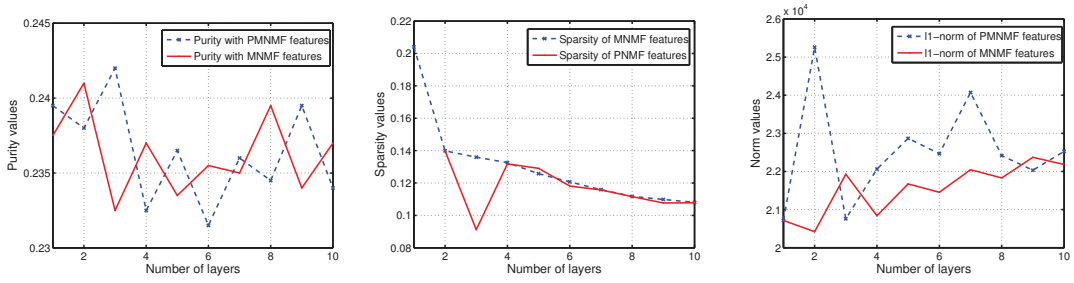


Figure A.3: Results on purity, sparsity and $\ell 1$ norm of features on ORL data set

These figures show that classification accuracy values obtained at each layer using both Projected and Standard Multilayer NMF are almost identical. It means that our

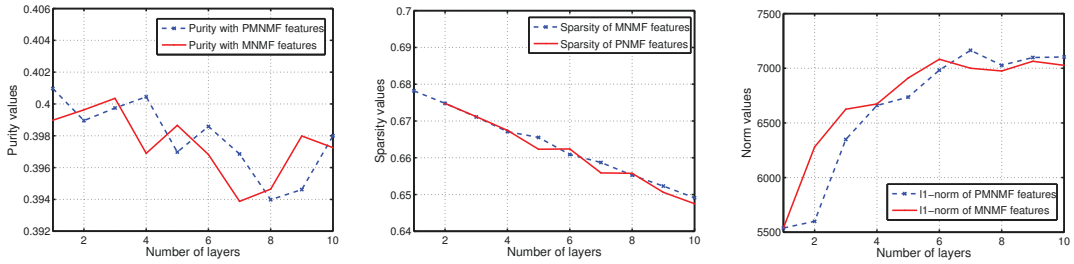


Figure A.4: Results on purity, sparsity and $\ell 1$ norm of features on PIE data set

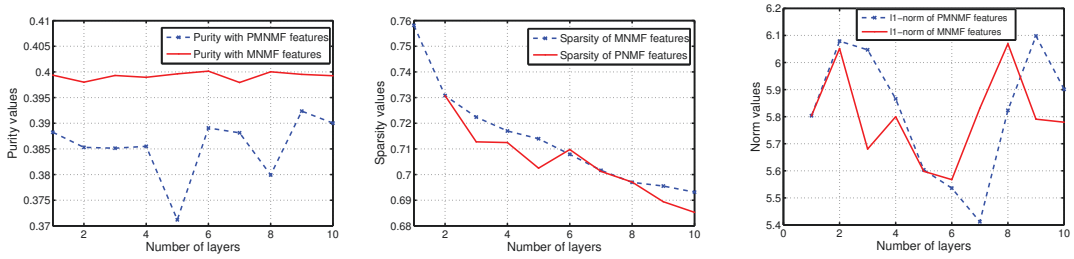


Figure A.5: Results on purity, sparsity and $\ell 1$ norm of features on USPS data set

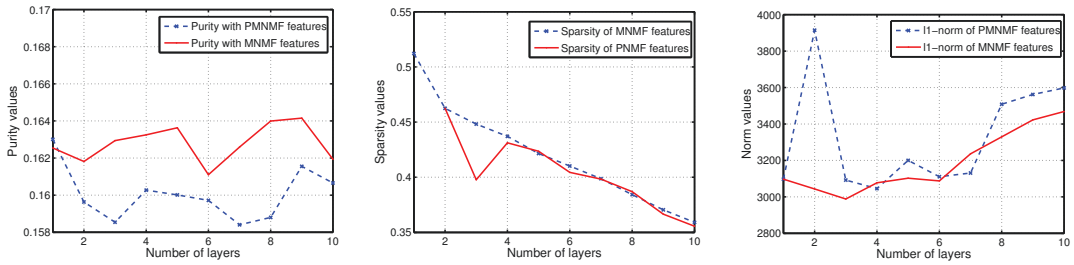


Figure A.6: Results on purity, sparsity and $\ell 1$ norm of features on MNIST data set

approach works just as we intended, i.e., it follows the original approach and preserves its clustering performance at each layer.

We can also see that sparsity decreases at each layer while $\ell 1$ norm is increasing. In terms of $\ell 1$ norm, our approach is once again very close to the original Multilayer NMF.

The results on sparsity and $\ell 1$ norm are rather surprising as it is common to justify the improved performance of the Multilayer NMF by the fact that the sparsity of factors is enhanced when one increases the number of layers. However, this result agrees well with [Berkes et al., 2009] where the neural responses to natural moves in the primary visual cortex of ferrets were studied. In contrast with prediction from a sparse coding model, the main conclusion of this work is that the representation in the primary visual cortex is not actively optimized to maximize sparseness. It seems logical that complex models used by visual cortex can be simplified only to some degree of sparsity where they remain discriminant. At this point, further sparsification of features can hurt the performance of classification due to the fact that they won't be able to capture the differences between objects in their fullest.

5 Conclusions

In this appendix we analyzed Multilayer NMF using Hoyer's projection operator. Our main idea was to show that adding depth to NMF can be achieved by simply projecting arising factors with a certain level of sparsity. This result can be seen as a variation of pooling approaches that are widely used in Deep neural networks. We also observed that for five chosen data sets sparsity decreases during the optimization procedure at each layer. This result is rather surprising as it was common to suppose that enhanced sparsity of prototypes makes Multilayer NMF more robust. Therefore, we conclude that hierarchical feature learning in general can be seen as a sequential application of projections as if one was using regularization imposed on obtained factors. From theoretical point of view, we used Johnson-Lindenstrauss lemma to show that the hidden layers learned using Hoyer's projection operator are more likely to produce a dictionary that keeps the distortion low with respect to the initial data and has a good clustering performance. Finally, the proposed approach is more computationally efficient than

the original one even when the simplest form of NMF is used at each layer.

In the future our work can be extended in multiple directions. Learning hierarchical structures are commonly used in lots of machine learning techniques. It means that our approach can be further extended and tested on them in the same fashion as it was done for NMF. For instance, one can use it for hierarchical feature learning in most powerful machine learning techniques such as Deep Neural Networks. However, determining the level of sparsity that ensures discriminative power of features and avoids overfitting remains an open problem. Finally, another interesting direction is to proceed in the same manner as the Deep NMF does - that is to add a global learning step for all layers. In this case, one can expect to learn hierarchical features minimize well the reconstruction error.

Appendix B

In this appendix we present figures for the rest of Office/Caltech data set's pairs divergence measures comparison from Chapter 6. In general, we can observe that MMD distance fails to follow the shape of the true target error in the following cases: D,W $\rightarrow$ A,C. These scenarios represent a situation where the source tasks is small compared to the target task while the main setting for the transfer learning is exactly the opposite. On the other hand, $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence does not follow the true error in more than half of the cases (A $\rightarrow$ D, C $\rightarrow$ D, D $\rightarrow$ A, D $\rightarrow$ C, W $\rightarrow$ A, W $\rightarrow$ D).

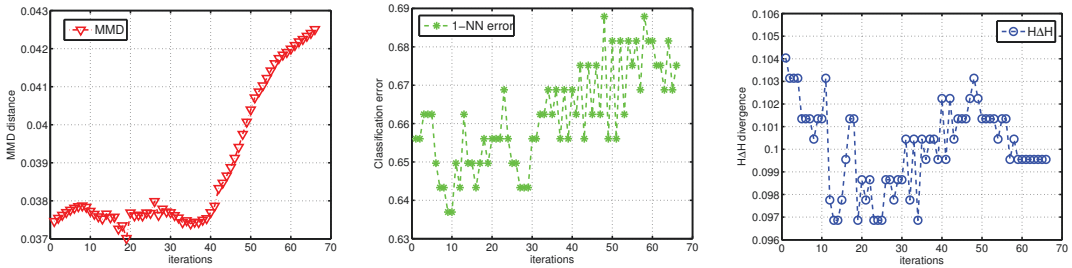


Figure B.1: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Amazon/DSLR pair of tasks using ITLDC

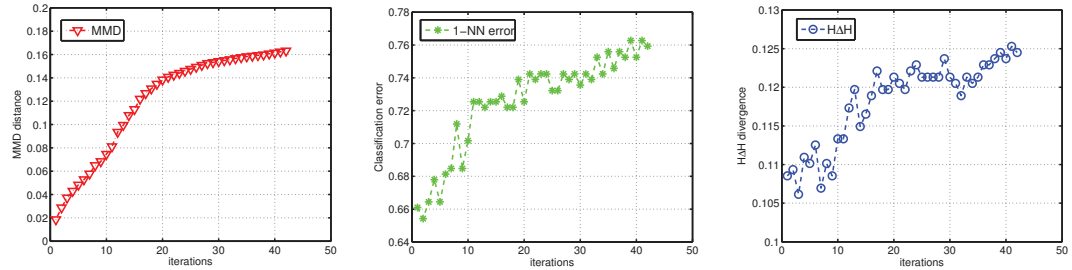


Figure B.2: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Amazon/Webcam pair of tasks using ITLDC

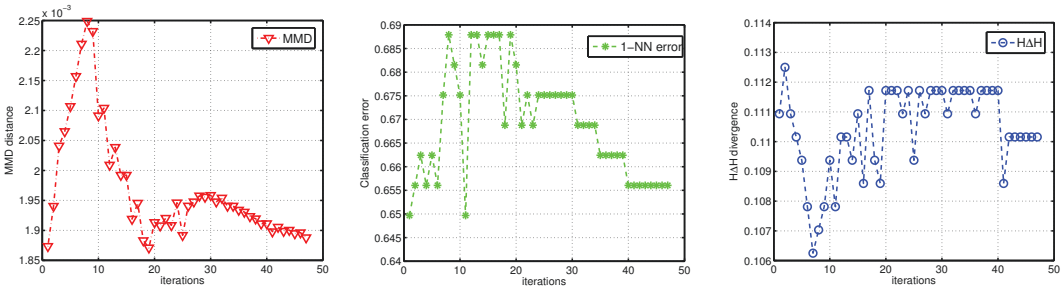


Figure B.3: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Caltech/DSLR pair of tasks using ITLDC

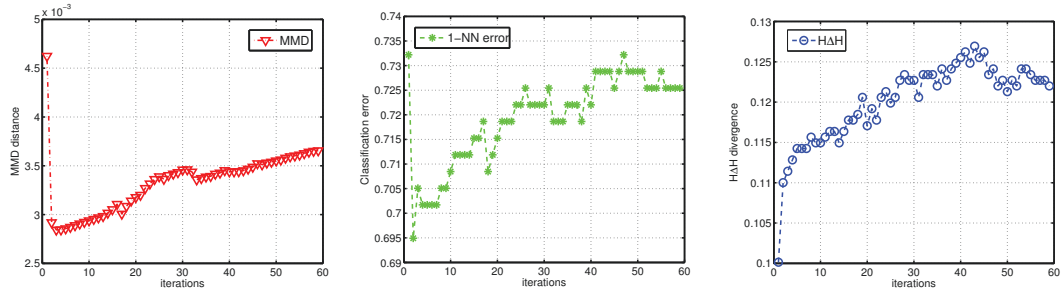


Figure B.4: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Caltech/Webcam pair of tasks using ITLDC

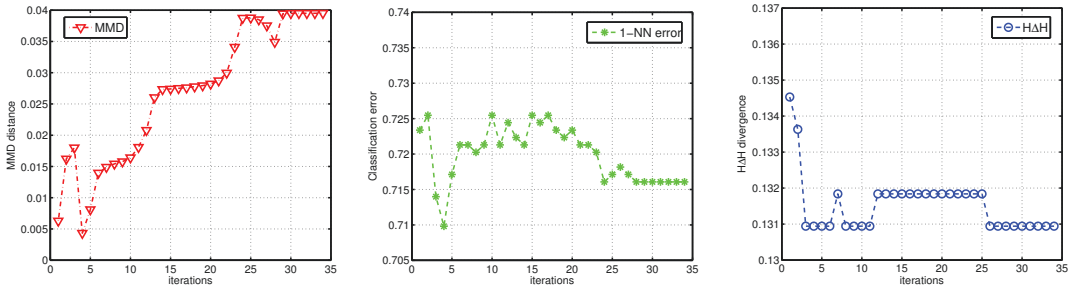


Figure B.5: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on DSLR/Amazon pair of tasks using ITLDC

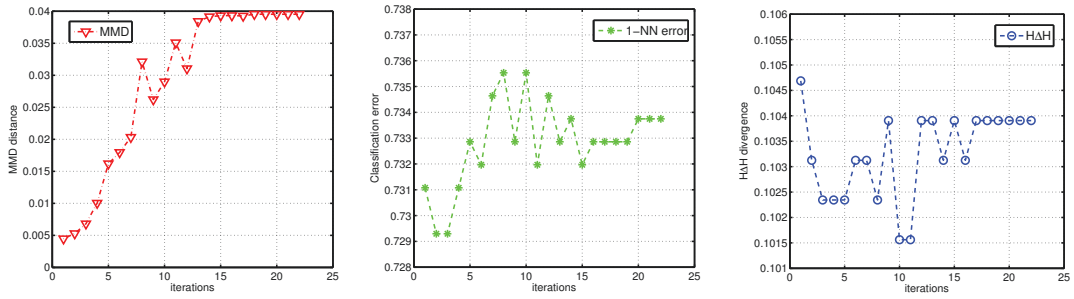


Figure B.6: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on DSLR/Caltech pair of tasks using ITLDC

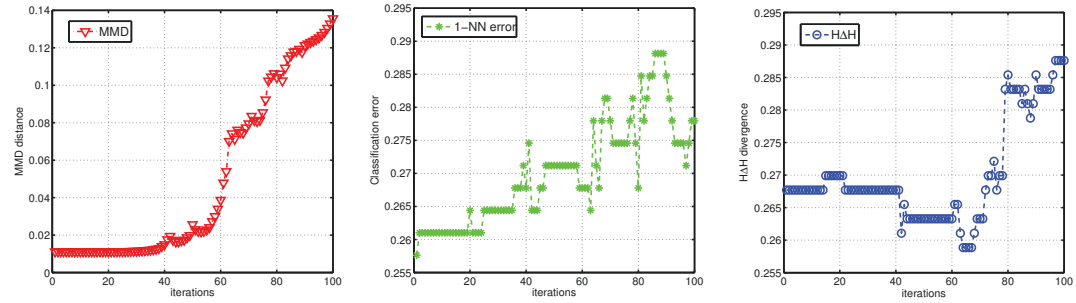


Figure B.7: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on DSLR/Webcam pair of tasks using ITLDC

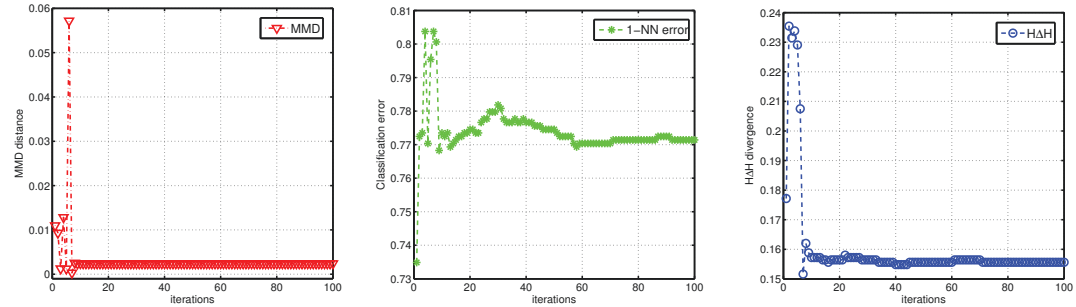


Figure B.8: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Webcam/Amazon pair of tasks using ITLDC

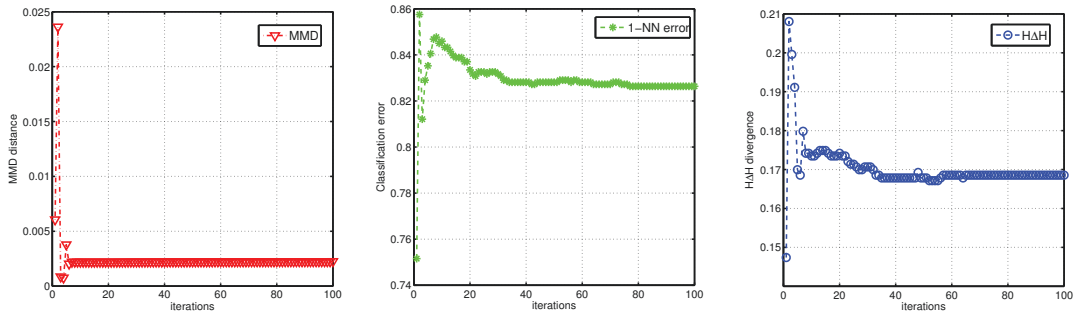


Figure B.9: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Webcam/Caltech pair of tasks using ITLDC

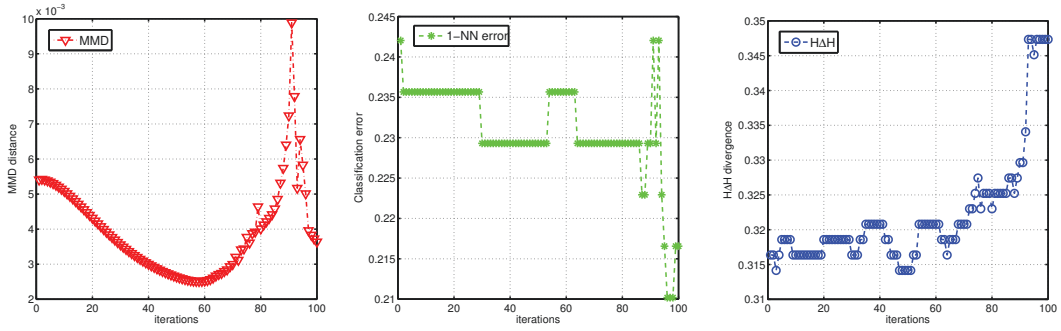


Figure B.10: $\hat{d}_{MMD}$ distance, 1-NN classification error and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence on Webcam/DSLR data set using ITLDC

List of publications

1. REDKO I., PODLADCHIKOVA T., PODLADCHIKOV V. (2011), Increasing of distribution parameters estimation efficiency under small sample size conditions, System Analysis and Information Technologies.

2. REDKO I., BENNANI Y. (2014), Random subspaces NMF for unsupervised transfer learning, Proceedings of the (IJCNN'14), International Joint Conference on Neural Networks, Beijing, China.

3. REDKO I., BENNANI Y. (2014), Controlling orthogonality constraints for better NMF clustering, Proceedings of the (IJCNN'14), International Joint Conference on Neural Networks, Beijing, China.

4. REDKO I., BENNANI Y. (2014), Non-negative Matrix Factorization with Schatten p-norms Reguralization, Proceedings of the (ICONIP'14), International Conference on Neural Information Processing, Kuching, Malaysia.

5. REDKO I., BENNANI Y. (2014), Universal Unsupervised Transfer Learning through Non-negative Matrix Factorization, AutoML workshop, International Conference on Machine Learning, Beijing, China.

6. REDKO I., BENNANI Y. (2014), NMF multi-couches aleatoire pour lapprentissage par transfert non-supervise, Revue des Nouvelles Technologies de lInformation.

7. REDKO I., BENNANI Y. (2015), Sparsity Analysis of Learned Factors in Multilayer NMF, Proceedings of the (IJCNN'15), International Joint Conference on Neural Networks, Killarney, Ireland.

8. REDKO I., BENNANI Y. (2015), Non-negative Embedding for Fully Unsupervised Domain Adaptation, submitted to Pattern Recognition Letters.

9. REDKO I., BENNANI Y. (2015), Domain Adaptation using Hilbert-Schmidt embeddings, submitted to Machine Learning Journal.

References

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 557–563, 2006. 114

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, December 2005. 38

Hubert L.J. Arabie, P. An overview of combinatorial data analysis, 1996. 10

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*. MIT Press, 2007. 27

Andreas Argyriou, Massimiliano Pontil, Charles A. Micchelli, and Yiming Ying. A spectral regularization framework for multi-task structure learning. In *In NIPS*, 2008. 28

Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, ICDMW '07, pages 77–82, 2007. 32, 37

Anthony Aue and Michael Gamon. Customizing sentiment classifiers to new domains:

A case study. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, 2005. 36

Liviu Badea. Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization. pages 267–278. World Scientific, 2008. 62

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Jourank of Machine Learning Research*, 3:463–482, 2003. 93

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000. 31

Shai Ben-David, John Blitzer, Koby Crammer, and O Pereira. Analysis of representations for domain adaptation. In *In NIPS*. MIT Press, 2007. 76

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010a. 41, 76, 77, 78, 79, 81, 82, 88, 90, 95, 96, 100

Shai Ben-David, Tyler Lu, Teresa Luu, and Dvid Pl. Impossibility theorems for domain adaptation. In *AISTATS*, volume 9, pages 129–136, 2010b. 76

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. 27

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1): 39–71, March 1996. 37

Pietro Berkes, Ben White, and Jzsef Fiser. No evidence for active sparsification in the visual cortex. In Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, editors, *NIPS*, pages 108–116. Curran Associates, Inc., 2009. 119

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 81–88, 2007. 34

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 120–128, 2006. 38

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '07, pages 187–205, 2007. 39, 45

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jenn Wortman. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems 21*, Cambridge, MA, 2008. MIT Press. 76

Edwin Bonilla, Kian Ming Chai, and Chris Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, pages 153–160. 2008. 30

Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. Non-negative matrix factorization on manifold. In *Proc. Int. Conf. on Data Mining (ICDM'08)*, 2008. 43

Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010. 50

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*, KDD '09, pages 179–188, 2009. 37, 38, 64, 87

Zheng Chen and Weixiong Zhang. Domain adaptation with topical correspondence learning. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 1280–1286, 2013a. 64

Zheng Chen and Weixiong Zhang. Domain adaptation with topical correspondence learning. In *IJCAI*, 2013b. 42, 44

Massimiliano Ciaramita and Olivier Chapelle. Adaptive parameters for entity recognition with perceptron hmms. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, 2010. 39

Andrzej Cichocki and Rafal Zdunek. Multilayer nonnegative matrix factorization using projected gradient approaches. *International Journal of Neural Systems*, 17:431–446, 2007. 16

Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley Publishing, 2009. 107

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014. 35, 76, 81, 82

Nicolas Courty, Remi Flamary, Alain Rakotomamonjy, and Devis Tuia. Optimal transport for domain adaptation. In *NIPS, Workshop on Optimal Transport and Machine Learning*, Montréal, Canada, December 2014. 82

Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9:1757–1774, 2008. 77

Shawe-Taylor J. Elisseeff A. Cristianini, N. and J. Kandola. On kernel-target alignment. *Advances in Neural Information Processing Systems*, 14:367–373, 2002. 50, 53

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 193–200, 2007. 26

Wenyuan Dai, Qiang Yang 0001, Gui-Rong Xue, and Yong Yu. Self-taught clustering. *Proceedings of the 25th International Conference on Machine Learning*, 307:200–207, 2008a. 41, 57

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 200–207, 2008b. 64

Jesse Davis and Pedro Domingos. Deep transfer via second-order markov logic. In *Proceedings of the 26th International Conference on Machine Learning (ICML-09)*, 2009. 32

Chris H. Q. Ding and Xiaofeng He. On the equivalence of nonnegative matrix factorization and spectral clustering. *In Proc. of SIAM International Conference on Data Mining*, pages 606–610, 2005. 15

Chris H. Q. Ding, Tao Li, and Michael I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (1):45–55, 2010a. 107, 113

Chris H. Q. Ding, Tao Li, and Michael I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:45–55, 2010b. 12, 13, 14, 15

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts ? *Advances in Neural Information Processing Systems*, 17, 2004. 107

Lixin Duan, Ivor W. Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, March 2012. 45

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000. 10

Richard M. Dudley. *Real analysis and probability*. Cambridge studies in advanced mathematics. Cambridge University Press, 2002. 84

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 109–117, 2004. 31

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 2960–2967, 2013. 39, 40, 64, 66

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm, 1996. 26

Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 283–291, 2008. 31

Z. Gao and A. Galvao. Minimum Integrated Distance Estimation in Simultaneous Equation Models. *ArXiv e-prints*, 2014. 84

Bo Geng, Dacheng Tao, and Chao Xu. DAML: domain adaptation metric learning. *IEEE Transactions on Image Processing*, 20(10):2980–2989, 2011. 87

Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1991. 10

Nicolas Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13(1):3349–3386, 2012. 107

Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of the 30th International Conference on Machine Learning*, pages 222–230, 2013a. 39

Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, pages 222–230, 2013b. 58, 72

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 999–1006, 2011. 45

Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 2066–2073, 2012. 39, 64, 66

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of the 16th International Conference on Algorithmic Learning Theory*, pages 63–77, 2005. 55

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012. 89, 91

Steffen Grünewälder, Guy Lever, Arthur Gretton, Luca Baldassarre, Sam Patterson, and Massimiliano Pontil. Conditional mean embeddings as regressors. In *ICML*, 2012. 87

Quanquan Gu, Zhenhui Li, and Jiawei Han. Joint feature selection and subspace learning. In *IJCAI*, pages 1294–1299. IJCAI/AAAI, 2011. 72

Yuhong Guo and Min Xiao. Cross language text classification via subspace co-regularized multi-view learning . In *ICML*, 2012. 28

Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham W. Taylor, and David W. Aha. Unsupervised and transfer learning challenge. In *The 2011 International Joint Conference on Neural Networks, IJCNN*, pages 793–800, 2011. 36

Daniel Haase, Erik Rodner, and Joachim Denzler. Instance-weighted transfer learning of active appearance models. June 2014. 26

Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000. 10

Patrik O. Hoyer and Peter Dayan. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004. 45, 109

J. Huang, A.J. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, volume 19, 2007. 34

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schlkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, pages 601–608, 2006. 87

Kejun Huang, Nikos D. Sidiropoulos, and A. Swamiy. NMF revisited: New uniqueness results and algorithms. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 4524–4528, 2013. 107

N. Hurley and S. Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, Oct. 2009. 110

Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat, 2001. 67

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, 1998. 114

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999. 10

Tony Jebara. Multi-task feature and kernel selection for svms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, 2004. 27

Jing Jiang and Chengxiang Zhai. Instance weighting for domain adaptation in nlp. In *In ACL 2007*, pages 264–271, 2007. 25

Wenbin Jiang, Yajuan L, Liang Huang, and Qun Liu. Automatic adaptation of annotations. pages 1–29, 2015. 32

Wenhao Jiang and Fu-Lai Chung. Transfer spectral clustering. *Proceedings of the ECML/PKDD*, pages 789–803, 2012. 41, 59

William B. Johnson, Joram Lindenstrauss, and Gideon Schechtman. Extensions of Lipschitz mappings into a Hilbert space., 1984. 113, 114

Leonid Kantorovich. On the translocation of masses. In *C.R. (Doklady) Acad. Sci. URSS(N.S.)*, volume 37, page 199201, 1942. 83

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 180–191, 2004. 76, 81

N. Kriegeskorte, M. Mur, and P. Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2(4):1–28, 2008. 52

Neil D. Lawrence and John C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML'04, 2004. 30

Yann LeCun and Corinna Cortes. MNIST handwritten digit database, 2010. URL `http://yann.lecun.com/exdb/mnist/`. 18

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999. xii, 11, 12, 17, 18

Honglak Lee, Rajat Raina, Alex Teichman, and Andrew Y. Ng. Exponential family sparse coding with application to self-taught learning. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1113–1119, 2009. 30

S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller. Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of International Conference on Machine Learning (ICML)*, 2007. 28

Tao Li, Chris H. Q. Ding, and Michael I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM*, pages 577–582, 2007. 55, 104

Xuejun Liao, Ya Xue, and Lawrence Carin. Logistic regression with an auxiliary data source. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, 2005. 25

Joseph J. Lim, Ruslan Salakhutdinov, and Antonio Torralba. Transfer learning by borrowing examples for multiclass object detection. In *Neural Information Processing Systems (NIPS)*, 2011. 26

Chih-Jen Lin. On the convergence of multiplicative update algorithms for non-negative matrix factorization. *Transactions on Neural Networks*, 18(6):1589–1596, 2007. 15

Anqi Liu and Brian D. Ziebart. Robust classification under sample selection bias. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 37–45, 2014. 35

Mingsheng Long, Jianmin Wang, Guiguang Ding, Wei Cheng, Xiang Zhang, and Wei Wang. Dual transfer learning. In *SDM*, pages 540–551, 2012. 42, 43, 64

Mingsheng Long, Jianmin Wang, Guiguang Ding, Sinno Jialin Pan, and Philip S. Yu. Adaptation regularization: A general framework for transfer learning. *IEEE Transactions on Knowledge Data Engineering*, 26(5):1076–1089, 2014a. 38

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer joint matching for unsupervised domain adaptation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1417, 2014b. 39, 40, 71

Siwei Lyu and Xin Wang. On algorithms for sparse multi-factor nmf. In *Advances in Neural Information Processing Systems 2013.*, pages 602–610, 2013. 109, 115

M.M.H. Mahmud and S.R. Ray. Transfer learning using kolmogorov complexity: Basic theory and empirical evaluations. *Proc. 20th Ann. Conf. Neural Information Processing Systems*, pages 985–992, 2008. 49

Yishay Mansour and Mariano Schain. Robust domain adaptation. *Mathematics and Artificial Intelligence*, 71(4):365–380, 2014. 77

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT 2009 - The 22nd Conference on Learning Theory, 2009*, 2009. 76, 77, 79, 81, 82, 96

Anna Margolis. A literature review on domain adaptation with unlabeled data, 2011. 36

Anna Margolis, Karen Livescu, and Mari Ostendorf. *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, chapter Domain Adaptation with Unlabeled Data for Dialog Act Tagging, pages 45–52. 2010. 37, 39

K. Markov and T. Matsui. Nonnegative matrix factorization based self-taught learning with application to music genre classification. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, page 15, 2012. 42, 44

Andreas Maurer, Massi Pontil, and Bernardino Romera-paredes. Sparse coding for multitask and transfer learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 343–351, May 2013. 29

Lilyana Mihalkova and Raymond J. Mooney. Transfer learning by mapping with minimal target data. July . 32

Lilyana Mihalkova, Tuyen Huynh, and Raymond J. Mooney. Mapping and revising markov logic networks for transfer learning. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, pages 608–614, 2007. 31

Andri Mirzal. Converged algorithms for orthogonal non-negative matrix factorizations. *CoRR*, abs/1010.5290, 2010. 14

Ankur Moitra. A singly-exponential time algorithm for computing nonnegative rank. *CoRR*, abs/1205.0044, 2012. 67

J. Neumann, C. Schnörr, and G. Steidl. Combined SVM-based Feature Selection and Classification. *Machine Learning*, 61:129–150, 2005. 53

Hiroki Ogino and Tetsuya Yoshida. Topic graph based non-negative matrix factorization for transfer learning. In *ISMIS*, pages 260–269, 2011. 42

B. A. Olshausen and D. J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–487, 2004. 108

Sinno Jialin Pan, James T. Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, AAAI'08, pages 677–682, 2008. 37, 38, 87

Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1187–1192, 2009. 37, 38, 64, 87

S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010. xii, 23, 24

Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33:pp. 1065–1076, 1962. 57

Anastasia Pentina and Christoph H. Lampert. A pac-bayesian bound for lifelong learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 991–999, 2014. 31

J.-B. Pothin and C. Richard. A greedy algorithm for optimizing the kernel alignment and the performance of kernel machines. In *In Proc. EUSIPCO 06*, pages 4–8, 2006. 53

Peter Prettenhofer and Benno Stein. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1118–1127, 2010. 39

Jose C. Principe. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Springer Publishing Company, Incorporated, 1st edition, 2010. 57

Rajat Raina. *Self-taught Learning*. PhD thesis, Stanford University, 2008. 30

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*

International Conference on Machine Learning, ICML '07, pages 759–766, 2007. 29, 44

Richard G. Ramona, M. and B. David. Multiclass feature selection with kernel gram-matrix-based criteria, 2012. 53

Erendira Rendon, Alejandra Arizmendi Itzel Abundez, and Elvia M. Quiroz. Internal versus external cluster validation indexes. *International Journal of Computers and Communications*, 5, 2011. 51, 116

Marx Z. Kaelbling L. P. Rosenstein, M. T. and T. G. Dietterich. To transfer or not to transfer. *NIPS Workshop on Transfer Learning*, 2005. 49

Ulrich Rückert and Stefan Kramer. Kernel-based inductive transfer. In *ECML/PKDD (2)*, pages 220–233, 2008. 28

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 213–226, 2010. 45

Saburou Saitoh. *Integral Transforms, Reproducing Kernels and their Applications*. Pitman Research Notes in Mathematics Series, 1997. 86

Ruslan Salakhutdinov, Antonio Torralba, and Joshua B. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, pages 1481–1488, 2011a. 31

Ruslan R. Salakhutdinov, Josh Tenenbaum, and Antonio Torralba. Learning to learn with compound hd models. pages 2061–2069. 2011b. 31

Oana Sandu, Giuseppe Carenini, Gabriel Murray, and Raymond Ng. Domain adaptation to summarize human conversations. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, 2010. 39

Sandeepkumar Satpal and Sunita Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, 2007. 37

Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning gaussian process kernels via hierarchical bayes. In *Advances in Neural Information Processing Systems 17*, pages 1209–1216, 2005. 30

D. W. Scott. *Multivariate Density Estimation. Theory, Practice, and Visualization*. Wiley, 1992. 10

Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. 2012. 97

Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge Data Engineering*, 22(7): 929–942, 2010. 38

Le Song. *Learning via Hilbert Space Embedding of Distributions*. PhD thesis, University of Sydney, 2008. 88

Nitish Srivastava and Ruslan Salakhutdinov. Discriminative transfer learning with tree-based priors. In *Advances in Neural Information Processing Systems 26*, pages 2094–2102, 2013. 30

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Von Bnau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *In NIPS*, 2008. 34

Qian Sun, Rita Chattopadhyay, Sethuraman Panchanathan, and Jieping Ye. A two-stage weighting framework for multi-source domain adaptation. In *NIPS*, pages 505–513, 2011. 35

Zoltán Szabó. Information theoretical estimators toolbox. *Journal of Machine Learning Research*, 15:283–287, 2014. 97

Fabian J. Theis, Kurt Stadlthanner, and Toshihisa Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *In Proceedings of the 13th European Signal Processing Conference (EUSIPCO05*, 2005. 107

Markus Thom and Günther Palm. Sparse activity and sparse connectivity in supervised learning. *Journal of Machine Learning Research*, 14(1):1091–1143, 2013a. 108, 110

Markus Thom and Gnther Palm. Efficient sparseness-enforcing projections. *CoRR*, abs/1303.5259, 2013b. 112

Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3081–3088, 2010. 31

Son N. Tran and Artur S. d'Avila Garcez. Adaptive feature ranking for unsupervised transfer learning. *CoRR*, abs/1312.6190, 2013. 41, 42

George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Bjoern Schuller. A deep semi-nmf model for learning hidden representations. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1692–1700, 2014. 108, 115

Jan Van Haaren, Andrey Kolobov, and Jesse Davis. TODTLER: Two-order-deep transfer learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, January 2015. 32

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. 81

Cdric Villani. *Optimal transport : old and new*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009. 83

Hua Wang, Feiping Nie, and Heng Huang. Robust and discriminative self-taught learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 298–306, 2013. 30

Jim Jing-Yan Wang, Yijun Sun, and Halima Bensmail. Domain transfer nonnegative matrix factorization. In *2014 International Joint Conference on Neural Networks*, pages 3605–3612, 2014. 42, 45

Kilian Q. Weinberger, Fei Sha, and Lawrence K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, 2004. 37

Junfeng Wen, Chun-Nam Yu, and Russell Greiner. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 631–639, 2014. 35

Pengcheng Wu and Thomas G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, 2004. 25

Huan Xu and Shie Mannor. Robustness and generalization. In *COLT*, pages 503–515. Omnipress, 2010. 77

Shizhun Yang, Chenping Hou, Changshui Zhang, and Yi Wu. Robust non-negative matrix factorization via joint sparse and graph regularization for transfer learning. *Neural Computing and Applications*, 23(2):541–559, 2013. 42, 43

Zhirong Yang and Erkki Oja. Linear and nonlinear projective non-negative matrix factorization. *Transactions on Neural Networks*, 21(5):734–749, 2010. 65

Yaoliang Yu and Csaba Szepesvri. Analysis of kernel mean matching under covariate shift. In *ICML*, 2012. 34

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*, page 435, 2003. 34

Bianca Zadrozny Zadrozny. Learning and evaluating classifiers under sample selection bias. In *In International Conference on Machine Learning ICML04*, pages 903–910, 2004. 34

Jianwen Zhang and Changshui Zhang. Multitask bregman clustering. *Neurocomputing*, 74(10):1720–1734, 2011. 41, 42

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 819–827, May 2013. 35, 56, 87

Zhou Z.H. Zhang, D. and S. Chen. Non-negative matrix factorization on kernels. *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI'06)*, pages 404–412, 2006. 13

Wenliang Zhong and James Tin-Yau Kwok. Convex multitask learning with flexible task clusters. In *ICML*, 2012. 28

Fuzhen Zhuang, Ping Luo, Changying Du, Qing He, and Zhongzhi Shi. Triplex transfer learning: Exploiting both shared and distinct concepts for text classification. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 425–434, 2013. 42, 44, 64

Hankz Hankui Zhuo and Qiang Yang. Action-model acquisition for planning via transfer learning. *Artificial Intelligence*, 212:80–103, 2014. 32

Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *NIPS'05*, 2005. 67