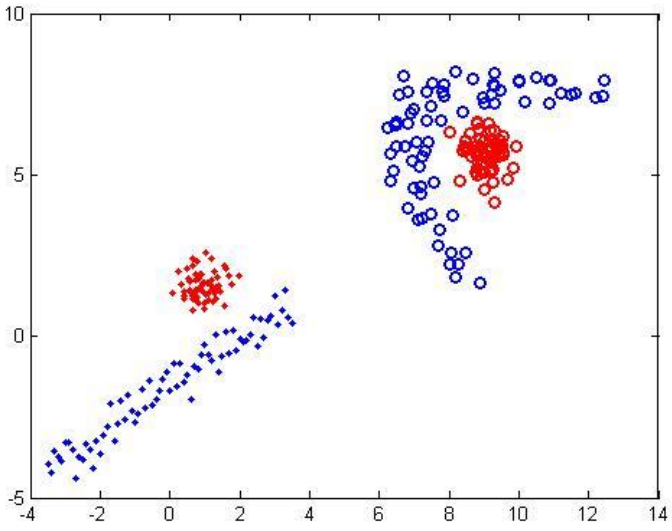
Transfer Learning with Kernel Methods

Xiaoyi Chen

Thèse
de doctorat
de l'UTT

Xiaoyi CHEN

Transfer Learning with Kernel Methods



Spécialité :
Optimisation et Sûreté des Systèmes

2018TROY0005 **Année 2018**

utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Xiaoyi CHEN

le 16 mars 2018

Transfer Learning with Kernel Methods

JURY

M. P. LARZABAL	PROFESSEUR DES UNIVERSITES	Président
M. F. ABDALLAH	PROFESSEUR	Examinateur
M. S. CANU	PROFESSEUR DES UNIVERSITES	Rapporteur
M. P. HONEINE	PROFESSEUR DES UNIVERSITES	Examinateur
M. R. LENGELLÉ	PROFESSEUR DES UNIVERSITES	Directeur de thèse
Mme L. OUKHELLOU	DIRECTEUR DE RECHERCHE IFSTTAR	Rapporteur

Acknowledgments

I would like to express my sincere gratitude to Mr. Régis LENGELLE, my supervisor during my three and half years' doctoral research. He has guided me with his knowledge and experience. I highly valued the cooperation and friendship between us.

I would like thank Mr. and Mrs. Gong who have helped me a lot since I were undergraduate. Thanks to secretaries of LM2S, Bernadette André and Véronique Banse for their availability and amiability, as well as the secretaries of doctoral school, Pascale Denis, Isabelle Leclercq and Thérèse Kazarian.

Thank you to my friends for accompanying me with much joy and my parents who have supported me all the time.

To my parents.

Summary

List of Figures

List of Tables

Introduction

Being a recently emerging machine learning orientation, transfer learning aims at solving learning tasks when there is a dataset shift. For traditional machine learning techniques, training and testing data should be generated from the same distribution while for transfer learning, there is no such obligation. Transfer learning aims to take advantage of relatedness among data and transfer knowledge from easily collected training data (source) to help the learning process of different but related testing data (target). However, using all possible source data is unwise : transferring from the least related or even unrelated source data may lead to a degradation of the performance. In general, a similarity criterion is applied to measure the relatedness and guarantees the quality of transfer. To the best of our knowledge, there are many transfer learning literatures that have successfully extended traditional machine learning methods to adapt to transfer learning context with effective similarity criteria taken into account.

As transfer learning can make use of all related data to help the learning process, it has recognized a wide range of applications. Since 1995, homogeneous transfer learning has attracted much attention : all data share the same feature space ; related application domains include computer vision, natural language processing, biological and medical data processing. Then, heterogeneous transfer learning was proposed in the last decade : source and target can have different features. New applications have also been studied : using text data to help image processing ; cross-language text classification, etc. The transfer learning task covers classification and regression with only labeled source, or labeled source and few labeled target, and clustering with no label information for source and target. Compared to traditional machine learning, a gain in performances is usually obtained. The efficiency of transfer learning is at least comparable as traditional machine learning. On transfer learning tasks, transfer learning approaches can provide both accuracy and efficiency.

Among a variety of transfer learning contexts, we focus on homogeneous transductive transfer learning (covariate shift and domain adaptation). For our specific context, sufficient labeled source data is available while no label information is provided for target data. Our objective is to align source and target data so that source classifier can perform well on target data. As there is no label information in target, considering probabilistically the above homogeneous transductive transfer learning problem may be a good solution and setting constraints on the conditional probabilities (as assumptions) seems to be reasonable.

After a general introduction to transfer learning, homogeneous transfer learning approaches are presented : we start from presenting the most simple homogeneous transfer learning - covariate shift ; then, on relaxing the covariate shift assumption, more generalized domain adaptation approaches are presented.

This thesis is organized as follows :

- In Chapter 1, fundamentals are presented : a brief introduction of statistical learning theory in classification, kernels and some kernel based machine learning approaches. In the latter chapters, this knowledge will be used and some presented approaches will be extended to fit our homogeneous transductive transfer learning context.

- In Chapter 2, a thorough overview of transfer learning is given : both homogeneous transfer learning and heterogeneous transfer learning are presented. More details about transductive transfer learning are given as well as other homologous transfer learning categories. To obtain a general view of transfer learning advances, many well-known transfer learning approaches are introduced by category. Negative transfer learning (transferring unrelated knowledge that degrades the learning performance) is also included in the presentation as well as some literatures against negative transfer. More transfer learning applications are listed in the end of this chapter.

- In Chapter 3, we start solving the homogeneous transductive transfer learning with a strong assumption on conditional probabilities :

$$p_S(y|x, x \in \mathcal{S}) = p_T(y|x, x \in \mathcal{T})$$

  where $\mathcal{S}$ represents the source and $\mathcal{T}$ represents the target ; $y \in \mathcal{Y}$ represents the label of an observation x. With this assumption, covariate shift transfer learning approaches were proposed. However, the performance of these approaches is unsatisfied in some complex simulated and real datasets. Therefore, we propose to relax the covariate shift assumption to become :

$$\exists A, b : p_S(y|x, x \in \mathcal{S}) = p_T(y|Ax + b, x \in \mathcal{T})$$

  with A and b being parameters. Based on this relaxed assumption, a parametric maximum likelihood transductive transfer learning approach is proposed : by maximizing likelihood w.r.t transformation parameters A and b (on $x \in \mathcal{T}$), we maximize the probability of $Ax + b, x \in \mathcal{T}$ under the distribution $p_S(.)$. In other words, we align $p_S(x, x \in \mathcal{S})$ and $p_T(Ax + b, x \in \mathcal{T})$ so that after the transformation A and b, source and target data are very similar.

- However in the previous chapter, the relaxed assumption is still strong. Thus, we further extend the relaxed assumption to become more general :

$$\exists g(.) : \mathbb{R}^m \to \mathbb{R}^n \mid p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$$

  where $g(.)$ is a smooth transformation function (linear or nonlinear). Within this assumption, we aim to align $p_S(x, x \in \mathcal{S})$ and $p_T(g(x), x \in \mathcal{T})$. As $g(.)$ is usually nonlinear, kernel methods are considered.

  In Chapter 4, we propose a transductive transfer learning approach (specifically, domain adaptation) based on this assumption : SVM subject to a Maximum Mean Discrepancy-like constraint (SVMMMD). There, source and target data are projected into a shared RKHS subspace where Maximum Mean Discrepancy (MMD) equals 0 ; therefore, after projection, source and target are expected to be well aligned.

– In Chapter 5, based on the same assumption made in Chapter 4, some KPCA based transductive transfer learning approaches (specifically, domain adaptation) have been proposed. All KPCA based approaches are designed to improve SVMMMD by aligning explicitly the RKHS representations of source and target. For the first approach, we fix the source KPCA subspace and align target to source by selecting the best target KPCA subspace, which minimizes the MMD between the two KPCA representations. The second approach further applies a linear transformation. The third approach linearly transforms target data in the original space for an alignment of KPCA representations of source and target. The similarity measure used is always MMD. We have also compared all of our methods to some state-of-the-art domain adaptation approaches, which shows the effectiveness and efficiency of our approaches.

We finally conclude this thesis by a brief summary of contributions and present perspectives for future works.

Chapter 1

Machine Learning and Kernels

Contents

1.1 Introduction

Since the last century, machine learning has been prosperous during decades. Its objective is to teach computers to learn from existing data and generate the models that allow good prediction capacity. Traditional machine learning includes supervised learning, semi-supervised learning and unsupervised learning depending on the availability of label information. Related data processing tasks include dimension reduction, regression, classification, variable selection, etc in a variety of applications. The whole of this thesis contributes to classification : to find a classifier from training data that can generalize to test data.

By 1990s, machine learning with kernel methods has recognized great success (because of the popularity of SVM). By mapping input data to a higher dimensional kernel space, it is possible to use traditional linear classification approaches to solve nonlinear classification tasks. Examples can be found in Section 1.4.2 (SVM) and Section 1.4.3 (KPCA).

In this chapter, we start by introducing the general framework : a brief review of statistical learning theory (Section 1.2). Then, fundamental knowledge of kernels is presented in Section 1.3. Further developments of kernel methods are presented in Section 1.4. The two latter sections consist of the basics that help to understand the following chapters. Note that we only consider binary classification in this chapter.

1.2 Statistical Learning Theory in Classification

There have been many books, surveys and tutorials on statistical learning theory and machine learning. Here is a short list of literatures : [145], [34], [47], [68], [78], [82], [89], [15], [67], etc.

1.2.1 Statistical Formulation

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ denote the output space ; let x be an observation in $\mathcal{X}$ and its corresponding label is y, $y \in \mathcal{Y}$. Since we consider only binary classification task, $\mathcal{Y} = \{+1, -1\}$. Any pair (x, y) (independent and identically distributed, denoted as iid afterwards) is sampled according to some distribution P. The classification task is to find a decision function $f : \mathcal{X} \to \mathcal{Y}$ that can predict the label of any new iid observation. The observations used for finding f form the training dataset while new iid observations form the testing dataset. The iid condition guarantees that all observations follow the same distribution (P) and they are maximally representative of the underlying distribution. The optimal f should perform well on both training and testing datasets. The performance can be evaluated by some loss function (in Section 1.2.3). But having the best performance on training dataset does not guarantee the optimum : we can obviously find a f that designates every training observation its label and the opposite labels for all the testing data (overfitting, in Section 1.2.4). Meanwhile, we expect f to be *consistent* : as there are more and more training data, the f found is getting closer and closer to the optimal decision function. Thus, there are needs for subtle restrictions on the class of f ($\mathcal{F} \ni f$) : the class of f should be as large as possible so that the optimal f is within the class, while the class should not be too rich (overfitting problems). Then, as some compromise, the solution of the optimal f is found by some trade-off between the loss and *complexity* (measured, for example by VC-dimension in Section 1.2.5). This compromise will be presented in Section

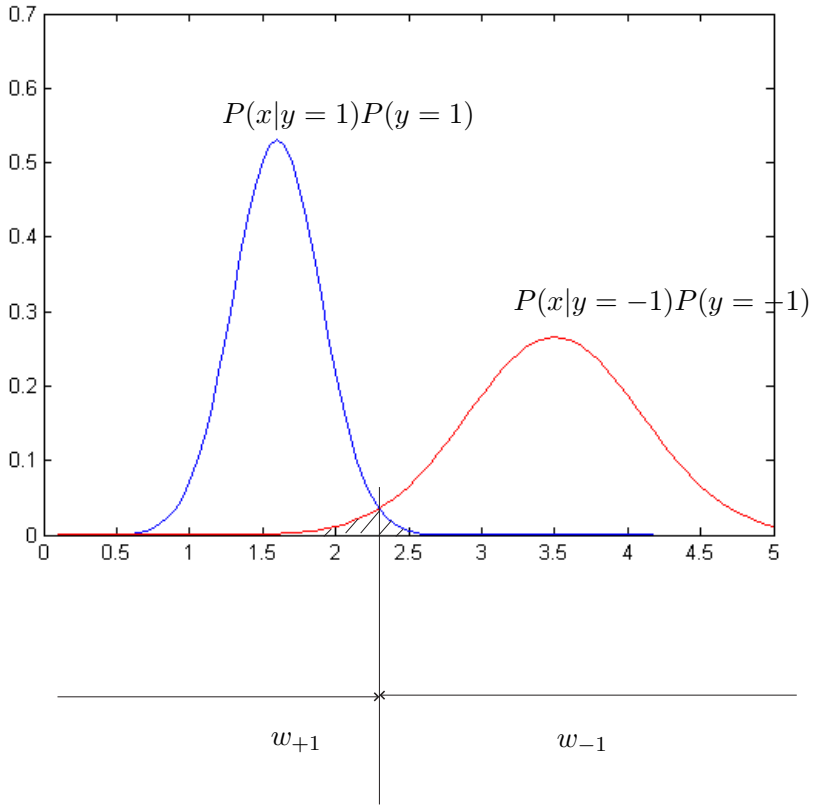


Figure 1.1 – Bayes Error (represented by the shadowed area).

1.2.6 and Section 1.2.7.

1.2.2 Bayes decision rule

If we assume that the a priori and a posteriori probabilities are known, we can get the minimum error for classification task.

Let $P(y|x)$ denote the a posterior probability of label y given observation x ; $P(x)$ the a priori probability ; $P(y)$ the prior probability on label y. Then, we assign the label ($+1$ or -1) to x depending on $P(y|x)$:

$$P(y=1|x) \underset{w_{+1}}{\overset{w_{-1}}{\lessgtr}} P(y=-1|x)$$

where w_{+1} and w_{-1} represent the assigned labels ($+1$ and -1, respectively).

The error occurs when the true label of x is $+1(-1)$ and it is assigned $-1(+1)$:

$$R_{bayeserror} = P(y=1)P(w_{-1}|y=1) + P(y=-1)P(w_{+1}|y=-1)$$

The total minimal error corresponds to the shadowed area in Figure 1.1 and analytically :

$$R^* = \int_{x \in \mathcal{X}} \min\{P(y=1|x), P(y=-1|x)\} P(x) dx$$

The Bayes decision rule achieves the smallest error rate among all the decision rules. However, for most of the time, $P(y|x)$ is not easily accessible, so instead we can first use $P(x|y)$ and then compute $P(y|x)$ by Bayes theorem :

$$P(y=1|x) = \frac{P(y=1)P(x|y=1)}{P(x)}$$
$$P(y=-1|x) = \frac{P(y=-1)P(x|y=-1)}{P(x)}$$

with $P(x) = P(y=1)P(x|y=1) + P(y=-1)P(x|y=-1)$.

Generally, when the true probability density functions and priors are not known, good estimation of distributions is crucial for approximating Bayes decision rule. However, in many applications, there is no guarantee for the quality of such estimation. Other data driven decision rules are proposed that achieve relatively small error rates. In the following sections, we will present another framework of risk minimization, which enables other decision rules (for example, the hyperplane classifier of SVM in Section 1.4.1).

1.2.3 Loss Functions for Binary Classification

We first define the loss function ([113]) :

Definition of loss function : Let $(x, y, f(x)) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ be a triplet with x an observation, y the label of x and $f(x)$ a prediction on x. The map $c : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ with $c(x, y, y) = 0, \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$ is called a loss function.

With $c(x, y, y) = 0, \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$, correct predictions will not lead to any loss and when wrong prediction exists ($y \neq f(x)$), $c > 0$. However, defining a good loss function for a given problem is not always an easy task. Furthermore, a loss function can be difficult to optimize.

In classification, the most intuitive loss function is to count the misclassification error :

$$c(x, y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{otherwise} \end{cases}$$

If x is correctly classified, there is no penalty ; otherwise, we increment the loss every time of misclassification. Some extensions have been made, for example replacing the counting by some functions that are related to x.

Then, instead of predicting only the membership of x, a $f(x)$ that measures the confidence level is further developed. In this case, $\text{sign}(f(x))$ represents the label and $|f(x)|$ represents the confidence level. For binary classification case ($y = +1$ or $y = -1$), an example of such loss functions is soft-margin loss ([8]) :

$$c(x, y, f(x)) = \max(0, 1 - yf(x)) = \begin{cases} 0 & \text{if } yf(x) \geq 1 \\ 1 - yf(x) & \text{otherwise} \end{cases} \quad (1.1)$$

where $yf(x)$ represents the confidence level and 1 represents a margin (which can also be considered as threshold). This soft-margin loss has been widely used in SVM based approaches. Generally, as the joint probability $P(x, y)$ and the prior $P(y)$ are unknown, to express the loss, we refer to the collection of observations $\mathcal{A} = \{(x_i, y_i)\},\ \forall i = 1, 2, \ldots, n : \sum_{i=1}^{n} c(x_i, y_i, f(x_i)) = \sum_{i=1}^{n} \max(0, 1 - y_i f(x_i)) = \sum_{i=1}^{n} \epsilon_i,\ \forall (x_i, y_i) \in \mathcal{A}$ (as in Section 1.4.2).

1.2.4 Expected Generalized Error and Empirical Risk

In the previous section, we have quantified the loss on a single observation x. However, the minimum error (zero error for every observation x) obtained by training dataset may not guarantee the low error rate on unseen testing dataset. Figure 1.2(a) shows an overfitting case of SVM. Obviously, compared to Figure 1.2(c), in Figure 1.2(a), we can build a classifier that fits exactly the data while performs badly on unseen data (testing data). This case is called overfitting. To avoid overfitting, the first issue is how to evaluate properly the error on testing dataset, which enables the classifier to generalize to unseen data.

Theoretically, the test error (called expected generalized error) is defined ([113]) :

$$R[f] = E[c(x, y, f(x))] = \int_{\mathcal{X} \times \mathcal{Y}} c(x, y, f(x)) dP(x, y)$$

where $P(x, y)$ is the joint probability of all iid pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

However, the expected generalized error is generally unmeasurable, because we have no analytical information about $P(x, y)$. As we have training dataset $((x_i, y_i) \in \mathcal{X} \times \mathcal{Y},\ \forall i = 1, \ldots, n)$, we can approximate $R[f]$ using the *empirical density* :

$$p_{emp}(x, y) = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}(x) \delta_{y_i}(y)$$

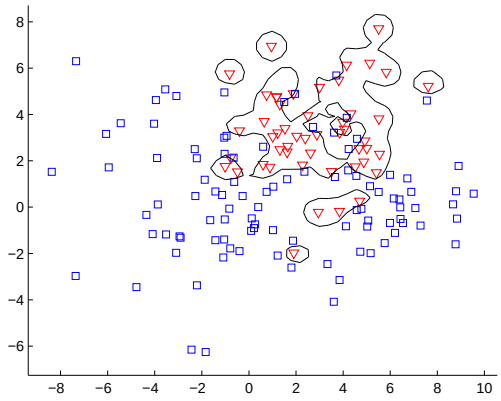
Then, we have the approximation (the empirical risk) :

$$R_{emp}[f] = \int_{\mathcal{X} \times \mathcal{Y}} c(x, y, f(x)) p_{emp} dx dy = \frac{1}{n} \sum_{i=1}^{n} c(x_i, y_i, f(x_i))$$

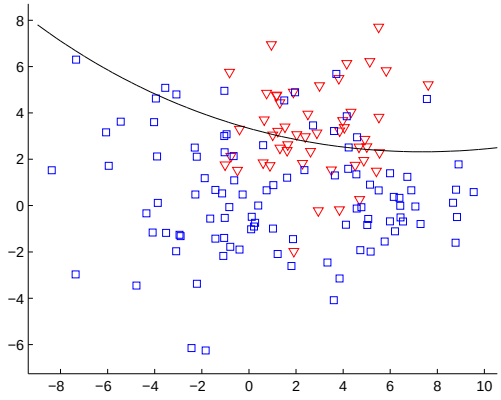
Unfortunately, with empirical risk, there is still no guarantee of a good generalization capacity. Because the minimization of $R_{emp}[f]$ w.r.t f is ill-posed. Thus the second issue that we need to take into account is a suitable class of function ($\mathcal{F} \ni f$). To achieve good performance, $\mathcal{F}$ should be rich enough to include the optimal f, otherwise, there are underfitting cases as shown in Figure 1.2(b) and $\mathcal{F}$ should also be not too rich to avoid overfitting (Figure 1.2(a)). (Compared to Figure 1.2(a) and Figure 1.2(b), an appropriate SVM classification is shown in Figure 1.2(c).)

1.2.5 VC dimension

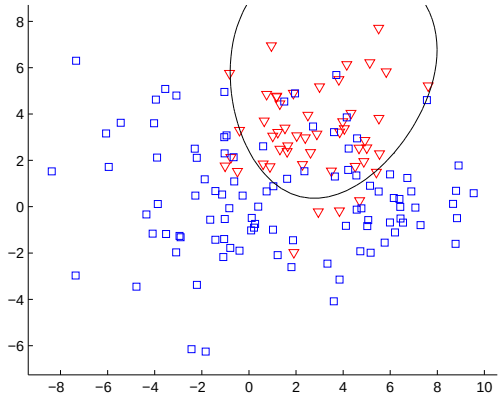
Before limiting the class $\mathcal{F}$, we will first introduce a classification capacity measure (complexity) of $\mathcal{F} \ni f$: the VC dimension.



(a) SVM Overfitting with gaussian kernel $\sigma = 0.4$

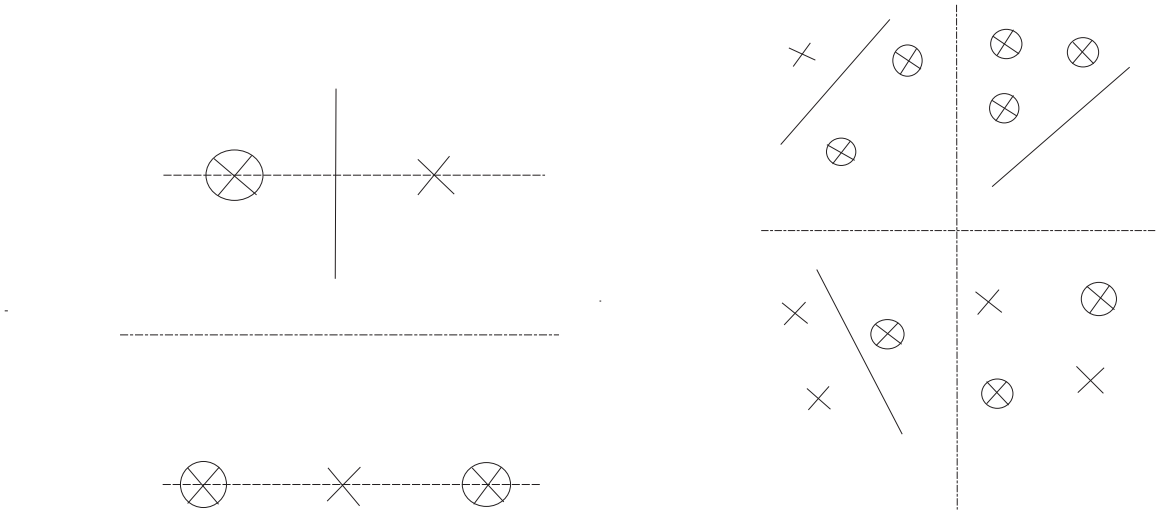


(b) SVM Underfitting with gaussian kernel $\sigma = 50$



(c) Good SVM with gaussian kernel $\sigma = 10$

Figure 1.2 – SVM classification on synthetic data. A good classifier is shown in Figure 1.2(c) while overfitting case is shown in Figure 1.2(a) and underfitting case is shown in Figure 1.2(b).



(a) VC dimension $d_{VC} = 2$ (in $\mathbb{R}$) (b) VC dimension $d_{VC} = 3$ (in $\mathbb{R}^2$)

Figure 1.3 – Illustration of VC dimension of the family of linear detectors. "$\times$" represents a point from a group while "$\otimes$" represents a point from another group. When VC dimension $d_{VC} = 2, 3$ points cannot be shattered (Figure 1.3(a)). Similarly, when VC dimension $d_{VC} = 3, 4$ points cannot be shattered (Figure 1.3(b)).

Definition of VC Dimension ([113]) : in binary classification, since the labels are only $\{+1, -1\}$, there are at most 2^n different possible labeling for n observations ; *a very rich function class might be able to realize all* 2^n *separations, in which case it is said to shatter the* n *points.*. The VC dimension of a class $\mathcal{F}$, is the largest number of n points that $\mathcal{F}$ can shatter. If there is no such n, the VC dimension is defined to be ∞.

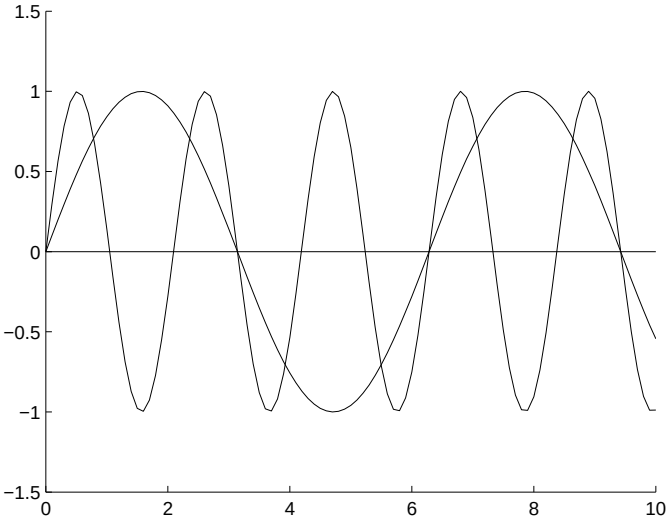


Figure 1.4 – VC dimension is ∞ : $y = \text{sign}(\sin(xt))$, where t is the parameter and $x \in \mathbb{R}$. The figure shows two possible curves of $y = \sin(xt)$.

As shown in Figure 1.3(a), when $\mathcal{F} \subseteq \mathbb{R}^d$ and $d = 1$, any linear detector ($f \in \mathcal{F}$) can shatter 2 points but not the set of 3 points ($d_{VC} = 2$) ; in Figure 1.3(b), when $d = 2$, we can shatter 3 points but not the set of 4 points ($d_{VC} = 3$). In this case, VC dimension is $d + 1$.

Figure 1.4 is an example for infinite VC dimension. By different value of t, $\mathcal{F}$ can shatter an infinity of points ($x \in \mathbb{R}$).

1.2.6 Structural Risk

Intuitively, we want VC dimension to be high so that the classifier is capable of shattering a large number of points. However, constructing a too complicated decision rule can lead to overfitting. It is the same with a too rich class of function f ($\mathcal{F}$ in Section 1.2.4). Structural Risk is the sum of two antagonistic terms proposed to find a compromise between classification capacity and the empirical risk :

$$R_{struc}[f] = R_{emp}[f] + r(capacity, size)$$

where $r(capacity, size)$ is a penalty term w.r.t the size of the training set and the capacity defined by VC dimension. We suppose that the number of training data points ($size$) is fixed here.

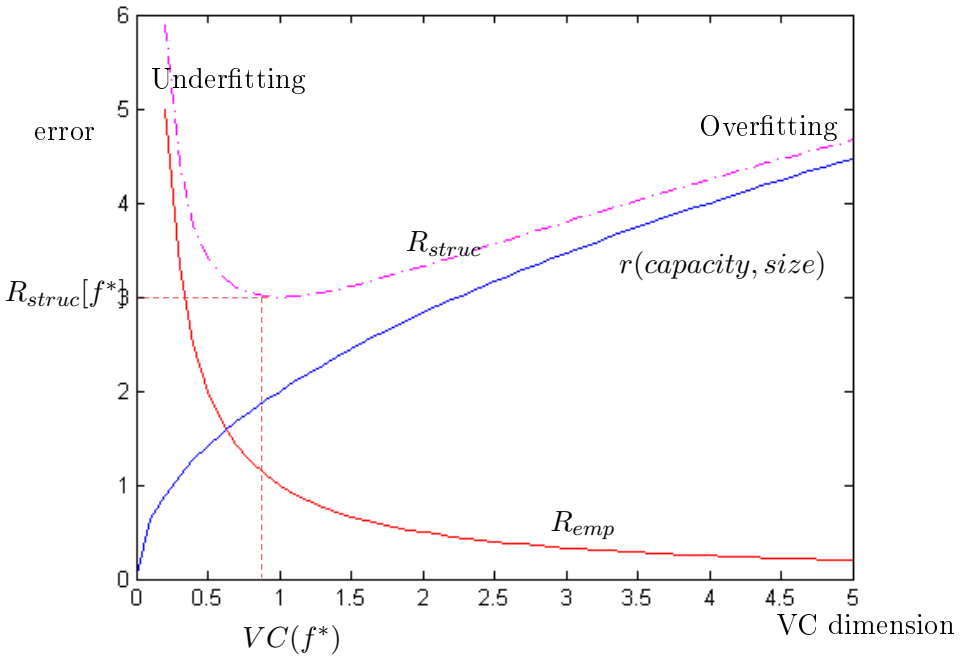
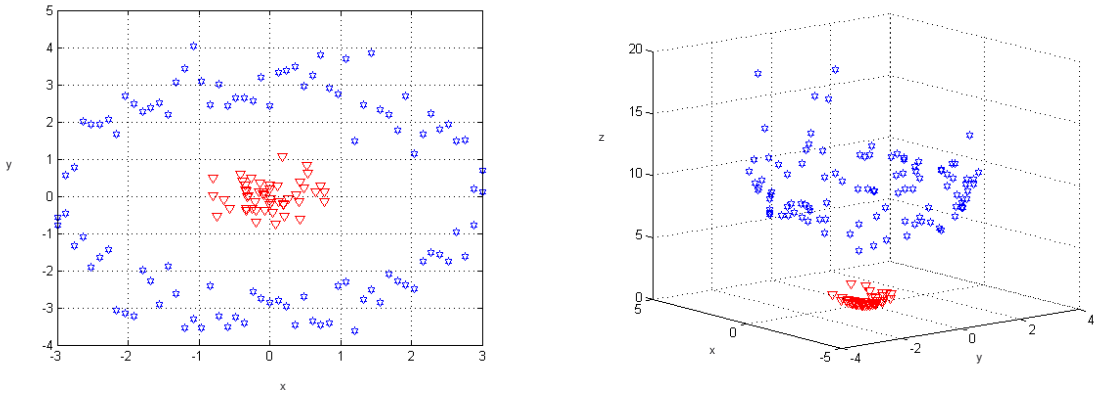


Figure 1.5 – Structural Risk Minimization

Usually, the choice of f^* is given by finding the minimum of $R_{struc}[f]$, which makes a compromise between the empirical risk (R_{emp}) and the complexity of f (measured here by VC dimension). Figure 1.5 is a good illustration of minimizing structural risk.



(a) Data in Original Space (concentric circles).

(b) Data in Higher Dimensional Space.

Figure 1.6 – Same data in different spaces. Figure 1.6(a) represents concentric circles in the original space ($\mathbb{R}^2$). Figure 1.6(b) represents the same data in higher dimensional space ($\mathbb{R}^3$).

1.2.7 Regularization

Other approaches have also been proposed to find the best decision function f within the suitable $\mathcal{F}$. Generally, $\mathcal{F}$ is limited to a compact set based on the solution of ill-posed problems ([138], [107]). Then, we add a regularization term to the empirical risk minimization task :

$$\min R_{reg}[f] = \min R_{emp}[f] + \lambda reg(f)$$

where $reg(f)$ represents the regularization term and λ is the regularization parameter that controls the trade-off between $R_{emp}[f]$ and $reg(f)$.

The regularization term can serve as enforcing the simplicity of f or the smoothness of f, etc. For example, in coding language, $reg(f)$ can be chosen as the length of description of a model ; in classical statistics, the free parameter [1] can be chosen as $reg(f)$ (as complexity). Compared to structural risk minimization, regularization is more easily implemented and authors can design specific regularization term depending on the special task. However, tuning intelligently the regularization parameter λ is still under research.

1.3 Kernel Fundamentals

Before introducing kernel methods in machine learning, the fundamentals of kernels are presented in this section.

1.3.1 Positive Definite Kernels

We start by illustrating the principle of kernels in Figure 1.6.

1. A free parameter is a parameter that can be adjusted to make the model fit the data.

In Figure 1.6, we map data of $\mathbb{R}^2$ to be in a space of $\mathbb{R}^3$:

$$\mathbb{R}^2 \rightarrow \mathbb{R}^3$$
$$(x, y) \rightarrow (x, y, z)$$

with $z = x^2 + y^2$. From the figure, we can observe that nonlinearly separable original data can be separated linearly in higher dimensional spaces. However, the transformation from $\mathbb{R}^2$ to $\mathbb{R}^3$ is not unique. To take advantage of nonlinear aspect of transformations to higher dimensional space, *kernel* is proposed for any such possible transformation ($\phi(.)$).

Definition of Kernels

Let $\phi(.)$ denote the transformation from original space to a higher dimensional space (which is also called *feature mapping*) :

$$\phi : \mathcal{X} \rightarrow \mathcal{H}$$
$$x \rightarrow \phi(x)$$

where $\mathcal{H}$ represents the space after transformation ϕ.

Then, we can define the kernel (k) as a measure of the similarity between any two observations by a inner product between their corresponding $\phi(x)$ and $\phi(x')$:

$$k(x, x') = < \phi(x), \phi(x') >_{\mathcal{H}}$$

and replacing the inner product $< \phi(x), \phi(x') >_{\mathcal{H}}$ by $k(x, x')$ is called the *kernel trick*.

There are several properties of kernels.

Positive Definiteness

Positive Definite Matrix : let M denote a $n \times n$ real matrix, for any $c_i \in \mathbb{R}$,

$$\sum_{i,j} c_i c_j M(i, j) \geq 0, \ \forall i, j = 1, \ldots, n$$

where $M(i, j)$ represents the element (i, j) of matrix M ; then M is called positive definite.

Gram Matrix ([113]) : *Given a function $k : \mathcal{X}^2 \rightarrow \mathbb{R}$ and observations $x_i \in \mathcal{X}, \forall i = 1, \ldots, n$, the $n \times n$ matrix K with elements $K(i, j) = k(x_i, x_j)$ is called Gram Matrix of kernel k with respect to x_i ($x_i \in \mathcal{X}, \forall i = 1, \ldots, n$).*

Definition of Positive Definite Kernel : The kernel that always induces a positive definite Gram Matrix, is called positive definite kernel.

1.3.2 Other Properties of (Positive Definite) Kernels

Cauchy-Schwarz Inequality for Kernels ([113]) : Given k a positive definite kernel, and $x_i, x_j \in \mathcal{X}, \forall i, j = 1, \ldots, n$, then,

$$|k(x_i, x_j)|^2 \leq k(x_i, x_i) k(x_j, x_j)$$

Proof This can be easily shown by computing the determinant of the Gram matrix associated to elements x_i and x_j.

Other properties : we present here the simple manipulations on positive definite kernels that do not change the kernel nature.

- Let $\mathbf{k}$ be a positive definite kernel, for any constant $\lambda > 0$, $\lambda\mathbf{k}$ is always a positive definite kernel.
- Let $\mathbf{k_1}$ and $\mathbf{k_2}$ be two positive definite kernels, then $\mathbf{g}(x, x') = \mathbf{k_1}(x, x') + \mathbf{k_2}(x, x')$, where $x, x' \in \mathcal{X}$ is always a positive definite kernel.
- Let $\mathbf{k}$ be a positive definite kernel, for any integer p, $\mathbf{k}^p$ is always a positive definite kernel.
- Let $\mathbf{k}$ be a positive definite kernel, $exp(\mathbf{k})$ is always a positive definite kernel.
- Let $\mathbf{k}$ be a positive definite kernel, $f : \mathcal{X} \to \mathbb{R}$, $\mathbf{g}(x, x') = f(x)\mathbf{k}(x, x')f(x')$ is always a positive definite kernel.

Here after, we simply use kernel for any positive definite kernel.

1.3.3 Reproducing Kernel Hilbert Space (RKHS)

Definition ([113])

Given $\mathcal{X}$ a nonempty set and $\mathcal{H}$ a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$. Then, $\mathcal{H}$ is called Reproducing Kernel Hilbert Space with the dot product $< ., . >$ (and the norm $||f|| = \sqrt{< f, f >}$), if there exists a function $k : \mathcal{X} \times \mathcal{X} \to R$ with the following properties :

- *Reproducing Property :*

$$< f, k(x, .) >= f(x), \ \forall f \in \mathcal{H}$$

in particular,

$$< k(x, .), k(x', .) >= k(x, x')$$

- *Closed Space Property : k spans $\mathcal{H}$*

Such k is called reproducing kernel.

Theorem of Uniqueness ([117])

For a RKHS, k is the unique reproducing kernel if it exists.

Proof We suppose that there are two reproducing kernels for a RKHS, namely k and k'. With the reproducing properties :

$$< k(x, .), k'(x', .) >_{\mathcal{H}} = k(x, x') = k'(x, x')$$

Thus, there is a unique k associated to a RKHS.

Theorem of Existence ([117])

$\mathcal{H}$ is a RKHS if and only if it has a reproducing kernel.

Collary ([136])

Every reproducing kernel is a kernel.

Proof We can find a feature map ϕ such that :

$$\begin{aligned}\phi &: \mathcal{X} \to \mathcal{H} \\ \phi(x) &= k(x, .)\end{aligned}$$

where k is a reproducing kernel. Then, $k(x, x') =< k(x, .), k(x', .) >_{\mathcal{H}}$. Thus every reproducing kernel is a kernel.

Moore–Aronszajn Theorem ([9], Theorem 3)

Given any kernel function k on $\mathcal{X}$, then there is a unique RKHS associated.

However, for the same k, the feature representation (also called feature map) is not unique. For example (from [116]), if we suppose $\mathcal{X} \in \mathbb{R}^2$ and $k(x, y) =< x, y >^2$, we have :

$$\begin{aligned}k(x, y) &= x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 x_2 y_2 \\ &= [x_1^2 \quad x_2^2 \quad \sqrt{2}x_1 x_2] \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2}y_1 y_2 \end{bmatrix} \\ &= [x_1^2 \quad x_2^2 \quad x_1 x_2 \quad x_1 x_2] \begin{bmatrix} y_1^2 \\ y_2^2 \\ y_1 y_2 \\ y_1 y_2 \end{bmatrix}\end{aligned}$$

Therefore, we can define two different feature mappings $\phi_1(x) = [x_1^2 \; x_2^2 \; \sqrt{2}x_1 x_2]$ and $\phi_2(x) = [x_1^2 \; x_2^2 \; x_1 x_2 \; x_1 x_2]$, both of which are valid feature representations associated to $k(x, y) =< x, y >^2$. Their feature space $\mathcal{H}_1 = \mathbb{R}^3$ for $\phi_1(.)$ and $\mathcal{H}_2 = \mathbb{R}^4$ for $\phi_2(.)$ are different.

1.3.4 Mercer's Theorem

In this section, we present the Mercer's Theorem, with the help of which a RKHS can be defined.

Mercer's Theorem Given a continuous kernel k, we can define a linear operator $[T_k g](x)$:

$$[T_k g](x) = \int_{\mathcal{X}} k(x, x') g(x') dx'$$

Then, we have

$$k(x, x') = \sum_{j=1}^{\infty} \lambda_j \psi_j(x) \psi_j(x')$$

where $\psi_j(.)$ is an eigenfunction of T_k and λ_j ($\lambda_j > 0, \; \forall j$) is the corresponding eigenvalue.

With Mercer's Theorem satisfied, a feature map can be defined ([113]) :

$$\phi : x \to \sqrt{\lambda_j}\psi_j(x), \quad \forall j = 1, \ldots, N_{\mathcal{H}}$$

The inner product $< \phi(x), \phi(x') >$ can approach $k(x, x')$ within some accuracy. Furthermore, if $\mathcal{X}$ is a compact set and k a continuous function, then the finite dimensional feature space induced by $\phi(.)$ is a RKHS (such k is called Mercer Kernel).

1.3.5 Examples of Kernels

In this section, we list some commonly used kernels :

Type of Kernels	Expressions
Polynomial Kernels	$k(x, x') =< x, x' >^d$
Inhomogeneous Polynomial Kernels	$k(x, x') = (< x, x' > +c)^d, c \geq 0$
Gaussian Kernels	$k(x, x') = \exp(-\frac{\|\|x-x'\|\|^2}{2\sigma^2}), \sigma > 0$
Sigmoid Kernels	$k(x, x') = \tanh(\theta < x, x' > +\upsilon), \theta > 0, \upsilon \geq 0$

Table 1.1 – Commonly used Kernels

More details about specific kernels can be found in [1], [83], [111],etc.

1.3.6 Representer Theorem

Proposed in 1971 by Kimeldorf and Wahba ([65]), the Representer Theorem represents the optimal solution of any regularized minimization task :

Representer Theorem : *Given $\Gamma : [0, \infty) \to R$ a strictly monotonic increasing function ; a set $\mathcal{X}$; a kernel k ; any loss function c ; then, the minimizer f^* in RKHS ($\mathcal{H}$) of the objective function :*

$$\frac{1}{n}\sum_{i=1}^{n} c(x_i, y_i, f(x_i)) + \Gamma(||f||_{\mathcal{H}})$$

admits a representation of the form :

$$f^*(.) = \sum_{i=1}^{n} \alpha_i k(x_i, .)$$

Proof :
Let $f(x) = f^*(x) + f^*_\perp(x)$, then,

$$\begin{aligned} f(x_j) =< f(.), k(x_j, .) > &=< f^*(.) + f^*_\perp(.), k(x_j, .) > \\ &= \sum_{i=1}^{n} \alpha_i k(x_i, x_j) + < f^*_\perp(.), k(x_j, .) >_{\mathcal{H}} \\ &= \sum_{i=1}^{n} \alpha_i k(x_i, x_j) \\ &= f^*(x_j) \end{aligned}$$

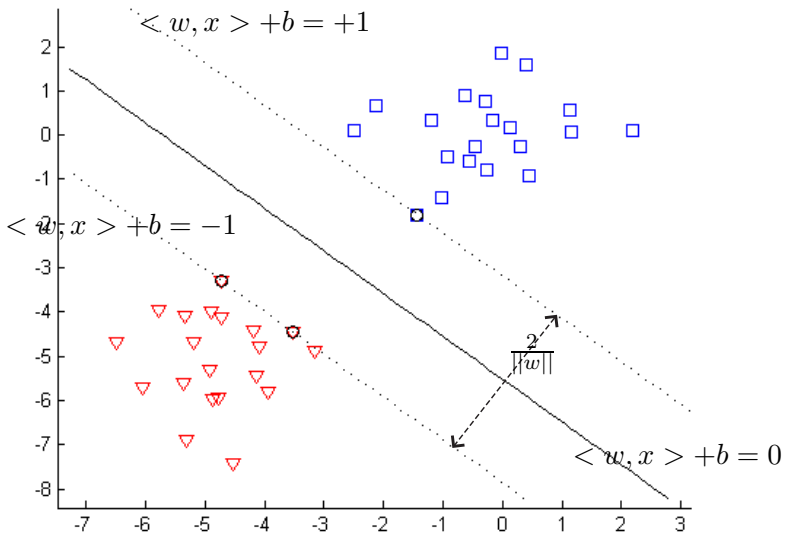


Figure 1.7 – Illustration of a hard margin SVM. The red triangles represent the negative class and the blue squares represent the positive class. Dotted lines are margins and the classifier is the solid line.

Therefore, $\frac{1}{n}\sum_{i=1}^{n} c(x_i, y_i, f(x_i)) = \frac{1}{n}\sum_{i=1}^{n} c(x_i, y_i, f^*(x_i))$.

For the second term of objective function :

$$\Gamma(||f||_{\mathcal{H}}) = \bar{\Gamma}(||f^*||^2_{\mathcal{H}} + ||f^*_{\perp}||^2_{\mathcal{H}}) \geq \bar{\Gamma}(||f^*||^2_{\mathcal{H}})$$

The equality holds when $||f^*_{\perp}||^2_{\mathcal{H}} = 0$ and the minimizer should be f^*.

This theorem has been used in many kernel based applications, for example SVM, KPCA (presented in Section 1.4).

1.4 Machine Learning with Kernels

1.4.1 Hard Margin Support Vector Machines (SVM)

SVM is a supervised classification approach that separates data with maximal margin :

$$\begin{aligned} &\min \frac{1}{2}||w||^2 \\ &\text{s.t. } \; y_i(< w, x_i > +b) \geq 1, \quad \forall i = 1, \ldots, n \end{aligned} \tag{1.2}$$

where w is parameter that influences the margin ; y_i is the label of an observation x_i ; b is the bias. In Figure 1.7, we illustrate SVM classification with the geometrical interpretation of every element. The final decision function is $f = \text{sign}(< w, x > +b)$ (for binary classification). Multi-class SVM can be found in [152].

Actually, theoretical arguments show that SVM optimal hyperplane possesses good generalization capacity :

Theorem (from [144]) : Given a hyperplane defined by $< w, x >= 0$ and w is normalized with respect to $\mathcal{X} \ni x : \min_{x \in \mathcal{X}} | < w, x > | = 1$. If all decision functions defined by $f = \text{sign}(< w, x >), (x \in \mathcal{X})$ satisfy the condition that $||w|| \leq \Lambda$, then their VC dimension h is upper bounded by :

$$h \leq R^2 \Lambda^2 \tag{1.3}$$

where R represents the radius of the smallest hyperplane centered on the origin containing all $x \in \mathcal{X}$. When the margin is maximal, $||w||$ is minimal then the upper bound on VC dimension in Eq. 1.3 is as tight as possible. Thus, the maximal margin induces good generalization performance.

1.4.2 Soft Margin Support Vector Machines (SVM)

However, the SVM problem 1.2 allows no classification error. To allow overlaps between clusters, the soft-margin SVM is proposed :

$$\begin{aligned} &\min \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \epsilon_i \\ &\text{s.t.} \quad \epsilon_i \geq 0, \quad \forall i = 1, \ldots, n \\ &\qquad y_i(< w, x_i > + b) \geq 1 - \epsilon_i, \quad \forall i = 1, \ldots, n \end{aligned} \tag{1.4}$$

where $\epsilon_i \geq 0$ is the possible error and C is the trade-off parameter between the maximum margin and the error. In this formulation, $\sum_{i=1}^{n} \epsilon_i$ can be considered as a loss function (soft-margin loss in Eq. 1.1) and it is regularized by the margin parameter $\frac{1}{2}||w||^2$. An illustration is given in Figure 1.8.

One limitation for both hard-margin and soft-margin SVM is that they only perform well on data for which the best classifier is linear. Therefore, the kernel technique has been introduced to SVM so that nonlinear SVM classifier is simply realized : we first transform the original input data x by a nonlinear kernel mapping $\phi(.)$; then, we use the standard SVM and obtain :

$$\begin{aligned} &\min \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \epsilon_i \\ &\text{s.t.} \quad \epsilon_i \geq 0, \quad \forall i = 1, \ldots, n \\ &\qquad y_i(< w, \phi(x_i) > + b) \geq 1 - \epsilon_i, \quad \forall i = 1, \ldots, n \end{aligned} \tag{1.5}$$

We then solve the above problem by Lagrangian method :

$$\mathcal{L} = \max_{\alpha, \beta} \min_{w, b, \epsilon} \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \epsilon_i - \sum_{i=1}^{n} \beta_i \epsilon_i - \sum_{i=1}^{n} \alpha_i [y_i(< w, \phi(x_i) > + b) + \epsilon_i - 1]$$

where α and β are Lagrangian multipliers and they are non-negative.

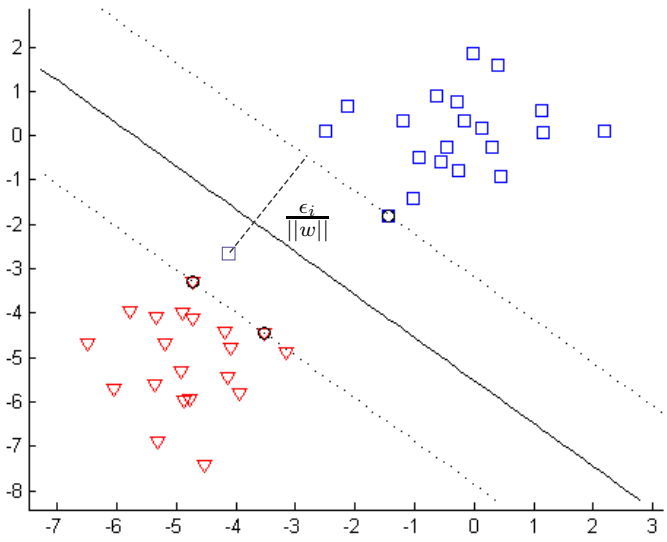


Figure 1.8 – An illustration of soft-margin SVM. The colors designate the classes $+1$ or -1. There is a blue square that is wrongly classified as the class of red triangles.

Then, we can first minimize $\mathcal{L}$ w.r.t w, b, ϵ :

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w} &= w - \sum_{i=1}^{n} \alpha_i y_i \phi(x_i) = 0 \\
\frac{\partial \mathcal{L}}{\partial b} &= \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\partial \mathcal{L}}{\partial \epsilon_i} &= C - \beta_i - \alpha_i = 0, \ \forall i = 1, \dots, n
\end{aligned}$$

Therefore,

$$\begin{aligned}
w &= \sum_{i=1}^{n} \alpha_i y_i \phi(x_i) \\
\beta_i &= C - \alpha_i, \ \forall i = 1, \dots, n \\
\sum_{i=1}^{n} \alpha_i y_i &= 0
\end{aligned} \tag{1.6}$$

we replace w and β_i and obtain :

$$\max_{\alpha} -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i y_i \alpha_j y_j < \phi(x_i), \phi(x_j) > + \sum_{i=1}^{n} \alpha_i$$
$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C$$

which is equivalent to :

$$\begin{aligned} \min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i y_i \alpha_j y_j < \phi(x_i), \phi(x_j) > - \sum_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C \end{aligned} \tag{1.7}$$

Using the kernel trick, $< \phi(x_i), \phi(x_j) >$ can be replaced by $k(x_i, x_j)$.

The final dual form (problem 1.7) is a convex quadratic problem w.r.t α and the feasible domain is also convex. Convexity guarantees uniqueness of the solution. Then, standard quadratic optimization tools can be used to solve SVM. In general, most of the $\alpha_i, i = 1, \ldots, n$ are zeros. Geometrically, most of the data are located outside the margin and are correctly classified. The data within the margin or wrongly classified are called Support Vectors (their corresponding α_i are strictly positive). The optimal hyperplane is only determined by support vectors (Eq 1.6). Thus, there are fast optimization algorithms proposed.

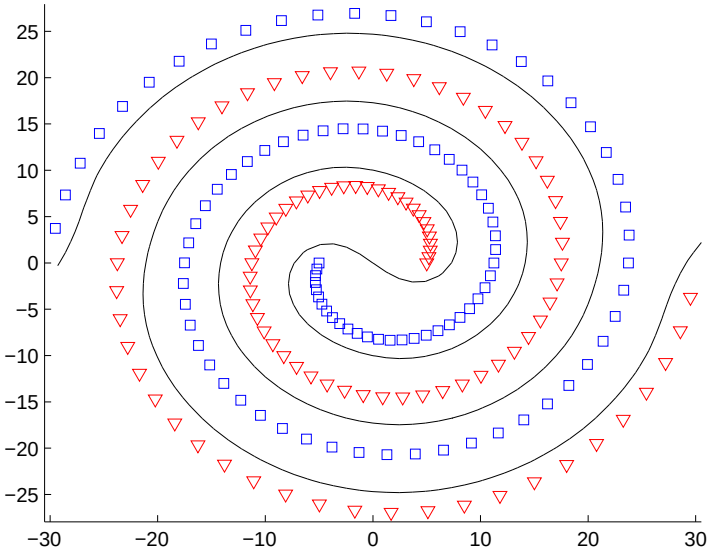


Figure 1.9 – SVM on spiral data (nonlinear case). Red triangles and blue squares are from two different classes.

Then, the prediction on a new observation z is :

$$y_z = \text{sign}(< w, \phi(z) > + b) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i < \phi(x_i), z > + b) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i k(x_i, z) + b)$$

The bias b is found by using KKT conditions ([16]). A SVM on nonlinearly separable data is shown in Figure 1.9.

1.4.3 Kernel Principal Component Analysis (KPCA)

PCA

Before introducing KPCA, we first review the fundamentals of Principal Component Analysis (PCA). PCA is an unsupervised statistical procedure that linearly transforms data to a new coordinate system so that the greatest variance lies on the first coordinate, the second greatest variance on the second coordinate and so on. A principal component designates each coordinate.

Let X be the matrix $(n \times m)$ of observations, the objective is to find the vector u so that Xu extract the most important part of data information, here measured by the variance of Xu. We begin by determining the first principal component defined by the vector u that captures the largest variance of Xu. We first center the data : $x_{c(j)} = x_j - \frac{1}{n}\sum_{i=1}^{n} x_i, x_i \in X, \forall i = 1, \ldots, n$ and let X_c denote the matrix of centered elements $(x_{c(j)} \in X_c,\ \forall j = 1, \ldots, n)$; then, the problem is defined as follows :

$$\arg\max_u ||X_c u||^2 = \arg\max_u u^T X_c^T X_c u \quad \text{s.t. } u^T u = 1 \tag{1.8}$$

The constraint $u^T u = 1$ guarantees that the optimization problem is not ill-posed. Introducing the Lagrangian, we have :

$$\mathcal{L} = u^T X_c^T X_c u - \lambda(u^T u - 1)$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial u} = 2X_c^T X_c u - 2\lambda u = 0 \Leftrightarrow X_c^T X_c u = \lambda u$$

From the above solution, we can find that the Lagrange parameter λ can be considered as an eigenvalue of $X_c^T X_c$ and the the first principal component u is the eigenvector that corresponds to the largest eigenvalue λ. With a similar reasoning, we understand that the l^{th} principal component is the eigenvector that corresponds to the l^{th} largest eigenvalue of the covariance matrix $X_c^T X_c$.

KPCA

KPCA is a kernelized version of PCA that extracts the principal components in a RKHS. After transformation of the data into the RKHS, the covariance matrix of the data becomes : $C_K = \frac{1}{n}\sum_{i=1}^{n} \phi_c(x_i)\phi_c(x_i)^T$, where $\phi_c(x_i) = \phi(x_i) - \frac{1}{n}\sum_{j=1}^{n} \phi(x_j)$; $\phi(X)$ is the kernel transformation of original data X ; $\phi_c(x_i), i = 1, ..., n$ centers all the data in the high-dimensional space spanned by $\phi(.)$, corresponding to X_c in (1.8). The formulation for KPCA is :

$$\arg\max_V V C_K V^T \quad \text{s.t. } V^T V = I \tag{1.9}$$

As KPCA manipulates data in the same RKHS, eigenvectors (V) should be spanned by $\phi_c(x_i)$. So we have $V = \sum_{i=1}^{n} \alpha_i \phi_c(x_i)$. Then,

$$\tilde{\lambda} V = C_K V \Leftrightarrow n\tilde{\lambda}\alpha = M\alpha$$

where $M = (I_n - \frac{1}{n}1_n 1_n^T)K(I_n - \frac{1}{n}1_n 1_n^T), K = < \phi(X), \phi(X) >_{\mathcal{H}}$, I_n is the identity matrix of size $n \times n$ and 1_n is the $n \times 1$ column vector with all elements equaling 1. As in problem 1.9, $V^T V = I$. Therefore, $\alpha^T M \alpha = I$, which is equivalent to $n\tilde{\lambda}\alpha^T\alpha = I$, with α being the eigenvectors of M and $\tilde{\lambda}$ being eigenvalues

Then, expressing the coordinates ($V\phi_c(Z)$) of any data set Z after KPCA, we obtain :

$$V\phi_c(Z) = \alpha^T K_c \text{ where } K_c(i,j) =< \phi_c(x_i), \phi_c(z_j) >_{\mathcal{H}}$$

and $\phi_c(Z)$ can be presented in the new coordinate system (axes are $\overline{V}$) by :

$$\phi_c(Z) = \alpha^T K_c \overline{V}$$

Like PCA, for KPCA, a small number of principal components is generally sufficient to retain the main structure of the data. For the whole thesis, we generally use 2 or 3 principal components.

One disadvantage of KPCA is its inefficiency when the number of observations augments, because of the computation of K_c.

1.4.4 Maximum Mean Discrepancy (MMD)

Introduced in [45], Maximum Mean Discrepancy (MMD) is a non-parametric *distance* between two probability distributions. It measures the maximum distance between the expected values of these distributions (any distribution p and any distribution q) w.r.t any transformation ($f : x \to f(x)$, where x is a random variable drawn from the distribution)

$$MMD[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} (\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(y)])$$

In [41], from the Theorem on MMD, we can conclude that distributions p and q are equal iff $MMD = 0$. Then, using kernel embedding of distributions, Smola [127] and Gretton et al. [55] have shown that MMD can be easily evaluated in a RKHS. Accordingly, MMD can be expressed as

$$MMD = ||\mu_p - \mu_q||_{\mathcal{H}} \tag{1.10}$$

where $\mathcal{H}$ represents a RKHS, $\mu_{\{p,q\}}$ stands for $\mathbf{E}_{\{p,q\}}[k(x,.)]$ and $k(x,.)$ is the representation of x in the RKHS, which is equivalent to any transformation function f because of the nature of kernel. But the kernel must be *universal*.

Definition of Universal Kernels : Given a compact set $\mathcal{X}$, a RKHS kernel k and a finite Borel measure μ, if the kernel embedding :

$$\mu \to \int_{\mathcal{X}} k(.,x) d\mu(x)$$

is injective, then k is called universal kernel. (For more details see [129] and [130]). Gaussian kernels are universal ([84]).

In other words, given a universal kernel, the kernel mean embedding $\mu : p \to \mu_p$ (where p is any distribution) is injective ([127]). Thus, the MMD can be kernelized as in Eq 1.10.

The kernelization of squared MMD is shown as follows :

$$\begin{aligned} MMD^2[\mathcal{F}, p, q] &= [\sup_{||f||_{\mathcal{H}} \leq 1} (\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(y)])]^2 \\ &= [\sup_{||f||_{\mathcal{H}} \leq 1} (\mathbf{E}_p[< \phi(x), f >_{\mathcal{H}}] - \mathbf{E}_q[< \phi(y), f >_{\mathcal{H}}])]^2 \\ &= [\sup_{||f||_{\mathcal{H}} \leq 1} < \mu_p - \mu_q, f >_{\mathcal{H}}]^2 \\ &= ||\mu_p - \mu_q||^2_{\mathcal{H}} \end{aligned}$$

Then, the theorem proposed by Dudley is extended to :

Theorem (Steinwart [130] and Smola [128]) : $MMD[\mathcal{F}, p, q] = 0$ iff $p = q$ when $\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \leq 1\}$ provided that $\mathcal{H}$ is universal.

Then using the kernel trick, the squared MMD can be further developed as follows :

$$\begin{aligned} MMD^2[\mathcal{F}, p, q] &= \|\mu_p - \mu_q\|^2_{\mathcal{H}} \\ &= \mathbf{E}_{p,p}[k(x, x')] - 2\mathbf{E}_{p,q}[k(x, y)] + \mathbf{E}_{q,q}[k(y, y')] \end{aligned}$$

Here, x and x' are independent observations drawn from distribution p, y and y' are independent observations from distribution q, k designates a universal kernel function.

To make kernelized MMD calculable, an unbiased estimation for squared MMD is proposed in [118] :

$$\begin{aligned} \widehat{MMD}^2_u[\mathcal{F}, X, Y] &= \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) \\ &+ \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) - \frac{2}{nm} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j) \end{aligned} \tag{1.11}$$

where $x_i, i = 1, \ldots, m$ and $y_i, i = 1, \ldots, n$ are iid examples drawn from p and q respectively.

1.4.5 Kernel Density Estimation (KDE)

Kernel Density Estimation (KDE) is a non-parametric density estimation approach. We first review the fundamentals of non-parametric density estimation to show how such density estimation works.

In a probabilistic aspect, given an observation x, drawn from distribution $p(x)$, it will fall in a region $\mathcal{X}$ with probability :

$$P = \int_{\mathcal{X}} p(x')dx'$$

Now, given n observations, the probability of k iid observations (out of n) fall in the region $\mathcal{X}$ is :

$$P(k) = \binom{n}{k} P^k (1 - P)^{n-k}$$

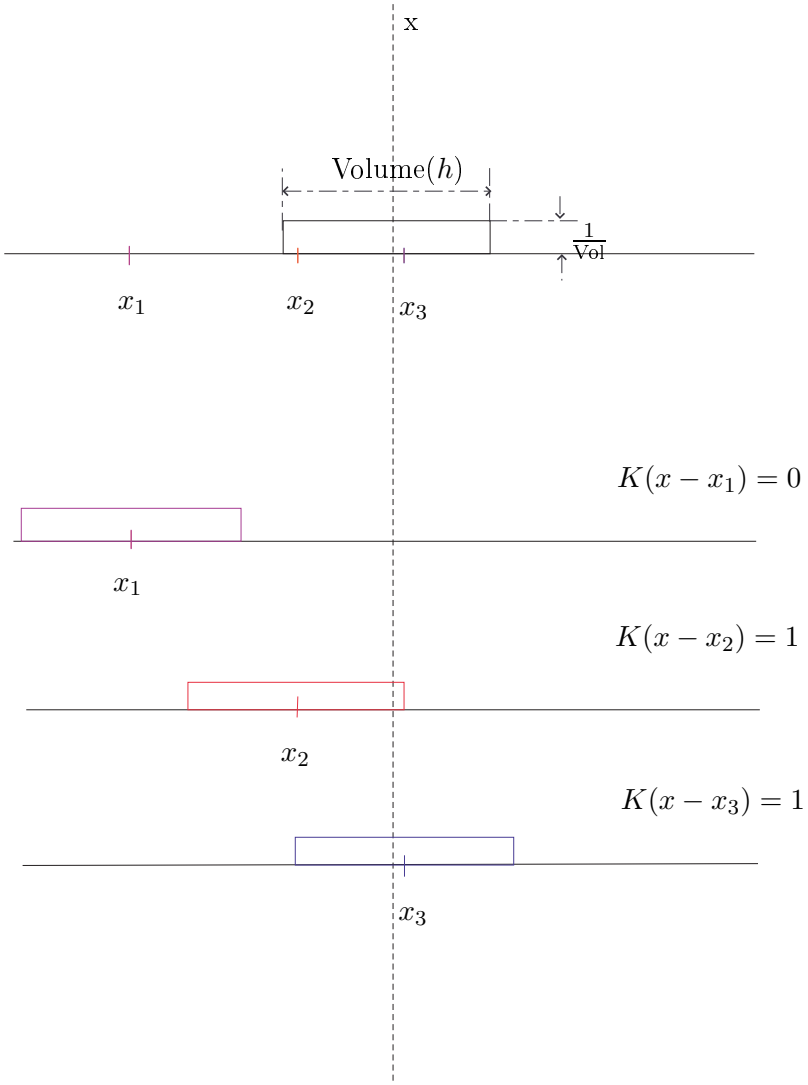


Figure 1.10 – An example of Rectangular Windows estimation. The vertical dashed line indicates the position of x.

which is a binomial distribution with the mean $E[k] = nP$ and the variance $Var[k] = nP(1-P)$. Thus,

$$E[\frac{k}{n}] = P$$
$$V[\frac{k}{n}] = \frac{P(1-P)}{n}$$

When $n \to \infty$, $V[\frac{k}{n}] \to 0$ so that we can estimate P by $\frac{k}{n}$.

However, when the region $\mathcal{X}$ is small, we need to take into consideration the volume (v) of $\mathcal{X}$. The estimation then becomes :

$$P \approx \frac{k}{nv}$$

According to the above principle, Rectangular Windows (of KDE) is proposed by estimating on fixing v and counting the number of observations (k) within v.

Rectangular Windows

Figure 1.10 shows the Rectangular Windows density estimation. In this figure, Denote k a kernel function :

$$k(\frac{x - x_n}{h}) = \begin{cases} 1, & \text{if } x_n \text{ is within the rectangular window centered on } x \\ 0, & \text{otherwise} \end{cases}$$

Then the total number of points inside the rectangular window $n_{\text{inside}} = \sum_{i=1}^{n} k(\frac{x-x_i}{h})$ and the density can be estimated by $\frac{1}{nh}\sum_{i=1}^{n} k(\frac{x-x_i}{h})$.

KDE (Parzen Windows) with Smooth Kernels

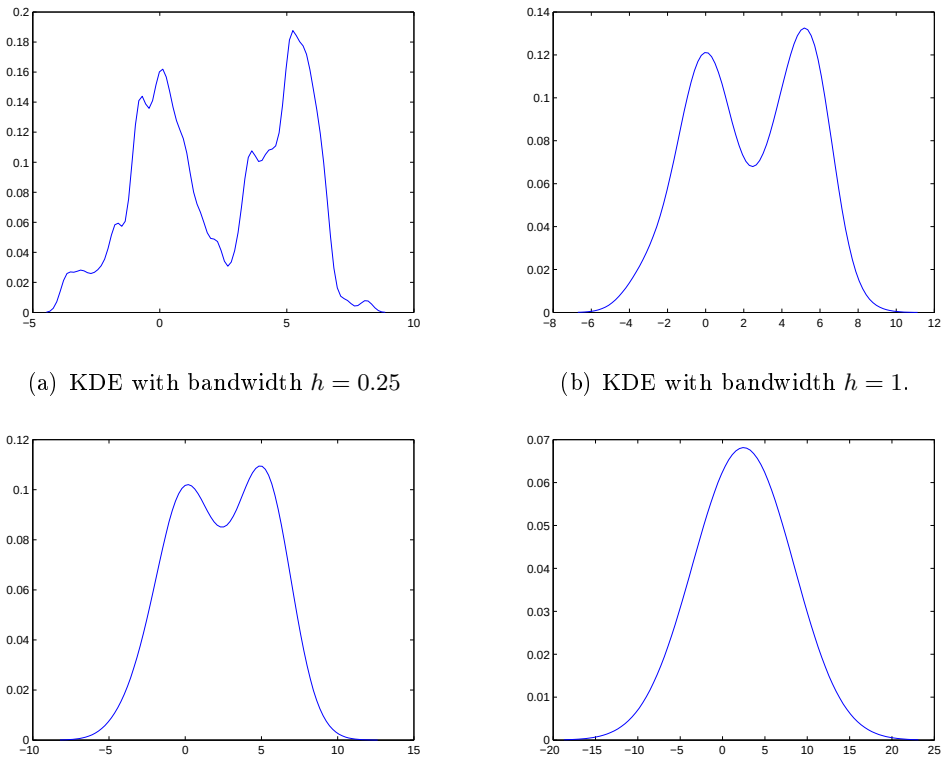
However, Rectangular Windows Estimation distributes equal weight to all observations, regardless of their distance to the center of the window (x in Figure 1.10) and there may be discontinuities in the estimated density. Therefore, smooth kernel functions are used, for example gaussian kernel.

An example of estimation is shown in Figure 1.11.

The h, called smoothing parameter or bandwidth is crucial for density estimation. Figure 1.11(a) shows the extreme cases when h is too small and Figure 1.11(d) shows the case when h is too large. We set h , the one that matches our priors.

1.4.6 Preimage

Generally, explicit kernel mapping is not necessary and the exact correspondence between original input space and kernel space is unclear. To find the corresponding features in the original input space of features from kernel space, preimage is proposed. It has been successfully applied to image denoising.



(a) KDE with bandwidth $h = 0.25$

(b) KDE with bandwidth $h = 1$.

(c) KDE with bandwidth $h = 1.5$.

(d) KDE with bandwidth $h = 5$.

Figure 1.11 – KDE with smooth kernels (bandwidth h). The real distribution is a univariate gaussian mixture : one centered on 0 with a variance of 1.5^2, the other centered on 5 with a variance of 1^2 (estimated using 100 observations).

As the kernel mapping is projecting data into a (much) higher dimensional space, we cannot guarantee the existence of preimage for every $\phi(.)$. However, we can still approximate preimages by minimizing some criterion. For example, in [110], the preimage problem is :

$$\arg\min_z \mathcal{G} = \arg\min_z ||\phi(z) - P_n\phi(x)||^2$$

where $P_n\phi(x)$ is the kernel feature ; z is the preimage of $P_n\phi(x)$ such that the square distance between $\phi(z)$ and $P_n\phi(x)$ is minimized. For an illustration, see Figure 1.12. For image denoising application, $P_n\phi(x)$ can be the KPCA representation of $\phi(x)$ ([86]). If we use gaussian kernels, z can be found by setting the derivative of $\mathcal{G}$ (w.r.t z) to zero. Obviously, $P_n\phi(.)$ can be other kernel representations ([3]).

Then, the previously introduced preimage technique is extended by regularizing $\mathcal{G}$ ([176]) ; some others have also taken into consideration the influence of the some nearest neighbors (from training set $\phi(x), x \in$ Training set) of $P_n\phi(x)$ ([175] ,[69]).

We applied the preimage method proposed in [60]. The objective function is replaced by

$$\min \mathcal{G} = \min \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} ||x_i^T x_j - \Psi_{x_i}^T \Psi_{x_j}||^2 + \lambda ||\Psi_{x_i}^T \Psi_{x_j}||$$

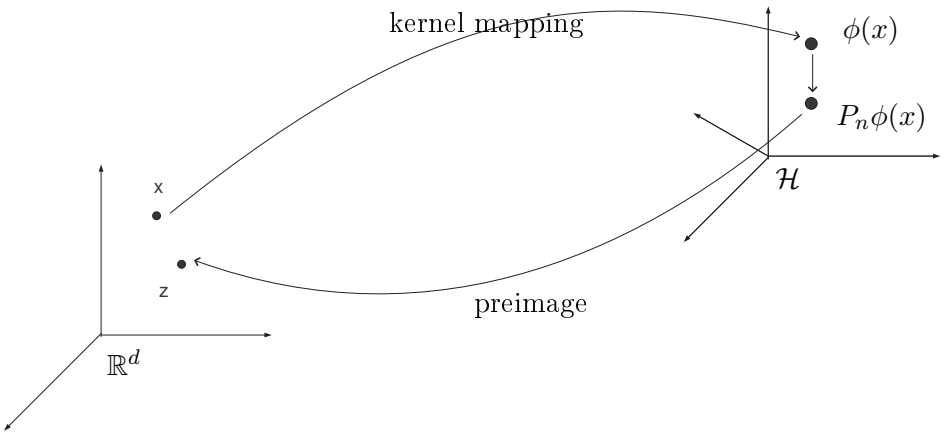


Figure 1.12 – An illustration for preimage

where $\Psi_. = \sum_{i=1}^{k} \alpha_i \phi_k(.)$ and $\alpha_i, \forall i = 1, \ldots, k$ is to be found. Instead of minimizing feature space distances, this $\mathcal{G}$ preserves the isometry of input space and kernel space and finds the preimage by

$$\Psi_{x_i}^T \Psi_z^* = x_i^T z$$

where Ψ_z^* is the KPCA representation of $\Psi(z)$ and z designates the preimage.

In Section 5.5, preimage is applied to show the input space alignments after KPCA domain adaptation alignment.

1.5 Conclusion

In this chapter, we first presented the general framework of statistical learning (Section 1.2), based on which the learning approaches are designed. Then in Section 1.3, the kernel fundamentals were introduced to help the comprehension of the following thesis. A large part of Chapter 1 contributed to introducing machine learning approaches based on kernel methods, namely SVM, KPCA, MMD, KDE, Preimage. Of course, there are many other kernel based approaches ; we limited our presentation to the approaches that we have used. Based on these fundamentals, we developed our domain adaptation approaches (KDE based approach in Section 3.4, SVM and MMD based approach in Section 4.3.3, KPCA based approaches in Section 5.3.2, 5.3.3 and Section 5.4).

Chapter 2

Overview of Transfer Learning

Contents

2.1 Introduction

Traditional machine learning has a major assumption : training and testing data are generated from the same distribution. When this assumption is violated, training and testing data are considered as two different tasks. However, to learn individually from testing data may be difficult and the collect of corresponding training data is tedious and unreliable. *Transfer Learning*, which aims at solving this problem by making full use of easily obtained different but related training data, has attracted more and more attention since 1995. In the transfer learning context, training and testing data can follow different distributions and their priors, features, distributions of features, labels, distributions of labels can all be different. The only assumption is that such training and testing data are related. Taking a good advantage of this relatedness (generally latent) allows the knowledge transfer among data from different distributions.

We first define necessary elements of transfer learning ([94]) :

- **Definition of Domains** : Given observations $x_i \in \mathcal{X}, \forall i = 1, 2, 3, \ldots$, a domain $\mathcal{D}$ is defined by $\mathcal{X}$ and the prior distribution $P(\mathcal{X})$: $\mathcal{D} = (\mathcal{X}, P(\mathcal{X}))$.

- **Definition of Tasks** : Given labels $y_i \in \mathcal{Y}, \forall i = 1, 2, 3, \ldots$, a task $\mathcal{T}$ is defined by $\mathcal{Y}$ and the prediction function f $(f : \mathcal{X} \to \mathcal{Y})$: $\mathcal{T} = (\mathcal{Y}, f(.))$. From the probabilistic point of view, $f(x_i)$ is equivalent to the conditional probability $P(y_i|x_i)$ with $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$. Here after, we use the notation $\mathcal{T} = (\mathcal{Y}, P(\mathcal{Y}|\mathcal{X}))$.

- **Source** : training data with enough information that will help the learning on objective data, denoted by S.

- **Target** : data on which we would like to achieve good performance and which is difficult to train on its own, denoted by T.

- **Definition of Transfer Learning** : Let $\mathcal{D}_S$ and $\mathcal{T}_S$ be the domain and task for source and $\mathcal{D}_T$, $\mathcal{T}_T$ be domain and task for target, respectively, then transfer learning is the technique that helps the learning of target using source, where $\mathcal{D}_S \neq \mathcal{D}_T$ and/or $\mathcal{T}_S \neq \mathcal{T}_T$.

We take spam detection as a real example to illustrate the difference between traditional machine learning and transfer learning as well as the fundamental elements of transfer learning. In spam detection, given a mail $(x, x \in \mathcal{X})$, the objective is to label the mail to be spam or ham (useful mails) $(\mathcal{Y})$. As the same mail may be considered as spam for some users and ham for other users, spam detection task for different users should be considered different. To achieve good detection, traditional machine learning needs training data from every considered user and the relationships among users are not taken into account. Still, for the same user, the classification of spam and ham may change along time. If only insufficient information is provided, we cannot expect good results. However, for transfer learning approaches, there are usually few or even no labeled data from the considered user (target data). Instead of constructing training data from the same target distribution, we take advantage of related data (source), for example the well labeled data from similar users. Obviously, the domains and tasks among similar users are related yet different. How to make full use of the related information is the focus of transfer learning.

2.2 Taxonomy

In this section, we give several taxonomies of transfer learning methods. The taxonomy allows us to easily compare different methods ; it also defines the context of our research. To the best of our knowledge, besides application-based categorization, transfer learning can be categorized by three criteria : the availability of labels in source and target, the differences in feature/label space and the transfer approach. We present the last three categorizations and leave application-based categorization for further research.

2.2.1 Transfer Learning Categorized by availability of labels in source and target

Like traditional machine learning, the label information is also very important in transfer learning. In most of the cases, the label of source is easily accessible while the label of target is usually difficult to obtain.

abundant $\mathcal{Y}_t$

Here, there is no need for transfer learning, because we can adopt directly traditional machine learning methods for the target. But if we have source and target to be learnt simultaneously, *multi-task learning* is proposed ([20]).

abundant $\mathcal{Y}_s$ but scarce $\mathcal{Y}_t$

Because of the scarcity of labeled data in target domain, direct training will lead to a large bias, which degrades the generalization ability. Therefore, transfer learning outperforms traditional semi-supervised learning, as the former benefits from the related source domain knowledge. Many real and synthetic experiments have proved the effectiveness of transfer learning in this case. However, we need to treat different-domain knowledge differently, as the available target domain knowledge is more important than that from source domain. How to make use of the related source domain knowledge, the labeled target domain data and the unlabeled target domain data all together, is the key problem of this type of transfer learning. Typical methods include [30], [32], [38], [59], [157], [159], [160], [166], etc.

abundant $\mathcal{Y}_s$ but no $\mathcal{Y}_t$

Compared with the previous case, "abundant $\mathcal{Y}_s$ but no $\mathcal{Y}_t$" is more challenging. We have no labeled target data to guide the transfer. How to transfer correctly and efficiently becomes a more difficult task. Some literatures assume a shared conditional probability : [62], [167], [10], [40], [131], [132], etc ; some others assume latent relationships between (among) source(s) and target : [11], [12], [53], [54], [92], [93], etc.

neither $\mathcal{Y}_s$ nor $\mathcal{Y}_t$

The difficulty increases and this is the most difficult case. In the literature, the situation is called *unsupervised transfer learning*. One solution is to modify traditional clustering methods to adapt to the transfer learning context. Typical methods include [29], [31], etc.

Generally, the label of target data is crucial to transfer learning task, although some approaches can achieve good performances without taking into consideration target labels. With more and more labeled target data available, the learning tasks becomes easier and easier and the decision becomes more and more reliable. In [174], [21], [150], active learning has been introduced to transfer learning : labeling actively the most ambiguous target data (selected with the help of source data) leads to improvement in efficiency and accuracy.

2.2.2 Transfer Learning Categorized by differences in feature space/label space

inductive transfer learning

- **Definition** ([94]) : given a source and a target, inductive transfer leaning aims at solving transfer learning problems where $\mathcal{Y}_s \neq \mathcal{Y}_t$ or/and $P(\mathcal{Y}_s|\mathcal{X}_s) \neq P(\mathcal{Y}_t|\mathcal{X}_t)$.

- $\mathcal{Y}_s \neq \mathcal{Y}_t$: generally, when label spaces are different, the corresponding conditional probabilities are also different. In other words, there may be different numbers of classes in source and target data, for example clustering source and target into different numbers of clusters. Generally, the classes in source cover a wider range of classes than those of target. There are also cases when new class(es) appears(appear) in target data, which are more challenging.

- $P(\mathcal{Y}_s|\mathcal{X}_s) \neq P(\mathcal{Y}_t|\mathcal{X}_t)$ **while** $\mathcal{Y}_s = \mathcal{Y}_t$: Without equality of conditional probabilities of source and target ($P(\mathcal{Y}|\mathcal{X})$), the equality of source and target no longer holds, even if they have the same domain. Unlike the second case of transductive transfer learning, we need to further take into account the alignment of conditional probabilities of source and target. Most of the literatures presented in Section 2.3.1 belong to this category.

transductive transfer learning

- **Definition** ([94]) : given a source and a target, transductive transfer learning aims at solving transfer learning problems where $\mathcal{X}_s \neq \mathcal{X}_t$ or/and $P(\mathcal{X}_s) \neq P(\mathcal{X}_t)$.

- $\mathcal{X}_s \neq \mathcal{X}_t$: In general, when the feature spaces are different, so are their marginal probability density functions. This situation is denoted as *Heterogeneous Transfer Learning*. A typical example is when we have images as target domain while texts or acoustic information as source. Because of the large difference in the source and the target, a particularly important issue is how to select the effective source features (instances), because transferring some irrelevant features (instances) will degrade the classification performance in target domain (denoted as *negative transfer*). A brief review of recent advances in heterogeneous transfer learning will be given in Section 2.4.

– $P(\mathcal{X}_s) \neq P(\mathcal{X}_t)$ **while** $\mathcal{X}_s = \mathcal{X}_t$: with the assumption that source and target share the same task, if we align $P(\mathcal{X}_s)$ and $P(\mathcal{X}_t)$, we can ensure that, after the alignment, source and target are very similar ; because in probabilistic aspect, the distribution is governed by the joint probability $P(\mathcal{X}, \mathcal{Y}) = P(\mathcal{Y}|\mathcal{X})P(\mathcal{X})$. Therefore, many literatures have contributed to this situation, for example sample selection bias and covariate shift.

implicit transfer learning

– **Definition** : given a source and a target, implicit transfer learning occurs when the source and the target are different in both feature space and label space. However, we assume some latent relationships between source and target data. Sometimes, when label information is unavailable on source and target data, we also consider it implicit transfer learning.

– Implicit transfer learning can be applied to heterogeneous transfer learning and the learning tasks for implicit transfer learning are generally clustering or dimension reduction. [31], [151] for homogeneous transfer learning and [22], [177] for heterogeneous transfer learning.

Note that transfer learning among totally different source and target is meaningless and sometimes negative transfer can degrade the performance. Thus, the relatedness between source and target is a key issue in transfer learning. Suitable relatedness measures (or similarity measures) include KL-divergence ([28]), Maximum Mean Discrepancy ([92]) and other statistical distances [2].

Transfer Learning Taxonomy in Section 2.2.2	Subordinate	$\mathcal{Y}_s$	$\mathcal{Y}_t$
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias/Covariate Shift	$\surd$	$\times$ [3]
Inductive Transfer Learning	Multi-task Learning	$\surd$	$\surd$
	Self-taught Learning	$\times$	$\surd$
Implicit Transfer Learning		$\times$	$\times$

Table 2.1 – Taxonomy of Transfer Learning

2.2.3 Transfer Learning Categorized by transfer approach

instance-based transfer learning

Instance-based transfer learning connects source and target by instances or reweighed instances.

In importance sampling based transfer learning approaches, the instances from source are weighted according to source and target distributions and then such instances are used to help

2. A summary of statistical distance : `https://en.wikipedia.org/wiki/Statistical_distance`.

3. Generally, some labeled target data can improve the learning performances and in some literatures, few $\mathcal{Y}_T$ is supposed available.

the target machine learning tasks. More details are in Section 2.3.2.

TrAdaboost ([30]) is another instance based approach with a few labeled target data needed. It follows the principle of boosting strategy : a supervised machine learning approach that combines the "weak classifiers", by weighing differently the data instances, to form a final "strong classifier" (a more accurate classifier). To fit the transfer learning context, *TrAdaboost* treats the labeled source data and the labeled target data differently : it applies traditional boosting method (*Adaboost*) for labeled target data, augmenting the weights for instances that are misclassified by the previous "weak classifier", while for the source, it decreases iteratively the weights of the misclassified instances using the same "weak classifier" as in the target domain. In this way, *TrAdaboost* is expected to avoid the possible negative transfer and meanwhile boosting the performance of target-domain "weak classifier" with the help of source data.

Some other approaches contribute to selecting the most influential source instances to target data, for example, DASVM ([18]). The SVM based only on source data can be considered as an initial result. With more and more target data, the SVM is iteratively adapted jointly by target instances and influential source instances.

feature-based transfer learning

Feature-based transfer learning aims at finding a feature space where source and target are similar. Some transfer source (target) to the same feature space of target (source) ; some transfer both source and target to a shared common feature space. The final common feature spaces can be of lower dimensionality or higher dimensionality and the way of finding such feature spaces is different depending on the specific case considered. Brief summaries are given in Section 2.3.1 and Section 2.3.2.

parameter-based transfer learning

In this category, parameters are shared between source and target. For example, in SVM based transfer learning, a common margin parameter can be shared between source and target (more details in Section 2.3.1 and Section 4.2.1). In [48], a weighting parameter that indicates the neighborhood is transferred between source and target. Most parameter-based transfer learning approaches contribute to transferring free parameters among source and target probabilistic models ([71], [13], [114]). Generally, when there are a few source domains, better performances are observed than in the single source transfer learning case.

relation-based transfer learning

Relation-based transfer learning consists principally in network-based transfer learning, for example, Markov logic network based transfer learning ([85], [33]), bayesian network based transfer learning ([146]). There are also deep learning related transfer learning that transfers the CNN networks. For a brief survey on transfer learning with deep CNN, readers can refer to [27] and [96].

Moreover, there are hybrid methods that combines instance-based and parameter-based transfer learning principles : [155].

2.3 Overview of Homogeneous Transfer Learning

2.3.1 Inductive Transfer Learning

Instance-based approaches

TrAdaboost ([30]) is a well known instance based inductive transfer learning proposed by Dai et al. : Dai et al. weight source data so that well related source data influence more the final decision while the less related source data have reduced effect on the decision (the error is evaluated only on target data).

Then it is extended in [166] to multiple sources setting. They first train only on source data to obtain weak classifiers. Then, the weak classifiers of source are used as input to target domain boosting phase. The final classifier is found by boosting sources weak classifiers on labeled target data.

[160] further extends [166] to multi-source and multi-view transfer learning. The first step is the same as in [166]. Then, they apply multi-view to transfer boosting algorithm : the weights of weak classifiers are determined by the agreement of classification results on different views.

Feature-based approaches

In [104], authors propose to construct transferrable features (b_j). First, they optimize on unlabeled source data :

$$\min_{b,a} ||x_S^i - \sum_j a_j^i b_j||_2^2 + \lambda ||a^i||_i, \text{ s.t. } ||b_j||_2 \leq 1, \ \forall j = 1, \ldots, s$$

Then new features of target data are computed by a_T where $a_T^i = \arg\min_{a_T^i} ||x_T^i - \sum_j a_{T_j}^i b_j||_2^2 + \lambda ||a_T^i||$. Finally SVM is applied to final features.

In [123], a discriminative clustering based metric learning transfer approach is proposed : both the discriminability of target clusters and the discriminability of source and target are taken into account ; thus, with the help of source clustering, the final clustering works well on target data.

Parameter-based approaches

In inductive transfer learning, as there is a distribution shift in conditional probabilities $p(y_s|x_s) \neq p(y_t|x_t)$, to align the conditional probabilities with transferring parameters, multiple sources transfer learning is generally the case considered. At least, there should be some labeled target data to help the transfer. The parameters to be transferred depend on the model constructed. To the best of our knowledge, the model can be SVM or Gaussian Process based classification.

For SVM based inductive transfer learning, the margin parameters can be transferred or even the decision function can be transferred. Here,we list three different SVM based inductive transfer learning :

- **Transferring directly the decision functions** :
  In [38], source decision models are integrated into the transfer learning task :

$$\min_{d,w,b,\beta,f_T} \frac{1}{2}(||w||^2 + \lambda||\beta||^2) + C\sum_{i=i}^{n_t^u} l_\epsilon(f_T(x_i) - f(x_i)) + \frac{\gamma}{2}\sum_{s=1}^{m_S} d_s \sum_{i=1}^{n_t^u}(f_T(x_i) - f_S(x_i))^2$$
$$\text{s.t.} \quad f(x) = \sum_{s=1}^{m_S} d_s\beta_s f_S(x) + < w, \varphi(x) > + b$$
$$\sum_{s=1}^{m_S} d_s \geq 1, \quad d_s \in \{1, 0\}$$

  where n_t^u is the number of unlabeled target data ; m_S is the number of sources ; l_ϵ is the loss function ; f_T is the target classifier on unlabeled data ; $f(.)$ is the classifier found by source classifier $\sum_{s=1}^{S} d_s\beta_s f_S(x)$ and labeled target data ; d_s, β_s are the weights ; λ, γ are parameters. The final target classifier is found as a compromise between classification performances on labeled target and the sources models. Similar approaches include [36]

- **Transferring margin parameters** :
  [139] and [35] transfer the margin parameters. For example, in [139], the target model is found by optimizing :

$$\min_{w,b} \frac{1}{2}||w - \sum_{s=1}^{m_S} \gamma_s w_s||^2 + C\sum_{i=1}^{n_t^l}(y_i - < w, \phi(x_i) > - b)^2$$
$$\text{s.t.} \quad 0 \leq \gamma_s \leq 1, \ \forall i = 1, \ldots, m_S;$$
$$\text{with} \quad w = \sum_{s=1}^{m_S} \gamma_s w_s + \sum_{i=1}^{n_t^l} \alpha_i \phi(x_i)$$

  where m_S represents the number of sources and n_t^l represents the number of labeled target data ; the final w is a compromise between classification error on labeled target data and a combination of source-classifiers parameter w_s.

- **Transferring inductively taking into account the transformation between source and target** :
  In [59], not only the parameter but also the transformation between source and target (U) are taken into account in a single objective function :

$$\min_{U,w,b} \frac{1}{2}||U||_F^2 + \frac{1}{2}||w||^2$$
$$\text{s.t. } y_s^i(\begin{bmatrix} x_s^i \\ 1 \end{bmatrix}^T \begin{bmatrix} w \\ b \end{bmatrix} \geq 1), \ \forall x_s^i \in \mathcal{D}_S$$
$$y_t^i(\begin{bmatrix} x_t^i \\ 1 \end{bmatrix}^T U^T \begin{bmatrix} w \\ b \end{bmatrix} \geq 1), \ \forall x_t^i \in \mathcal{D}_T$$

Other SVM based inductive transfer learning approaches include [63], [4], [37], [79], etc.

For Gaussian Process based inductive transfer learning, the gaussian process parameters can be transferred among sources and target. Related works include [71], [13], [114].

2.3.2 Transductive Transfer Learning

Instance-based approaches

For Transductive Transfer Learning, one of the important research branch is the sample selection bias/covariate shift : the conditional probability $p(y|x)$ is supposed to be unchanged between source and target and there is a single support $(\mathcal{X} \times \mathcal{Y})$ for source and target ; only the marginal distributions differ $(p_S(x_s) \neq p_T(x_t))$.

If we base on the expected risk minimization ($R[f]$, which equals to $E[l(x, y, f)]$, in Section 1.2.4), the best solution for target data is [4] :

$$\begin{aligned}
\theta^* &= \arg\min_{\theta\in\Theta} E_{P_T(x,y)}[l(x, y, \theta)] \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} l(x, y, \theta)P_T(x, y)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} P_T(y|x)P_T(x)l(x, y, \theta)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} P_S(y|x)P_T(x)l(x, y, \theta)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} \frac{P_T(x)}{P_S(x)}(P_S(y|x)P_S(x))l(x, y, \theta)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} \frac{P_T(x)}{P_S(x)}P_S(x, y)l(x, y, \theta)dxdy
\end{aligned}$$

To estimate θ^*, we can make use of the labeled source data and estimate expected risk by empirical risk :

$$\hat{\theta}^* \sim \arg\min_{\theta\in\Theta} \sum_{(x_i,y_i)\in\mathcal{S}} \frac{P_T(x_i)}{P_S(x_i)} l(x_i, y_i, \theta)$$

so that the target solution can be found by weighting the source solution (by $\frac{P_T(x_i)}{P_S(x_i)}$). This approach is called *importance sampling* and it has been widely used in sample selection bias/covariate shift transfer learning problems. Related literatures include [62], [167], [10], [40], [131], [132], [140], [124], [26], [106], etc. (More details can be found in Section 3.3.)

However, previously introduced approaches require good estimation of marginal probabilities. Thus, other approaches have been proposed for sample selection bias/covariate shift. With the same assumption, in [157], a PageRank Algorithm ([95]) based refinement strategy is proposed to attribute a reliable label to a document. There are two steps :

4. A necessary condition is that the support of $P_T(.)$ is the same as the support of $P_S(.)$ ([62]), denoted by $\mathcal{X} \times \mathcal{Y}$.

- Step 1 : take advantage of both labeled source and unlabeled target, starting from an initial unrefined confidence score, a refined confidence score is generated.

- Step 2 : use the refined confidence score on only unlabeled target to re-generate further refined score. This final score will be used to provide reliable prediction of labels.

The algorithm of generation of refined confidence score can be found in [157] Algorithm 1.

Feature-based approaches

Generally, feature-based transductive transfer learning aims to find a common feature space that bridges source and target. In this common feature space, it is expected that source and target are very similar.

- *Feature augmentation* :

  To find a common feature space, one approach is that source and target features are augmented by common features and target-specific or source-specific data ([32]) :

  $$\varphi_s(\mathcal{X}_s) = \left[\mathcal{X}_{common},\ \mathcal{X}_{source-specific},\ \mathbf{0}\right] \quad \varphi_t(\mathcal{X}_t) = \left[\mathcal{X}_{common},\ \mathbf{0},\ \mathcal{X}_{target-specific}\right] \quad (2.1)$$

  where $\varphi_{\{s,t\}}$ represents the feature augmentation on source or target and $\mathbf{0}$ represents zero padding. After this transformation, all data ($\mathcal{X} \in \mathbb{R}^d$) are represented by common features, source-specific features and target-specific features ($\mathcal{X}_{aug} \in \mathbb{R}^{3d}$). For source (target) data, there is no target-specific (source-specific) features and we use 0s to fill in. Finally, traditional machine learning technique can be applied on $\varphi_s(\mathcal{X}_s)$ and $\varphi_t(\mathcal{X}_t)$.

- *Pivot feature* :

  [12] proposes another solution to common feature space based transductive transfer learning. Authors first define a set of *pivot* features. Such features frequently appear in both source and target data and can be extracted without the label information of both domains. Then, the mapping (U) from original data to a low-dimensional common feature space is found. With the help of labeled information, a classifier is trained on labeled source (no labeled target) augmented with UX_s (shared features). Bliter et al. further extend the idea to take into account the label information ([11]). The pivot features selection is based on both the co-occurrence in both source and target and the mutual information between pivot features and the source labels. Then in the final step, with a few labeled target data, the performance of the trained classifier can be improved.

- *Dictionary Learning* :

  For more complicated applications, dictionary based transfer learning is proposed. Like the pivot features, the dictionary connects source and target. Related works are [101], [91], [121], [98]. For example, in [121], source and target are transformed to a shared latent space, based on which a shared discriminative dictionary is built.

– *Common projection subspace* : To find a common feature space, some project source and target data so that after projection, source and target are expected to be as similar as possible.

- In [93] and [92], Maximum Mean Discrepancy(MMD) is used as a similarity criterion to control the projection : finding a subspace where MMD is the smallest with respect to some constraints can lead to good transfer learning results. Similar literatures can be found in [125], [141], [87], [74], [36] : either other similarity criteria are applied or the constraints are modified and/or regularization terms are added.

- [54] and [53] propose to interpolate the change from source to target by geodesic flow. The features of intermediate domains (of low-dimensionality) form new shared common space, which can help the learning task on target.

- Metric learning has also been extended to transfer learning : given any two observations $x_i, x_j \in \mathcal{X}$, a distance metric can be defined as $(x_i - x_j)M^T M(x_i - x_j)^T$; in transfer learning, a general framework of metric learning ([159]) is proposed as follows :

$$\min_{M,w,f} tr(M^T M) + \lambda\psi(w) + \beta l(f, M, w; \mathcal{D}_S, \mathcal{D}_T^l)$$

  where the first term is a regularization term that controls the complexity of M ; the second term is to regularize instance weights (w) ; the third term is the loss function with the decision function f applied to weighted (by w) source data $(\mathcal{D}_S)$ and transformed (by M) labeled target data $(\mathcal{D}_T^l)$. Such a framework generalizes classification task and regression task in one framework and it summarizes the ideas of metric transfer learning in [46], [172] and [19] (most related).

2.3.3 Implicit Transfer Learning

Inductive transfer learning deals with the cases when source and target are different in tasks $(\mathcal{T}_S \neq \mathcal{T}_T)$, while transductive transfer learning deals with the cases when source and target share the same task but differ in domains $(\mathcal{D}_S \neq \mathcal{D}_T)$. Implicit transfer learning takes into account both differences $(\mathcal{T}_S \neq \mathcal{T}_T$ and $\mathcal{D}_S \neq \mathcal{D}_T)$. Thus, it is a more challenging transfer learning problem and to the best of our knowledge, there have not been many researches in this area.

– *Clustering based implicit transfer learning* :
In [151], a dimension reduction based unsupervised transfer learning approach is proposed. [151] reduces different source and target domains into a common subspace where the intra-cluster differences are minimized and inter-cluster differences are maximized (the principle of Linear Discriminant Analysis). Moreover, [151] can also take advantage of the label information of source data and the intrinsic structure of unlabeled target data. Other clustering based implicit transfer learning approaches include [31].

– *Aligning both marginal and conditional probabilities* :

- In [22], marginal distributions are first aligned by landmark selection with Maximum Mean Discrepancy (detailed in Section 4.2). Then to align the conditional distributions :

with sources and target sharing the same labels ($\mathcal{Y}_s = \mathcal{Y}_t$), multiple sources data are used to first find source models ($f_s^i(.), i = 1, \ldots, n_{\text{source}}$). Given labeled target sample x_t, source models on x_t are weighted to approach the true label of x_t :

$$\sum_{i=1}^{n_{\text{source}}} \beta_i f_s^i(x_t) \approx y_t = f_t(x_t)$$

where $f_t(.)$ is the objective decision function. As $f_{\{s,t\}}$ is equivalent to $p_S(y|x_s)$ and $p_T(y|x_t)$, respectively, with weights β, $p_S(y|x_s)$ is aligned to $p_T(y|x_t)$. Similar approaches can be found in [61].

- Others build meta features based on multi-views ([155]). The learning approach is to consider every view as a source. Thus, [155] is equivalent to multi-sources based implicit transfer learning in [22].

- [178] also chooses to combine the models intelligently to align conditional probabilities. But differently from former literatures, their weighting strategy is based on cross validation. Then, to align marginal distributions, [178] applies importance sampling.

2.4 Overview of Heterogeneous Transfer Learning

Homogeneous transfer learning can be extended to heterogeneous transfer learning. In the latter transfer learning context, source and target have different feature spaces, for example we can use text data as source to help the learning on image data (target). Their relatedness is usually latent and reflected by some co-occurrence matrices (if there exist). How to take good advantage of co-occurrence matrices is studied in many literatures (Section 2.4.1). Generally, the reliability of co-occurrence matrices greatly influences the learning performance. However, there are cases when the co-occurrence information is difficult to collect and there is no available prior correspondences between source and target data. In such cases, the transfer learning task becomes more challenging and new models should be used (Section 2.4.2). Popular heterogeneous transfer learning with/without co-occurrence information are reviewed in this section.

2.4.1 Heterogenous Transfer Learning with Co-occurrences

Although source and target have different feature spaces, we can take advantage of their relationship to transfer knowledge. Usually, the relationship is modeled by co-occurrences between (among) source(s) and target domains.

One of the most intuitive approaches is to find for every domain a transformation matrix and to project all data (sources and target) into the same latent space. In [148], it is supposed that there are $l-1$ source domains and authors transform l times (for both sources and target) by manifold alignment. For sources domains, they use the co-occurrence information and never-co-occurrence information to build similarity and dissimilarity matrices, respectively (similarly to [161]). For target domain, they use $\exp(-||x_t^i - x_t^j||^2)$ to model the similarity matrix. A regularization technique is applied to achieve the transfer learning objective.

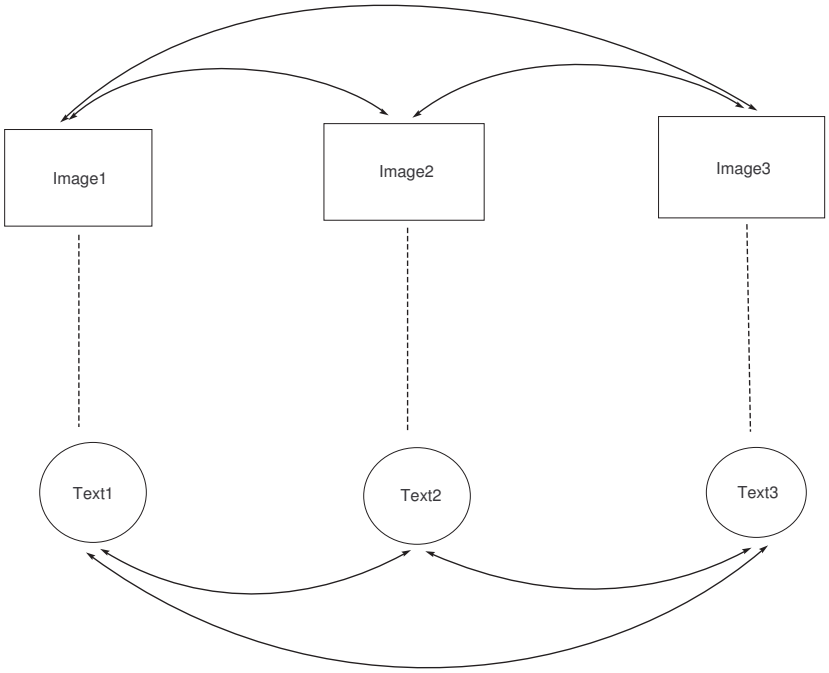


Figure 2.1 – An illustration of coupled Markov chain heterogeneous transfer learning.

Probabilistic approaches

In [28], a probabilistic correspondence is modeled by a Markov chain. In source space, we have $c \to z_s \to x_s$, while in target, the Markov chain model is $c \to z_t \to x_t$, where c represents the label information and $z_{\{s,t\}}$ represents the features of data $x_{\{s,t\}}$. To relate source and target, the co-occurrence is the relationship $z_s \to z_t$ and the translated chain is $c \to z_s \to z_t \to x_t$. The objective function is the risk :

$$R(c, x_t) \approx \Delta(\hat{\theta}_c, \hat{\theta}_{x_t}) p(\hat{\theta}_c|c) p(\hat{\theta}_{x_t}|x_t)$$

with $\hat{\theta}_c = \arg\max_{\theta_c} p(\theta_c|c)$ and $\hat{\theta}_{x_t} = \arg\max_{\theta_{x_t}} p(\theta_{x_t}|x_t)$, representing the classification models only related to c and x_t, respectively ; with classification models integrated, the Markov chains become :

$$\hat{\theta}_c \to c \to z_t \to x_t \to \hat{\theta}_{x_t}$$
$$\hat{\theta}_c \to c \to z_s \to z_t \to x_t \to \hat{\theta}_{x_t}$$

Based on the above chains, $\Delta(\hat{\theta}_c, \hat{\theta}_{x_t})$ (representing the dissimilarity between models $\hat{\theta}_c$ and $\hat{\theta}_{x_t}$) is measured by KL-divergence : $KL(p(z_t|\hat{\theta}_c)||p(z_t|\hat{\theta}_{x_t}))$.

A coupled Markov chain based heterogenous transfer learning is proposed in [90], [134], [154] and [163]. A simple illustration is presented in Figure 2.1 : the rectangle represents the image data and the circle represents the text data ; the co-occurred text and image are connected by dotted lines ; the solid curve represents the transition from an instance to another instance based on transition matrix. In this context, the transition matrix is calculated by similarity between

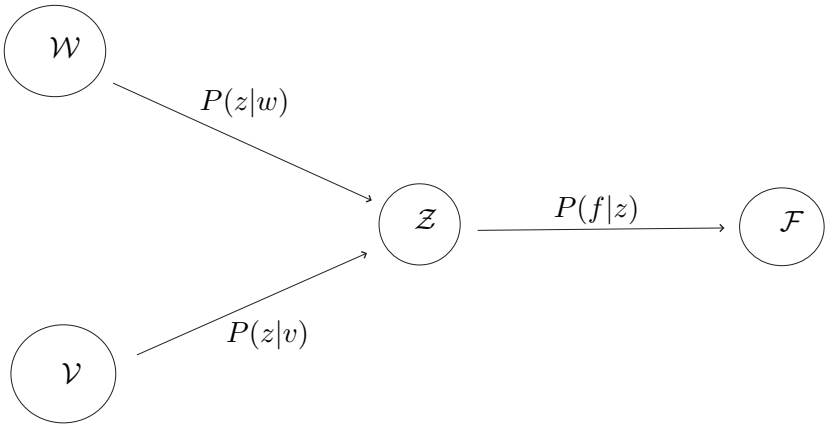


Figure 2.2 – Model of *aPLSA*.

instances : the similarity among instances cross-domain ([154]) ; or the similarity taking into account both intra-domain (target domain) instances and inter-domain (cross-domain) instances ([90], [134], [163]). The co-occurrence rate helps to construct the similarity cross domains.

Another probabilistic approach is *aPLSA* model ([164]). *aPLSA* can be summarized as in Figure 2.2 : $\mathcal{W}$ represents the source - text data and $\mathcal{V}$ represents the target - image data ; $\mathcal{Z}$ is the shared latent variable that indicates the cluster belongings ; $\mathcal{F}$ is the common feature space. The objective is to estimate the clustering function : $g = \arg\max_{z \in \mathcal{Z}} P(z|v), \forall v \in \mathcal{V}$ with the help of two co-occurrence matrices : text-to-feature co-occurrence matrix (A_{ij} is the frequency of the feature $f_j (\in \mathcal{F})$ appearing in the instance $v_i (\in \mathcal{V})$) and image-to-feature co-occurrence matrix (B_{ij} is the frequency that $f_j (\in \mathcal{F})$ co-occurs with the text feature $w_i (\in \mathcal{W})$). The optimization algorithm is EM clustering.

Feature based approaches

Other papers use feature-based transfer learning methods. For example, in [100], authors first proposed a similarity measure between source and target in heterogeneous transfer learning case, denoted as d (a similarity after some transformation on source and target data with the help of co-occurrence rate). Then the objective function is formed :

$$\min_{f \in \mathcal{F}} \gamma l(f, X_S, X_T) + \frac{\eta}{2} \sum_{i,j=1}^{n_t} \mathcal{S}(Q_{ij}, d_{ij}) + \Omega(f)$$

The first term is the loss function, in [100], the optimal f is acting on a common feature space of source (UX_S) and target (VX_T) and the optimization can be transformed to minimizing w.r.t U and V. The second term is a penalty term that forces similar instances in target domain (the similarity is measured by Q) to remain similar after transformations (the similarity is measured by d). The third term is a complexity regularization term. A similar idea can also be found in [99].

[179] also proposed a feature-based heterogeneous transfer learning approach. In this paper, common tags for source and target data should be available

$$\text{text (source)} \overset{F}{\Longleftrightarrow} \text{tag}(V) \overset{G}{\Longleftrightarrow} \text{image (target)}$$

where G and F are the co-occurrence information matrices and V is the latent variable that represents the shared tags. The learning problem can be :

$$\min_{U,V,W} \lambda||G - UV^T||_F^2 + (1-\lambda)||F - WV^T||_F^2 + \Omega(U,V,W)$$

where U and W are latent semantic representations of image and text, respectively ; $\Omega(U,V,W)$ is the regularization term. A collective matrix factorization ([126]) can be used to optimize the above criterion.

2.4.2 Heterogenous Transfer Learning without Co-occurrences

To the best of our knowledge, heterogeneous transfer learning without co-occurrences models are principally developed from homogeneous transfer learning. Thus, a list of such models is presented.

Heterogeneous Feature Augmentation

One direct approach is to find a common feature space where source and target data become similar. In [39] and [72], the common feature space is found by feature augmentation (first proposed in [32]). Similarly to Eq. 2.1, all data are represented by common features, source-specific features and target-specific features. Unlike [32], the common features are found by projecting source and target data to have the same dimensionality :

$$\varphi'_s(\mathcal{X}_s) = \left[\mathcal{X}_s P,\ \mathcal{X}_s,\ \mathbf{0}\right] \quad \varphi'_t(\mathcal{X}_t) = \left[\mathcal{X}_t Q,\ \mathbf{0},\ \mathcal{X}_t\right] \tag{2.2}$$

where P and Q are projection matrices to be determined. Finally, a SVM is trained on $\varphi'_s(\mathcal{X}_s)$ and $\varphi'_t(\mathcal{X}_t)$:

$$\begin{aligned}
\min_{P,Q} \min_{w,\xi_s,\xi_t,b} & \frac{1}{2}||w||^2 + C(\sum_{i=1}^{n_s} \xi_s^i + \sum_{j=1}^{n_t} \xi_t^j) \\
\text{s.t.} & y_s^i(w^T\varphi'_s(x_s^i) + b) \geq 1 - \xi_s^i, \quad \xi_s^i \geq 0 \\
& y_t^j(w^T\varphi'_t(x_t^j) + b) \geq 1 - \xi_t^j, \quad \xi_t^j \geq 0 \\
& ||P||_F^2 \leq \lambda_P, \quad ||Q||_F^2 \leq \lambda_Q
\end{aligned}$$

where ξ_s^i depends on P and ξ_t^j depends on Q (by Eq 2.2) ; $\lambda_{\{P,Q\}}$ are parameters. Note that few target labels are necessary.

Graph based Heterogeneous Transfer Learning

Graph based homogeneous transfer learning has also been extended to heterogeneous transfer learning. Different from the homogeneous case, a joint graph regularization is used to align transformed source and transformed target data (the transformations are independent). There is a Laplacian matrix (Annexe 2.1) indicator that distinguishes the case when source and target belong to the same category or not. Related references include [168].

Asymmetric Metric Transfer Learning

[66] establishes directly the connection between source and target by an asymmetric transformation matrix $W : W \in \mathbb{R}^{d_s \times d_t}$. The learning task can be :

$$\begin{aligned} &\min_W r(W) + \lambda \sum l(X_s W X_t^T) \\ &\text{s.t.} l(x_s^i W x_t^j) = (\max(0, v - x_s^i W x_t^j))^2, x_s^i \in X_s, x_t^j \in X_t \text{ and } x_s^i, x_t^j \in \text{S} \\ &\quad l(x_s^i W x_t^j) = (\min(0, u - x_s^i W x_t^j))^2, x_s^i \in X_s, x_t^j \in X_t \text{ and } x_s^i, x_t^j \in \text{D} \end{aligned} \tag{2.3}$$

where $x_s^i, x_t^j \in$ S(D) represents that x_s^i, x_t^j are from the same category (different categories) ; u and v are two threshold predetermined ; $r(W)$ is a regularization term w.r.t W (in [66], $r(W) = \frac{1}{2}||W||_F^2$). In their paper, a kernelized version of problem 2.3 has also been proposed.

Meta-feature Heterogeneous Transfer Learning

In the previously presented approaches, the label space of source and target should be similar. However, in [44], there is no such assumption. As there is no help of co-occurrence information, direct mapping from target to source is computationally expensive. Therefore, authors construct meta-features : features selected from source(s) that are most similar to target features. The selection is based on the similarity between meta-features and target data. Then, find the final decision rule on {meta-features $\cup$ target features} such that the minimum error of l(meta-features, $\mathcal{Y}_S$) is achieved. Finally, this decision rule is applied to target.

2.5 Comments on Negative Transfer

2.5.1 Negative Transfer

Although transfer learning can be applied in a variety of cases, however, in some literatures, if we compare the classification performances achieved using traditional machine learning approaches and transfer learning approaches, we can observe transfer learning degrades the performance in some specific learning task. Usually, when unrelated source(s) is(are) used during the learning process, such negative transfer will occur. [108] discusses negative transfer problem and negative transfer is shown in their experiments.

2.5.2 Overview of literatures against negative transfer

Generally, the principal strategy against negative transfer is to select among available source(s) so that only the related source data is used :

- in [42], the transfer relationship is modeled by a graph of available source models (vertex) and the transferability in-between (edges). The transferability (from source S_i to source S_j) is defined as the change in performance on task $\mathcal{T}_{S_j}$ between learning with and without transfer from source S_i ; only positive transfer is retained.

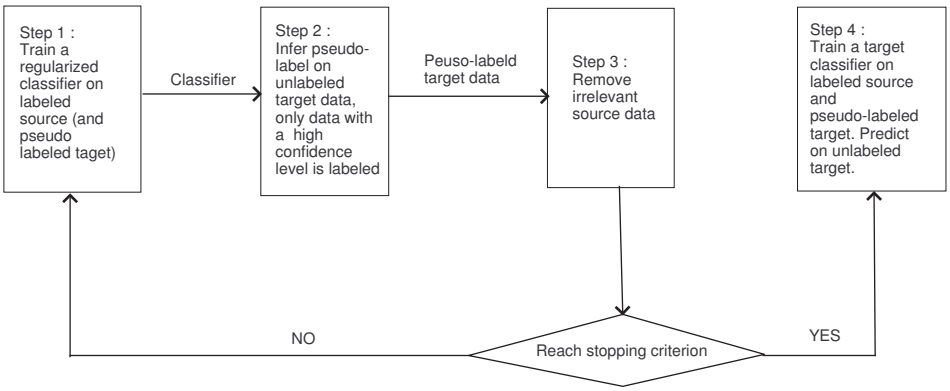


Figure 2.3 – Approach proposed in [115].

- in [50], relatedness of every source model is estimated by *Supervised Local Weight Scheme* : finding the weight of every source model so that the best compromise is found between the classification error on labeled target data and the value of a spectral regularization term with the same source model on unlabeled target data. In such a way, suitable source models can be found and so are the corresponding weights. These weights are then used to incorporate source model information into target classification task. Therefore, negative influence of unrelated source models can be mitigated. The strategy is motivated by [48] (Local Structural Mapping) and [50] achieves better performance than [48].

- in [22], different from the above strategy, an extra instance selection strategy is further applied : the selected instances will render the Maximum Mean Discrepancy (between source instances and target instances) minimum.

- in [115], authors relate the source and target by iteratively selecting the relevant source data. A summary of [115] is shown in Figure 2.3.

Similar approaches can be also found in [156] and [56].

There are not many literatures on negative transfer learning and how to avoid negative transfer learning more effectively and more efficiently remains an important issue to be further studied.

2.6 Applications

2.6.1 Applications for Homogeneous Transfer Learning

Transfer learning has been applied to many real situations. In [94], a few applications have been listed (see Table 2.2).

A more detailed application list of homogeneous transfer learning can be found in [81]. There are information retrieval (especially Natural Language Processing - NLP), biological data processing, TCP (Transmission Control Protocol) intrusion detection, acoustic applications, medical data processing.

text	different subcategories but same parent category	20-Newsgeoups, SRAA, Reuters-21578
email	spam or ham	2006 ECML/PKDD discovery challenge
Wifi-location	location task based on WIFI	ICDM-2007 Contest
Sentiment classification	different products with similar product reviews from Amazon.com	Amazon sentiment classification

Table 2.2 – Some applications for homogeneous transfer learning ([94]).

Transfer Learning has also achieved success in Computer Vision. There are surveys especially for domain adaptation in computer vision : [119], [96] and [27]. Transfer Learning tasks vary from face recognition, object detection, pedestrian re-identification, video concept classification, activity recognition, etc.

[77] summarizes more applications by sectors (biology, finance, business, etc).

2.6.2 Applications for Heterogeneous Transfer Learning

In Table 2.3, we show a list for heterogeneous transfer learning applications.

image recognition	using text information to help image recognition
cross-language text classification	texts from different source languages
drug efficacy classification	[122]
software defect classification	[88]

Table 2.3 – Some applications for heterogeneous transfer learning.

2.7 Conclusion

In this chapter, we first presented the general transfer learning context, which is different from traditional machine learning. Then more details were provided on transfer learning approaches : taxonomies of transfer learning (in Section 2.2.1) ; homogeneous transfer learning approaches (in Section 2.3) and heterogeneous transfer learning approaches (in Section 2.4). Although transfer learning can benefit from inter-domain relationships, it also has limitations : when source and target are unrelated, transfer learning can degrade the performance (negative transfer). There are literatures against negative transfer (in Section 2.5), but further research remains to be done. In the last section, we listed some well-known transfer learning applications.

We focus our works on single source domain homogeneous transductive transfer learning. In the following chapters, we will start from sample selection bias/covariate shift and relax the assumption on the equality of the conditional probabilities of source and target little by little : first, in Chapter 3, $\exists\ A, b : p_S(y|x, x \in \mathcal{S}) = p_T(y|Ax + b, x \in \mathcal{T})$ where A, b are parameters ; then, in Chapter 4 and Chapter 5, $\exists\ g(.) : p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$ where $g(.)$ is a smooth function (generally nonlinear). We aim at rendering source and target data as similar as possible so that a classifier trained on source data can also work well on target. In Section 5.6, experimental results will show the efficacy of our approaches.

Chapter 3

Covariate Shift and Relaxed Covariate Shift

Contents

3.1 Introduction

Based on the transfer learning fundamentals presented in the previous chapter, we now start to consider thoroughly, from a probabilistic point of view, homogeneous transfer learning (the feature space is shared by source and target) where the label space is also the same between source and target. We first consider the covariate shift,

$$p_S(y|x,\ x \in \mathcal{S}) = p_T(y|x,\ x \in \mathcal{T}) \tag{3.1}$$

The difference between source and target is only on the prior probabilities on $x : p_S(x,\ x \in \mathcal{S}) \neq p_T(x,\ x \in \mathcal{T})$. Aligning priors is enough to achieve good transfer.

However in many cases, the Assumption 3.1 is not always satisfied. Therefore, recent works aim at aligning both conditional probabilities and prior probabilities with some latent relationship shared between source and target. With the same objective, we explicitly model the latent relationship : in this chapter, we propose to solve transfer learning problems within the assumption :

$$\exists A, b : p_S(y|x,\ x \in \mathcal{S}) = p_T(y|Ax + b,\ x \in \mathcal{T}) \tag{3.2}$$

where A and b are parameters. This assumption relates source and target. Suitable A and b are expected to lead to good transfer.

Then in the two following chapters (Chapter 4 and Chapter 5), the above parametric assumption is further relaxed to a more general case, see Assumption 4.1.

This chapter is organized as follows : we will first introduce dataset shift in a probabilistic way where covariate shift is considered as a specific kind of dataset shift. Then, a brief review of covariate shift transfer learning is presented in Section 3.3. To adapt to more realistic situations, relaxed covariate shift approaches appeared (presented in Section 3.4). In Section 3.5, we use our proposed method (presented in Section 3.4) to solve a synthetic transfer learning problem where standard covariate shift might fail.

3.2 Background

3.2.1 Context and Definition of Covariate Shift

The objective of transfer learning is to take advantage of source to help the learning in target domain, even though there is some dataset shift in-between. In another aspect of transfer learning, we consider here to model dataset shift : for every common form of dataset shift, we relate a probabilistic model. In this way, we hope that a thorough presentation of dataset shift will help to understand possible difference between source and target. Probably, with the help of dataset shift, we can possibly specify the positive transfer conditions and build other new transfer leaning approaches accordingly.

We start the presentation by covariate shift, the simplest model ; then, other models are introduced, from the model the most related to covariate shift to less related models.

– **Covariate Shift**

Given data distributed according to $p(x, y) = p(y|x)p(x)$, covariate shift designates the case where only $p(x,\ x \in \mathcal{S})$ is different from $p(x,\ x \in \mathcal{T})$ while $p(y|x)$ remains the same. Figure 3.1 shows an example of covariate shift.

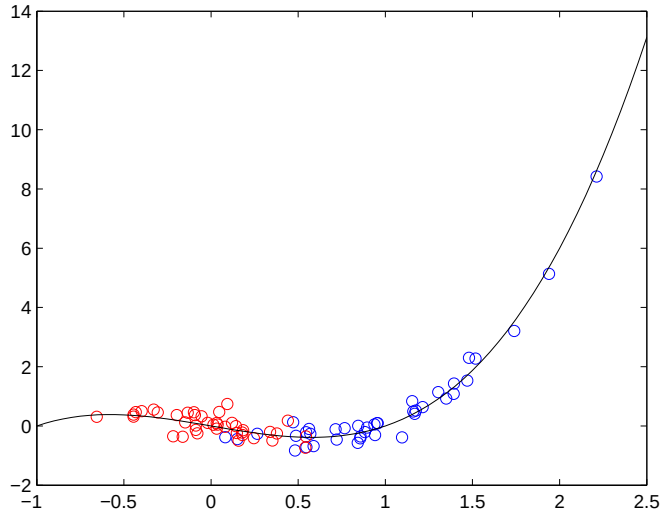


Figure 3.1 – An illustration of covariate shift (from [124]). Red circles are source data and blue circles are target data. All observations are generated according to $y = -x + x^3$ with 0.3 standard deviation of Gaussian noise.

For covariate shift, the prediction of y is dependent on x and the dependence does not vary between source and target. Therefore, finding $p(x,\ x \in \mathcal{T})$ is equivalent to finding the generation model $p_T(x, y,\ x \in \mathcal{T})$ with conditional probability $(p(y|x))$ known.

– **Prior Probability Shift**

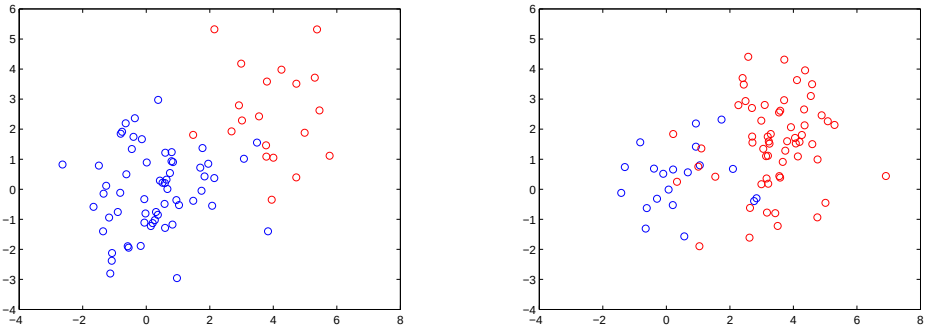
Given data distribution $p(x, y) = p(x|y)p(y)$, prior probability shift designates the case when $p(y)$ changes between source and target while $p(x|y)$ remains the same. In this case, the Bayes rule is applied to infer the prediction :

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\sum_{y \in \mathcal{Y}} p(x|y)p(y)}$$

Figure 3.2 illustrates the prior probability shift in 2D.

– **Sample Selection Bias**

Figure 3.4 gives an illustration of sample selection bias. Sample selection bias occurs when the source data cannot represent the target data, because the selection of every observation is dependent on the labels. This is a common situation in survey design : certain groups of people are very active in participating into the survey investigation while some other



(a) (b)

Figure 3.2 – Illustration of Prior Probability Shift. Figure 3.2(a) represents source and Figure 3.2(b) represents target. Different colors of points represent their labels. We can see that the distribution of x, $x \in \mathcal{S} \cup \mathcal{T}$ is dependent on prior probabilities (denoted as $p(y)$). However, given y, the distributions of x are similar.

groups refuse the participation ; thus, the result of survey is biased and influenced by the group of people.

– **Imbalanced Data**

The Imbalanced Data designates cases in multi-class learning where one (or several) class is rare compared to other class(es). Imbalanced Data can be considered as a special case of sample selection bias. Thus, one solution is to weight the rare observations to be more important than the common class observations or to reject common class observations during the learning process. But different from sample selection bias, the bias of imbalanced target data is only dependent on the class label.

– **Domain Shift**

Domain Shift designates the change of measurements or methods of description. It assumes some underlying unchanged latent representation shared by source and target data. Domain shift motivates many latent common factor based transfer learning, including heterogeneous transfer learning.

– **Source Component Shift**

Source Component Shift designates the case where domains can be represented by a combination of weighted different *sources*[5]. Both source and target come from such origins and as the proportion of *sources*[5] changes from source to target, source and target differ from each other.

Three practical cases of source component shift is proposed in [103], Chapter 1 :

5. Here, the source denotes the origin of the data

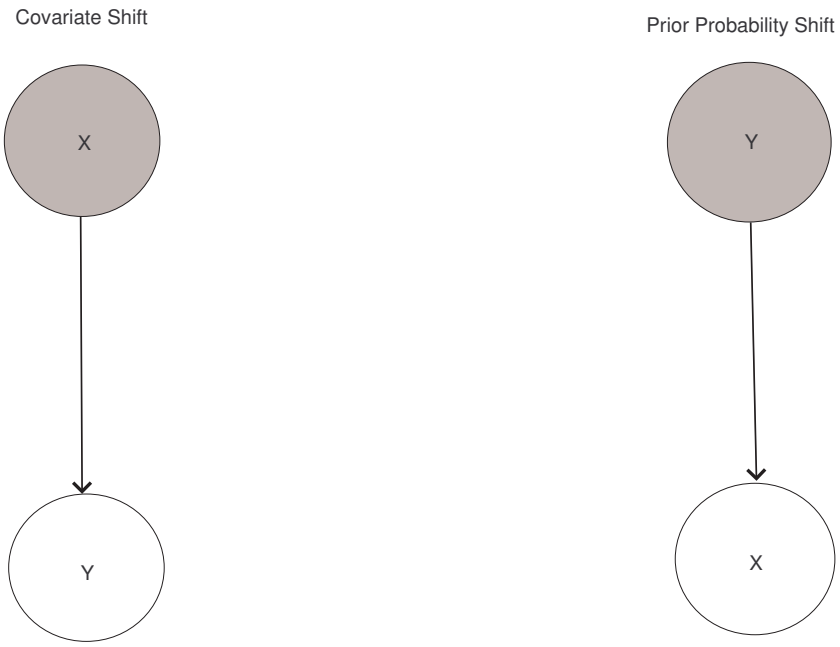


Figure 3.3 – Covariate Shift and Prior Probability Shift. For covariate shift, the shift in x causes the domain shift and y of target changes accordingly; for prior probability shift, it inverses the role of x and y : it is the change on prior of y that shifts the domain so that x of target changes accordingly.

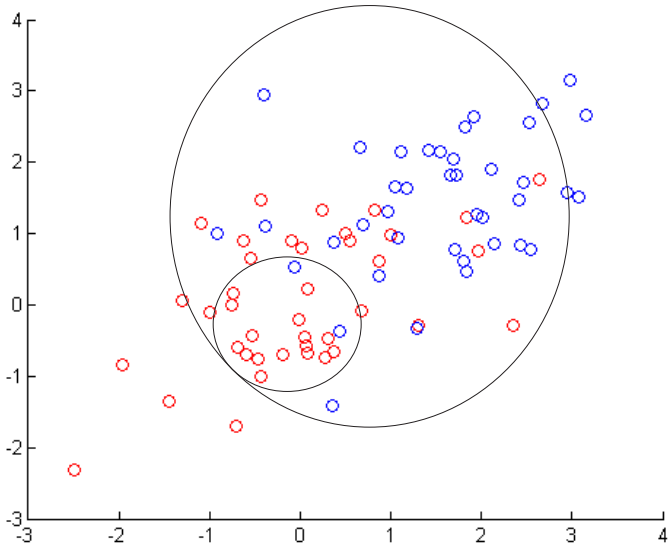
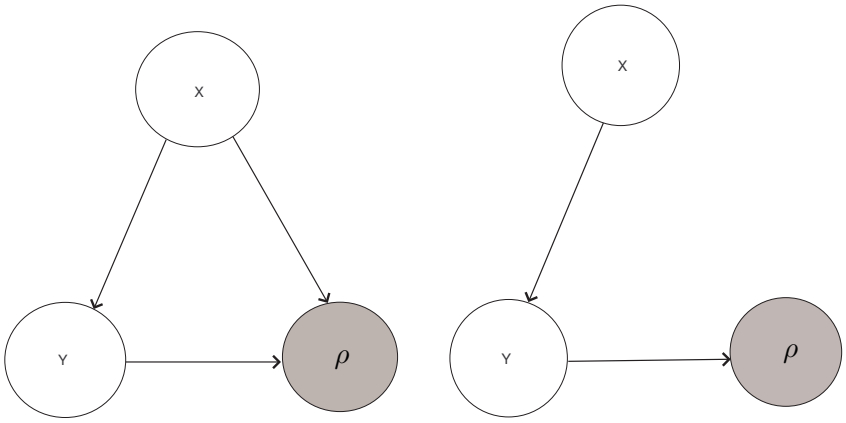


Figure 3.4 – An illustration of Sample Selection Bias. Red circles represent the source while blue circles represent the target. Obviously, if we select data inside the small circle, the selected data cannot well represent the whole data, neither the target data.

- *samples that could come from one of a number of subpopulations, between which the*



(a) Sample Selection Bias. (b) Imbalanced Data.

Figure 3.5 – Comparison of Sample Selection Bias and Imbalanced Data. In Figure 3.5(a), ρ represents the selection process : $\rho = 1$ takes into account the observation ; $\rho = 0$ rejects the observation. For sample selection bias, both domain information and label information influence the selection process : ρ is dependent on both x and y. If there is no dependence between ρ and y, it is the case of covariate shift ; in Figure 3.5(b), ρ has the same meaning as in Figure 3.5(a). Here, ρ is only dependent on y.

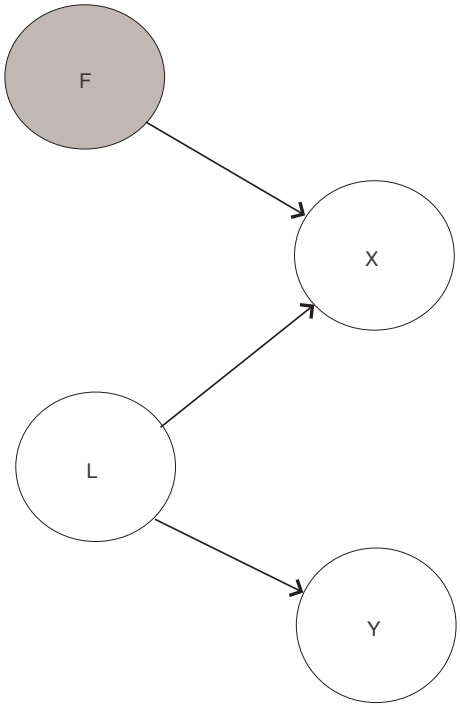


Figure 3.6 – Domain Shift. F designates some transformation from source to target : $x_t = f(L),\ f \in F$, where L designates some latent factors that are shared between source and target. $p(y|L)$ remains unchanged.

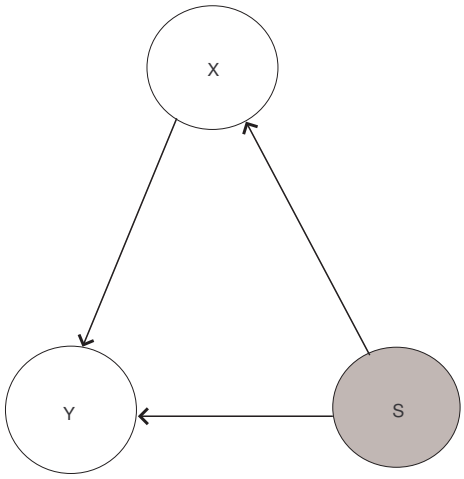


Figure 3.7 – Source Component Shift. S represents the origins of data. With the change of proportions of S, dataset shift occurs. Therefore, S influences both the domains and labels.

quantity to be predicted may vary ;

- *samples chosen are subject to factors that are not fully controlled for, and that could change in different scenario ;*
- *targets that are aggregated values averaged over a potentially varying population.*

Figure 3.3, Figure 3.5(a), Figure 3.5(b), Figure 3.6 and Figure 3.7 illustrates the dependencies among data factors for different types of dataset shifts (from [103]).

This thesis is a contribution to homogeneous transductive transfer learning. Covariate shift is one of the most important cases in homogeneous transductive transfer learning. Hereafter, we focus on covariate shift (Section 3.3), based on which the relaxed covariate shift approaches are proposed (in Section 3.4).

3.3 Overview of Covariate Shift

In this section, we focus on covariate shift and present a brief review of covariate shift. For sample selection bias, readers can refer to [167]. Approaches for other dataset shifts can be found in [170].

One well-known direct type of covariate shift solutions is importance sampling : source data are weighted to align target data distribution ; thus weighted source data can be considered as effective training data for target. How to find the weights is an important issue in importance sampling approaches. As presented in Section 2.3.2, reasoning from the shift of empirical risks, we can attribute the weight as $\frac{P_T(x_i)}{P_S(x_i)}\Big\|_{x_i \in \mathcal{D}}$ where $\mathcal{D}$ is the common support of source and target. For example in [124], the log-likelihood function (empirical risk) is weighted by $\frac{P_T(x_i)}{P_S(x_i)}\Big\|_{x_i \in \mathcal{D}}$; theoretical analyses show that when there is a sufficiently large number of data, the model found by weighted log-likelihood approach tends to be optimal.

3.3.1 Covariate shift transfer learning with Similarity Criteria integrated

With the same principle, Huang et al. ([62]) introduce importance sampling into other loss functions, for example SVM and penalized logistic regression. For SVM in binary classification, we have :

$$\begin{aligned}
&\min_{w,\epsilon} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n_s} \beta_i \epsilon_i \\
&\text{s.t. } y_i(< w, \phi(x_i) > +b) \geq 1 - \epsilon_i,\ \forall i = 1, \ldots, n_s \\
&\quad \epsilon_i \geq 0,\ \forall i = 1, \ldots, n_s
\end{aligned}$$

where $\beta_i = \frac{P_T(x_i)}{P_S(x_i)}$ and the optimization process remains similar. But instead of estimating directly $P_T(.)$ and $P_S(.)$, Huang et al. estimates β_i with the help of Maximum Mean Discrepancy :

$$\begin{aligned}
&\min_{\beta_i} || \frac{1}{n_s}\sum_{i=1}^{n_s} \beta_i \phi(x_s^i) - \frac{1}{n_t}\sum_{i=1}^{n_t} \phi(x_t^i) || \\
&= \frac{1}{n_s^2}\beta^T K_{SS}\beta - \frac{2}{n_s n_t}\mathrm{K}_{.S}\beta + \text{const} \\
&\text{s.t. constraints on } \beta
\end{aligned}$$

where $K_{SS}(i,j) =< \phi(x_s^i), \phi(x_s^j) >$ and $\mathrm{K}_{.S}(j) = \sum_{i=1}^{n_t} < \phi(x_t^i), \phi(x_s^j) >$. The above problem is a quadratic optimization problem w.r.t β. Thus, as $K_{SS} \succeq 0$, there is a unique solution for β. More details can be found in Section 4.3.1.

Instead of the Maximum Mean Discrepancy, the Kullback-Leibler divergence can also be applied to estimate the weight of importance sampling ([132] and [140]). Let $\hat{w}(x)$ be the weight $P_T(x) = \hat{w}(x)P_S(x)$, then

$$\begin{aligned}
KL(P_T(x)||\hat{P}_T(x)) &= \int_{x\in\mathcal{D}} P_T(x) log(\frac{P_T(x)}{\hat{w}(x)P_S(x)})dx \\
&= \int_{x\in\mathcal{D}} P_T(x) log(\frac{P_T(x)}{P_S(x)})dx - \int_{x\in\mathcal{D}} P_T(x)\log(\hat{w}(x))dx
\end{aligned}$$

We can use a linear model to express $\hat{w}(x)$: $\hat{w}(x) = \sum_{i=1}^{l} \alpha_i \varphi_i(x)$ where $\varphi_i(x)$ is the ith basis function and α_i is the parameter to be found. Then based on the above KL-divergence, the objective function is :

$$\begin{aligned}
&\min_{\alpha} KL(P_T(x)||\hat{P}_T(x)) \\
&\Leftrightarrow \\
&\min_{\alpha} -\int_{x\in\mathcal{D}} P_T(x)\log(\hat{w}(x))dx \\
&\approx \\
&\max_{\alpha} \frac{1}{n_t}\sum_{i=1}^{n_t} log(\hat{w}(x_t^i)) = \max_{\alpha} \frac{1}{n_t}\sum_{i=1}^{n_t} \log(\sum_{j=1}^{l} \alpha_j \varphi_j(x_t^i))
\end{aligned} \tag{3.3}$$

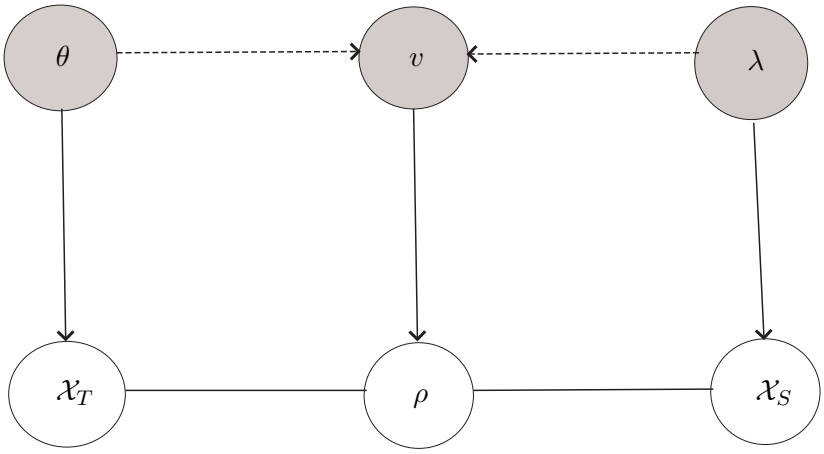


Figure 3.8 – Discriminative Learning, from [10].

Moreover $\alpha_j \geq 0,\ \forall j = 1, \ldots, l$ and

$$\begin{aligned} 1 &= \int_{x \in \mathcal{D}} \hat{P}_T(x)dx = \int_{x \in \mathcal{D}} \hat{w}(x) P_S(x)dx \\ &\approx \frac{1}{n_s} \sum_{i=1}^{n_s} \hat{w}(x_s) = \frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{j=1}^{l} \alpha_j \varphi_j(x_s^i) \end{aligned} \tag{3.4}$$

With Eq 3.4 and the constraints on α_j incorporated into problem 3.3, we have

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n_t} \log(\sum_{j=1}^{l} \alpha_j \varphi_j(x_t^i)) \\ &\text{s.t.} \sum_{i=1}^{n_s} \sum_{j=1}^{l} \alpha_j \varphi_j(x_s^i) = n_s \text{ and } \alpha_j \geq 0,\ \forall j = 1, \ldots, l \end{aligned}$$

which is a convex optimization problem and a unique solution exists.

[132] further proposes a strategy of selecting models (the basis formed by $\varphi_j(.),\ \forall j = 1, \ldots, l$) by likelihood cross validation and chooses the model with maximum likelihood.

[140] further improves [132] in efficiency.

3.3.2 Covariate shift transfer learning with Sample Selection Strategy

[10] proposes a totally different weighting strategy to solve covariate shift : selecting the most useful source observations to help the target learning process and the probability of selecting an observation given domain-dependent parameters is proportional to the importance sampling weights given parameters (Eq 3.5).

As shown in Figure 3.8, the sample selection parameter ρ relates source and target data : $p(\rho = 1|x, \theta, \lambda)$ represents the probability that x is from the target data and $p(\rho = 0|x, \theta, \lambda)$

represents the probability that x is from source data.

$$p(\rho = 1|x, \theta, \lambda) \propto \frac{p(x|\lambda)}{p(x|\theta)} \tag{3.5}$$

Proof

$$\begin{aligned}
\frac{p(x|\theta)}{p(x|\lambda)} &= \frac{p(\rho = 1|\theta, \lambda)}{p(\rho = 0|\theta, \lambda)} \frac{p(\rho = 0|\theta, \lambda)}{p(\rho = 1|\theta, \lambda)} \frac{p(x|\theta)}{p(x|\lambda)} \\
&= \frac{p(\rho = 1|\theta, \lambda)}{p(\rho = 0|\theta, \lambda)} (1 + \frac{p(\rho = 0|\theta, \lambda)}{p(\rho = 1|\theta, \lambda)} \frac{p(x|\theta)}{p(x|\lambda)} - 1) \\
&= \frac{p(\rho = 1|\theta, \lambda)}{p(\rho = 0|\theta, \lambda)} (\frac{p(\rho = 1|\theta, \lambda)p(x|\lambda) + p(\rho = 0|\theta, \lambda)p(x|\theta)}{p(\rho = 1|\theta, \lambda)p(x|\lambda)} - 1) \\
&= \frac{p(\rho = 1|\theta, \lambda)}{p(\rho = 0|\theta, \lambda)} (\frac{p(\rho = 1|x, \theta, \lambda)p(x|\theta)p(x|\lambda) + p(\rho = 0|x, \theta, \lambda)p(x|\lambda)p(x|\theta)}{p(\rho = 1|x, \theta, \lambda)p(x|\theta)p(x|\lambda)} - 1) \\
&= \frac{p(\rho = 1|\theta, \lambda)}{p(\rho = 0|\theta, \lambda)} (\frac{1}{p(\rho = 1|x, \theta, \lambda)} - 1)
\end{aligned}$$

In other words, $\frac{p(x|\lambda)}{p(x|\theta)}$ indicates the frequency of an observation $x, x \in \mathcal{X}_S$ that performs like target data.

The objective of the approach is to find parameters that maximize $p(v, w|\mathcal{X}_S, \mathcal{X}_T)$, where w is the classification parameter and v is the parameter that governs ρ.

3.4 Relaxed Covariate Shift

For most of the transfer learning cases, covariate shift conditions are not always satisfied. Thus, other importance sampling based methods have been proposed under relaxed covariate shift conditions to achieve better performances. We first introduce two existing approaches, although they belong to inductive transfer learning; then we propose our transductive relaxed covariate shift.

3.4.1 Inductive Relaxed Covariate Shift

In [178], a transfer cross validation is proposed :

$$f^* = \arg\max_f \frac{1}{k} \sum_{i=1}^{k} \frac{1}{n_k} \sum_{(x,y)\in\mathcal{D}_i} \frac{P_T(x)}{P_S(x)} |P_T(y|x) - P(y|x, f)|$$

where $P(y|x, f)$ denotes the conditional probability given a classifer f. This approach is similar to importance weighted cross validation in [131]. But different from [131], [178] takes into consideration the misalignment of conditional probabilities. By taking advantage of a reverse validation strategy, the approximated conditional probability approaches the target conditional probability ; the weight $\frac{P_T(x)}{P_S(x)}$ is estimated similarly as in [62] by using MMD.

[49] follows similar principle but extends the classification task to regression task. The objective of [49] is transfer learning based regression :

– Step 1 - Importance Sampling of Prediction Function

$$\begin{aligned} y_t &= \arg\max_y (P_T(y|x_t)P_T(x_t)) \\ &= \arg\max_y (\frac{P_T(y|x_t)P_T(x_t)}{P_S(x_t,y)} P_S(y|x_t)P_S(x_t)) \\ &= \arg\max_y (w(x_t,y)P_S(y|x_t)P_S(x_t)) \end{aligned}$$

– Step 2(a) - Regression by Least Square Error

$$\begin{aligned} &\min ||Y_t - \hat{Y}_t||^2 \\ &\iff \\ &\min_{\hat{w}} \sum_{i=1}^{n_t^l} (y_t^i - \arg\max_y (\hat{w}(x_t^i,y)P_S(y|x_t^i)P_S(x_t^i)))^2 \end{aligned}$$

As $P_S(x_t^i)$ is independent of y, thus the above objective function can be reduced to :

$$\min_{\hat{w}} \sum_{i=1}^{n_t^l} (y_t^i - \arg\max_y (\hat{w}(x_t^i,y)P_S(y|x_t^i)))^2$$

– Step 2(b) - KL-divergence based Weight Estimation

$$\arg\min_{\hat{w}} KL(P_T(x,y)||\hat{w}(x,y)P_S(x,y))$$

Similarly to Problem 3.3, $\hat{w}$ is obtained.

3.4.2 Transductive Relaxed Covariate Shift (MLRCV)

Different from the two former approaches, we remain in the transductive transfer learning context and assume a relaxed version of covariate shift (a reminder of Assumption 3.2) :

$$\exists A, b : p_S(y|x, x \in \mathcal{S}) = p_T(y|Ax + b, x \in \mathcal{T})$$

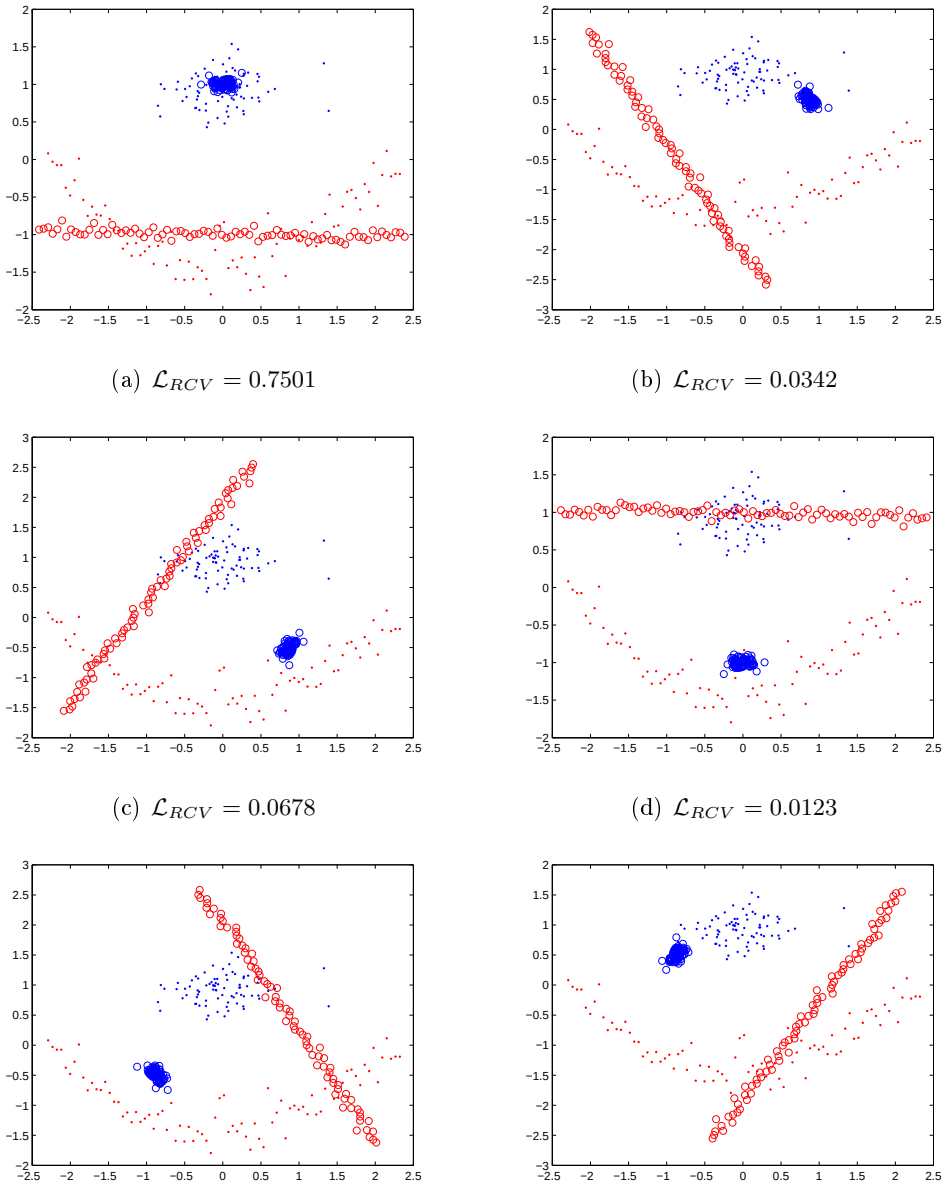
with A and b being parameters.

We propose a parametric maximum likelihood relaxed covariate shift approach (MLRCV) :

In the idealized case, after transformation A and b, the marginal probabilities of source and target are aligned ; then, with Assumption 3.2 satisfied, the distributions of source and target data are expected to become similar ; therefore, applying A and b to target data, we can use source classifier to well classify the transformed target data ($Ax_t^i + b, \forall x_t^i \in \mathcal{X}_T$). To find the best model (best parameters A and b), we apply maximum likelihood on iid x_t^i, $\forall\ x_t^i \in X_T$:

$$\max_{A,b} \mathcal{L}_{RCV} = \max_{A,b} \prod_{x_t^i \in \mathcal{D}_T} p_S(Ax_t^i + b|A, b)$$

We can estimate $p_S(.)$ by Kernel Density Estimation (KDE, Section 1.4.5). However, the optimization is non-convex. How to find reliably and efficiently the best A and b remains under research. Still, we validate our idea in Section 3.5 : in a synthetic case, with proper A and b, the objective function is maximized.



(a) $\mathcal{L}_{RCV} = 0.7501$ (b) $\mathcal{L}_{RCV} = 0.0342$

(c) $\mathcal{L}_{RCV} = 0.0678$ (d) $\mathcal{L}_{RCV} = 0.0123$

(e) $\mathcal{L}_{RCV} = 0.0778$ (f) $\mathcal{L}_{RCV} = 0.0428$

Figure 3.9 – Banana-orange datasets. Every subfigure represents a possible alignment with rotation w.r.t the same mean by 60° successively. The circles represent the target data and their labels are supposed unknown; the points represent the source data and their colors represent two different classes.

3.5 Simulations and Analysis

As shown in Figure 3.9, we apply the proposed MLRCV and observe that the likelihood is maximized in Figure 3.9(a). When there is more overlapping between source and target, the value of likelihood is higher. To make the likelihood value more sensible to distribution changes, we propose to normalize the likelihood by $\frac{n_t}{V}$. We have also varied the rotation and dilatation degree ; MLRCV can always find the ideal case with maximum likelihood value. In similar synthetic datasets, there is usually a large gap between the optimal case and the others.

3.6 Conclusion

In this chapter, we specified common types of dataset shifts (in Section 3.2.1) and related transfer learning (specifically covariate shift) with dataset shift. Then, we focused on covariate shift problems and summarized the principal approaches for this type of adaptation (in Section 3.3). However, for many applications, the assumption of covariate shift is too restraint. Therefore, relaxed covariate shift approaches are proposed (in Section 3.4) : previous works generally contribute to inductive transfer learning and we propose a transductive transfer learning (MLRCV) approach with relaxed assumption (Assumption 3.2). Although the optimization for our approach is non-convex, we have shown the validity of our approach on synthetic dataset (in Section 3.5). Efficient optimization strategy is under research and large-scale adaptation is to be made.

Even if MLRCV can be considered as an extended version of covariate shift, its assumption is limited to some linear transformation for A and b in Assumption 3.2. In the following chapters, we will further relax Assumption 3.2 and the proposed approaches have more generalization ability (also more simple).

Chapter 4

Domain Adaptation by SVM subject to a MMD-like constraint

Contents

4.1 Introduction

For real transductive transfer learning contexts, the assumption of covariate shift is often violated. Aligning source and target marginal distributions in the original space is no longer enough. Therefore, there have been literatures that transfer both marginal distribution and conditional distribution of source to target. That is also the objective of domain adaptation. For challenging domain adaptation tasks, many authors have taken advantage of few labeled target data so that with measurable conditional probabilities of target ($p_T(y, y \in \mathcal{T}|x, x \in \mathcal{T})$), they try to correct the difference between $p_T(y, y \in \mathcal{T}|x, x \in \mathcal{T})$ and $p_S(y, y \in \mathcal{S}|x, x \in \mathcal{S})$ as well as the difference between $p_T(x, x \in \mathcal{T})$ and $p_S(x, x \in \mathcal{S})$ ([155], [177], [178]). However, few labeled target data is not reliably representative for the whole distributions ; and for many cases, there is no available label information for target data. In order to align the conditional probabilities, other authors have used the clustering principle to generate pseudo labels to get reliable $p_T(y, y \in \mathcal{T}|x, x \in \mathcal{T})$ ([22], [61], [75]).

In this chapter, our context is domain adaptation with no labeled target data with the assumption that the conditional probabilities of labels of source and target data can become similar in a RKHS :

$$\exists g(.) : \mathbb{R}^m \to \mathbb{R}^n \mid p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T}) \tag{4.1}$$

where $g(.)$ is a smooth transformation function (linear or nonlinear).

It is a further relaxation of the assumption presented in Section 3.4.2. With this assumption, if the marginal distributions of source and target (after transformation $g(.)$) are aligned, source classifier is expected to well classify target data. Simulations below and experimental results in Section 5.6 show that the assumption is reasonable. Contrarily, in cases like the example presented in Figure 4.1 (where Assumption 4.1 is violated), even after linear or nonlinear transformations, the source classifier is not expected to perform well on target data.

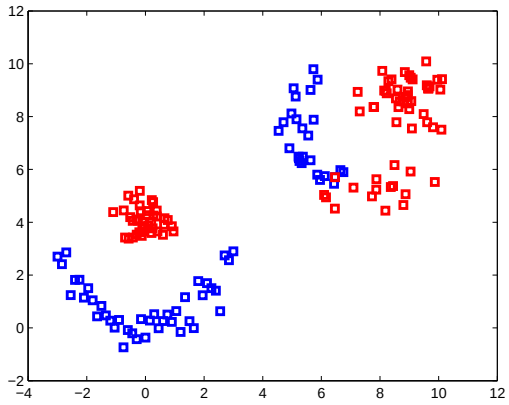


Figure 4.1 – An example that violates Assumption 4.1. The data on lower left represents source while the data on upper right represents the target. Even if we can easily find a transformation that leads to the superposition of source and target data, the classifier trained on source will not correctly classify target data. The colors designate labels of $+1$ and -1.

Based on the above assumption, we propose a new domain adaptation approach which combines SVM and MMD (Section 4.3.3). Overviews of SVM and MMD used in transfer learning are given in Section 4.2. To the best of our knowledge, both orientations have been widely studied. In general, to adapt to transfer learning context, standard SVM is modified to integrate the transfer ability. Sometimes, standard SVM is adapted by using MMD as a regularization term (Section 4.3.2). Different from those previous works, we utilize a MMD-like constraint to limit the RKHS to a common subspace where locates the SVM hyperplane. In this subspace, source and target data are expected to be similar so that the SVM hyperplane trained on source data can be applied directly to target data. Simulations on synthetic data are presented in Section 4.4, the results of which have shown that our approach is efficient and comparable or even better than its regularization counterpart ([102]).

4.2 Background

4.2.1 SVM based Transfer Learning

SVM has been a very popular supervised classification method (refer to Section 1.4.2). However, in general, standard SVM performs badly in solving transfer learning problems, because target data is scarcely labeled or unlabeled and source and target data follow different distributions. Many literatures are contributed to adapting standard SVM to fit the transfer learning context.

In [171], it is supposed that source and target data share a common parameter w_{common} and differ in domain specific parameters w_{source} and w_{target} :

$$\begin{aligned} \min_{w_{\{.\}},\epsilon_t^i,\epsilon_s^j,b} \quad & \frac{1}{2}||w_{common}||^2 + C_1||w_{source}||^2 + C_2||w_{target}||^2 + C\sum_{i=1}^{n}\epsilon_t^i + C\sum_{j=1}^{m}\epsilon_s^j \\ \text{s.t.} \quad & y_t^i((w_{common} + w_{target})x_t^i + b) > 1 - \epsilon_t^i \\ & y_s^j((w_{common} + w_{source})x_s^j + b) > 1 - \epsilon_s^j \\ & \epsilon_t^i \geq 0 \quad \text{and} \quad \epsilon_s^j \geq 0 \end{aligned}$$

where C_1, C_2 are trade-off parameters to control the preference of different domains, usually $C_2 > C_1$. Optimizing w.r.t these three parameters $(w_{common}, w_{source}, w_{target})$ leads to good performance. However, few labeled target is necessary and three trade-off parameters (C_1, C_2, C) are to be tuned.

Similar idea can be found in [162] : the target decision function is supposed to be the source decision function biased by $w'x_i$ and then, optimizing w.r.t w' can result in good target decisions. Like [171], labeled target data is important to the training problem.

Other approaches have been dedicated to generating pseudo-labels for target data, from which SVM is applicable. Transductive SVM ([17]) and DASVM ([18]) are the most famous approaches. With labeled source data, they iteratively label target data (pseudo label) and adjust the target classifier with more and more reliable target pseudo-labels. The generation of pseudo-labels is crucial and this instance-based strategy may be impaired by insufficient number of target instances.

There are also literatures that reweight the penalty term of standard SVM so that labeled target data is considered more important than labeled source data. However, labeled target data is necessary and the trade-off between source data error and target data error cannot be simply found. [23], [135], [143], [149], etc.

Another very common strategy for SVM based transfer learning is to regularize source SVM problem with a similarity term that takes into consideration the transfer between source and target. There have been a variety of similarity measures : manifold regularization ([7]), KL-divergence, Bregman divergence ([125]), Maximum Mean Discrepancy (MMD in Section 1.4.4), etc.

4.2.2 MMD based Transfer Learning

Maximum Mean Discrepancy (MMD) has been widely used as the similarity measure. Its estimation is simple and non-parametric. For most of the transfer learning cases, MMD aims at controlling the transfer.

In [142], MMD serves as a feature pre-selection criterion. It filters the dissimilar features of source data and leaves the most related features (between source and target) to train a classifier that also performs relatively well on target. In [93], MMD is used to extract useful features, based on which transfer learning is expected to be effective. Similar ideas can be applied in multi-source domains transfer learning, MMD can be used to pre-select the most related source(s).

Then, more recently, MMD has been developed in more flexible form.

In [5], MMD has been applied to transformed data :

$$\arg\min_{W} ||\frac{1}{n}\sum_{i=1}^{n}\phi(W^T x_s^i) - \frac{1}{m}\sum_{i=1}^{m}\phi(W^T x_t^j)||^2 \quad \text{s.t.} \quad W^T W = I$$

However, the optimization of the above problem is non-convex and authors have optimized on a Grassmann Manifold.

In [51], the landmarks are selected by weighted MMD :

$$\begin{aligned} \min_{\alpha} ||\frac{1}{\sum_m \alpha_m}\sum_m \alpha_m \phi(x_m) - \frac{1}{N}\sum_n \phi(x_n)||^2_{\mathcal{H}} \\ \text{s.t.} \frac{1}{\sum_m \alpha_m}\sum_m \alpha_m y_m^c = \frac{1}{M}\sum_m y_m^c \\ \alpha = \{\alpha_m \in \{0,1\}, m = 1, \ldots, M\} \end{aligned}$$

where y_m^c equals 1 if $y_m = c$, otherwise y_m^c equals 0. In this way, MMD is calculated only on the subregions - useful landmarks that keep the main structure of the whole data.

Gong et al. further extend the above idea to clustering and selecting the most distinctive

data ([52]) :

$$\min_{\beta^k,\beta^{k'}} ||\frac{1}{M_k}\sum_m \phi(x_m)\beta_m^k - \frac{1}{M_{k'}}\sum_m \phi(x_m)\beta_m^{k'}||_{\mathcal{H}}^2$$
$$\text{s.t. } \beta^{\{k|k'\}} \in \{0,1\}$$
$$\text{and } \frac{1}{M_k}\sum_m \beta_m^k y_m^c = \frac{1}{M}\sum_m y_m^c$$

In most literatures, MMD or modified MMD has been used as a regularizer. The transfer learning problem can be viewed as finding the trade-off between the classification performance on source data and the similarity between source and target. ([165], [105], [173], [92])

4.3 Domain Adaptation by SVM combined with MMD

As an intersection of the literatures presented in Section 4.2, domain adaptation by SVM with MMD can find an optimal classification hyperplane while transferring knowledge from source(s) to target.

4.3.1 Kernel Mean Matching (KMM)

Kernel Mean Matching (KMM [62]) aims at correcting sample selection bias, assuming that the conditional probability of labels is the same between source and target data. At first, KMM finds the reweighting factor $\beta = \frac{P_t(.)}{P_s(.)}$ by MMD ; then, β is integrated into standard SVM to reweight the penalty :

– Step 1 : finding reweighting factor (β) by minimizing marginal distribution distance - MMD

$$\min_\beta ||\mu_t[\phi(.)] - E[\beta(x)\phi(x)|x \in \mathcal{S}]||_{\mathcal{H}}^2$$
$$\text{s.t. } \beta(x) \geq 0 \text{ and } E[\beta(x)|x \in \mathcal{S}] = 1$$
$$\text{where } \mu_t[\phi(.)] = E[\phi(x)|x \in \mathcal{T}]$$
$$\Longleftrightarrow$$
$$\min_\beta ||\frac{1}{n_t}\sum_{i=1}^{n_t}\phi(x_t^i) - \frac{1}{n_s}\sum_{i=1}^{n_s}\beta_i\phi(x_s^i)||_{\mathcal{H}}^2$$
$$\min_\beta \frac{1}{n_s^2}\beta^T K_{SS}\beta - \frac{2}{n_s n_t}k^T\beta + \text{const}$$
$$\text{s.t. } \beta_i \geq 0 \text{ and } |\frac{1}{n_s}\sum_{i=1}^{n_s}\beta_i - 1| \leq \epsilon$$
$$\text{where } K_{SS}(i,j) = k(x_s^i, x_s^j) \text{ and } k(i,j) = k(x_s^i, x_t^j)$$

We can find that the final optimization problem is quadratic and can be optimized by standard quadratic optimization tools.

– Step 2 : β weighted soft margin SVM

$$\min_{w,\xi,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n_s}\beta_i\xi_i$$
$$\text{s.t. } y_i(< w, \phi(x_i) > +b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

Now, β is considered as a parameter and the SVM problem is similar to the standard SVM. Lagrangian optimization can be applied and the final dual form (Annexe 1.1) is quadratic.

KMM takes advantage of importance sampling (Section 3.3) and the simple quadratic optimization ; however, its assumption is too limited for domain adaptation and the two-step optimization appears tedious compared to one-step optimization problem.

4.3.2 Domain adapatation by SVM with MMD as a regularization term (LM and ARSVM)

Large Margin Transductive Transfer Learning - SVM (LM)

In order to transfer knowledge at the same time as to learn a classifer, Large Margin Transductive Transfer Learning - SVM (LM [102]) is proposed. LM regularizes the standard SVM with a projected MMD :

$$\begin{aligned} &\min_{w,\xi,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n_s}\xi_i + \lambda Dist(P_s,P_t)^2 \\ &\text{s.t.} \quad y_i(< w, \phi(x_s^i) > + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \end{aligned} \tag{4.2}$$

where $Dist(P_s, P_t)$ is the projected MMD :

$$Dist(P_s,P_t)^2 = ||\frac{1}{n_s}\sum_{i=1}^{n_s} < w, \phi(x_s^i) > - \frac{1}{n_t}\sum_{j=1}^{n_t} < w, \phi(x_t^j) > ||^2$$

Applying the Representer Theorem (Section 1.3.6), $w = \sum_{i=1}^{ns+nt} \alpha_i \phi(X_i)$, where$X = X_s \cup X_t$, problem 4.2 can be transformed to a quadratic optimization problem w.r.t α (Annexe 1.2). Finding the optimum α can directly result in the SVM hyperplane.

However, the regularization parameter should be tuned in a delicate way ; when faced with large-scale data, generalized singular value decomposition is applied to avoid imprecise approximation of a high-dimensionality inverse matrix in final dual form (Annexe 1.2) ; moreover, LM keeps the assumption of KMM.

Adaptation Regularization-SVM (ARSVM)

In order to take into consideration of difference of conditional probabilities, ARSVM is proposed. The much more complex regularized domain adaptation problem can be formulated as

([74]) :

$$\begin{aligned}
&\arg\min_{f\in\mathcal{H}} \sum_{i=1}^{n} l(f(x_i), y_i) + \sigma||f||^2 + \lambda D_f(P_s, P_t, Q_s, Q_t) + \gamma M_f(P_s, P_t) \\
&D_f(P_s, P_t, Q_s, Q_t) = D_f(P_s, P_t) + \sum_{c=1}^{C} D_f^{(c)}(Q_s, Q_t) \\
&\text{where} \quad D_f(P_s, P_t) = ||\frac{1}{n_s}\sum_{i=1}^{n_s} f(x_s^i) - \frac{1}{n_t}\sum_{j=1}^{n_t} f(x_t^j)||^2 \\
&\qquad\qquad = ||\frac{1}{n_s}\sum_{i=1}^{n_s} < w, \phi(x_s^i) > - \frac{1}{n_t}\sum_{j=1}^{n_t} < w, \phi(x_t^j) > ||^2 \\
&\qquad D_f^{(c)}(Q_s, Q_t) = ||\frac{1}{n_s^{(c)}}\sum_{i=1}^{n_s^{(c)}} f(x_{s(c)}^i) - \frac{1}{n_t^{(c)}}\sum_{j=1}^{n_t^{(c)}} f(x_{t(c)}^j)||^2
\end{aligned} \tag{4.3}$$

$D_f(P_s, P_t)$ is the squared projected MMD as in [102] and $D_f^{(c)}(Q_s, Q_t)$ is the squared projected MMD taking into consideration the label information. $D_f^{(c)}(Q_s, Q_t)$ minimizes the distribution distance of source and target data with the same label, so that the conditional probabilities $Q_s(x_s|y_s)$ and $Q_t(x_t|y_t)$ are rendered similar. From problem 4.3, an extra regularization term can be added, for example $M_f(P_s, P_t)$. In [74],

$$M_f(P_s, P_t) = \sum_{i,j=1}^{n_s+n_t} (f(x_i) - f(x_j))^2 W_{ij}$$

where W is the graph affinity matrix that measures the geometric similarity. By [7], if the marginal distributions ($P_s(x_s)$ and $P_t(x_t)$) are similar in intrinsic geometry, then the conditional distributions ($Q_s(y_s|x_s)$ and $Q_t(y_t|x_t)$) are similar. $M_f(P_s, P_t)$ is the manifold regularization that propagates the intrinsic manifold structure of the data.

However, whether the more we regularize, better the result we shall get, remains to be investigated. For [74], in the perfect alignment case, $P_s(y_s)$ should be similar to $P_t(y_t)$, which is not often the case, especially in dealing with imbalanced data transfer learning problems. Furthermore, during the optimization (see [74]), an inverse matrix is unavoidable, which may leads to imprecise optimization.

4.3.3 Domain adapatation by SVM subject to a MMD-like constraint (SVMMMD)

Instead of using MMD as a regularization term, we integrate MMD into the standard SVM problem as a constraint. The new model remains a quadratic optimization problem with no inverse matrix calculation.

Model

We also adopt the projected MMD proposed in [102]. The domain adaptation by SVM subject to a MMD-like constrint (SVMMMD) can be formulated as :

$$\begin{aligned} &\min_{w,\xi,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n_s}\xi_i \\ &\text{s.t.} \quad < w, \mu_{X_s} - \mu_{X_t} >_{\mathcal{H}} = 0 \\ &\quad y_i(< w, \phi(x_s^i) > + b) \geq 1 - \xi_i, \forall i = 1, \ldots, n_s \\ &\quad \xi_i \geq 0, \forall i = 1, \ldots, n_s \end{aligned} \tag{4.4}$$

By imposing $< w, \mu_{X_s} - \mu_{X_t} >_{\mathcal{H}} = 0$, where $\mu_{X_s} = \frac{1}{n_s}\sum_{i=1}^{n_s}\phi(x_s^i)$ and $\mu_{X_t} = \frac{1}{n_t}\sum_{j=1}^{n_t}\phi(x_t^j)$, we limit the search of SVM hyperplanes in a subspace of RKHS where $MMD = 0$. In this subspace, as their conditional probabilities become similar (Assumption 4.1), we suppose that by aligning source and target marginal distributions, the classification of target will succeed. Thus, a SVM classifier trained on source can perform well on target.

Comparaison with LM

Compared to LM (in Section 4.3.2), we use the projected MMD as a constraint instead of a regularization term. In this way, source and target data are projected into a common subspace of RKHS where MMD equals 0. In LM, there is no such geometric interpretation.

Moreover, there is no regularization trade-off parameter (λ in Equation 4.2) to be tuned. SVMMMD can be considered as the $\lambda \to \infty$ case of LM. In cases where λ has a finite value, the regularized problem may sacrifice the similarity to achieve a high classification performance on source data. By setting projected MMD as a constraint, we focus more on transfer.

Optimization

We solve the problem 4.4 by first applying the representer theorem to w :

$$w = \sum_{k=1}^{n_s}\beta_k^s\phi(x_s^k) + \sum_{l=1}^{n_t}\beta_l^t\phi(x_t^l)$$

Then, $< w, \mu_{X_s} - \mu_{X_t} >_{\mathcal{H}} = 0$ can be developed as :

$$\begin{aligned}
< w, \mu_{X_s} - \mu_{X_t} >_{\mathcal{H}} &= < \sum_{k=1}^{n_s} \beta_k^s \phi(x_s^k) + \sum_{l=1}^{n_t} \beta_l^t \phi(x_t^l), \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_s^i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_t^j) >_{\mathcal{H}} \\
&= \frac{1}{n_s} \sum_{k=1}^{n_s} \beta_k^s \sum_{i=1}^{n_s} < \phi(x_s^k), \phi(x_s^i) >_{\mathcal{H}} - \frac{1}{n_t} \sum_{k=1}^{n_s} \beta_k^s \sum_{j=1}^{n_t} < \phi(x_s^k), \phi(x_t^j) >_{\mathcal{H}} \\
&+ \frac{1}{n_s} \sum_{l=1}^{n_t} \beta_l^t \sum_{i=1}^{n_s} < \phi(x_t^l), \phi(x_s^i) >_{\mathcal{H}} - \frac{1}{n_t} \sum_{l=1}^{n_t} \beta_l^t \sum_{j=1}^{n_t} < \phi(x_t^l), \phi(x_t^j) >_{\mathcal{H}} \\
&= (\begin{bmatrix} K_{SS} & K_{TS} \\ K_{ST} & K_{TT} \end{bmatrix} [\underbrace{\frac{1}{n_s}, \ldots, \frac{1}{n_s}}_{n_s}, \underbrace{-\frac{1}{n_t}, \ldots, -\frac{1}{n_t}}_{n_t}]^T)^T [\beta^s, \beta^t]^T \\
&= (K\tilde{1})^T \beta
\end{aligned}$$

where $K_{SS} = < \phi(x_s^k), \phi(x_s^i) >_{\mathcal{H}}$, $K_{TS} = < \phi(x_t^j), \phi(x_s^k) >_{\mathcal{H}}$, $K_{ST} = < \phi(x_s^i), \phi(x_t^l) >_{\mathcal{H}}$, $K_{TT} = < \phi(x_t^j), \phi(x_t^l) >_{\mathcal{H}}$.

The $||w||^2$ in objective function of problem 4.4 can be replaced by :

$$\begin{aligned}
||w||^2 &= < \sum_{k=1}^{n_s} \beta_k^s \phi(x_s^k) + \sum_{l=1}^{n_t} \beta_l^t \phi(x_t^l), \sum_{k=1}^{n_s} \beta_k^s \phi(x_s^k) + \sum_{l=1}^{n_t} \beta_l^t \phi(x_t^l) >_{\mathcal{H}} \\
&= \sum_{k=1}^{n_s} \sum_{k'=1}^{n_s} \beta_k^s < \phi(x_s^k), \phi(x_s^{k'}) >_{\mathcal{H}} \beta_{k'}^s + 2 \sum_{k=1}^{n_s} \sum_{l=1}^{n_t} \beta_k^s < \phi(x_s^k), \phi(x_t^l) >_{\mathcal{H}} \beta_l^t \\
&+ \sum_{l=1}^{n_t} \sum_{l'=1}^{n_t} \beta_l^t < \phi(x_t^l), \phi(x_t^{l'}) >_{\mathcal{H}} \beta_{l'}^t \\
&= [\beta^s, \beta^t] \begin{bmatrix} K_{SS} & K_{TS} \\ K_{ST} & K_{TT} \end{bmatrix} [\beta^s, \beta^t]^T \\
&= \beta^T K \beta
\end{aligned}$$

Therefore, problem 4.4 becomes :

$$\begin{aligned}
&\min_{\beta, \xi, b} \frac{1}{2} \beta^T K \beta + C \sum_{i=1}^{n_s} \xi_i \\
&\text{s.t.} (K\tilde{1})^T \beta = 0 \\
&\quad y_i (\beta < \phi(X), \phi(x_s^i) > + b) \geq 1 - \xi_i, \forall i = 1, \ldots, n_s \\
&\quad \xi_i \geq 0, \forall i = 1, \ldots, n_s
\end{aligned} \tag{4.5}$$

where X represents the ensemble of X_s and X_t.

Applying Lagrangian optimization to problem 4.5 :

$$\mathcal{L} = \max_{\alpha, \mu, \eta} \min_{\beta, \xi, b} \frac{1}{2} \beta^T K \beta + C \sum_{i=1}^{n_s} \xi_i - \sum_{i=1}^{n_s} \alpha_i \xi_i - \sum_{i=1}^{n_s} \mu_i [y_i (\beta < \phi(X), \phi(x_s^i) > + b) - 1 + \xi_i] - \eta ((K\tilde{1})^T \beta)$$

with α, μ, η being Lagrangian Multipliers.

$$\frac{\partial \mathcal{L}}{\partial \beta} = K\beta - \sum_{i=1}^{n_s} \mu_i y_i < \phi(X), \phi(x_s^i) > -\eta K\tilde{1} = 0 \Leftrightarrow \beta = \sum_{i=1}^{n_s} K^{-1} < \phi(X), \phi(x_s^i) > \mu_i y_i + \eta\tilde{1}$$
$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \Leftrightarrow \alpha_i = C - \mu_i$$
$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{n_s} \mu_i y_i = 0$$

Replacing β, α_i using above equations, the dual form of problem 4.5 is :

$$\begin{aligned} \max_{\mu,\eta} \sum_{i=1}^{n_s} \mu_i - \frac{1}{2}(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T K^{-1} (\sum_{j=1}^{n_s} \mu_j y_j K_{j.}) - \frac{1}{2}\eta^2 \tilde{1}^T K \tilde{1} - \eta(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T \tilde{1} \\ \text{s.t.} \quad 0 \leq \mu_i \leq C \quad \text{and} \sum_{i=1}^{n_s} \mu_i y_i = 0 \end{aligned} \tag{4.6}$$

where $K_{.i} =< \phi(X), \phi(x_s^i) >$

In the above maximization problem, there are only two Lagrangian parameters left (μ and η). If we fix μ, we obtain :

$$\max_{\eta} -\frac{1}{2}\eta^2 \tilde{1}^T K \tilde{1} - \eta(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T \tilde{1}$$

We have to maximize a quadratic form w.r.t η with Hessian matrix $-\tilde{1}^T K \tilde{1} < 0$. The optimum η can be expressed as $\eta = -\frac{(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T \tilde{1}}{\tilde{1}^T K \tilde{1}}$. Then we replace η in problem 4.6 and the final dual form is :

$$\begin{aligned} \max_{\mu_i} \sum_{i=1}^{n_s} \mu_i - \frac{1}{2}(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T (K^{-1} - \frac{\tilde{1}\tilde{1}^T}{\tilde{1}^T K \tilde{1}})(\sum_{j=1}^{n_s} \mu_j y_j K_{.j}) \\ \text{s.t.} \quad 0 \leq \mu_i \leq C \quad \text{and} \quad \sum_{i=1}^{n_s} \mu_i y_i = 0 \end{aligned}$$

Let γ_i denote $\mu_i y_i$, the previous problem becomes :

$$\begin{aligned} \max_{\gamma} \gamma^T Y - \frac{1}{2}\gamma^T (K_{SS} - \frac{K_{S.}\tilde{1}\tilde{1}^T K_{S.}^T}{\tilde{1}^T K \tilde{1}})\gamma \\ \text{s.t.} \quad \sum_{i=1}^{n_s} \gamma_i = 0 \quad \text{and} \quad min(0, Cy_i) \leq \gamma_i \leq max(0, Cy_i) \end{aligned}$$

where $K_{S.} = \sum_{i=1}^{n_s} K_{i.}$.

Demonstration of Positive Semi-Definitivenss : The matrix $K_{SS} - \frac{K_{S.}\tilde{1}\tilde{1}^T K_{S.}^T}{\tilde{1}^T K \tilde{1}}$ can be considered as the inner product of source data in a subspace orthogonal to w. As stated in [97], if $\mathcal{H}$ is a RKHS on X and $\mathcal{H}_0 \subseteq \mathcal{H}$ is a closed subspace, then $\mathcal{H}_0$ is also a RKHS on X. Therefore, let $K_{new} = K_{SS} - \frac{K_{S.}\tilde{1}\tilde{1}^T K_{S.}^T}{\tilde{1}^T K \tilde{1}}$, then K_{new} is the new Gram matrix corresponding to the projected kernel and it is positive semi-definite.

Perspectives

Although in Section 4.4 and Section 5.6, SVMMMD is proved efficient, conditional probabilities after kernel transformation are not taken into consideration. Perhaps, extra term(s) that aligns conditional probabilities $P_s(y_s|f(x), x \in \mathcal{S})$ and $P_t(y_t|g(x), x \in \mathcal{T})$ can be incorporated, either as a constraint or as a regularization term. Moreover, theoretical error bound is to be developed.

4.4 Simulations and Analysis

4.4.1 Illustration of Principles of SVMMMD

We first illustrate the principal idea of SVMMMD by using a linear kernel.

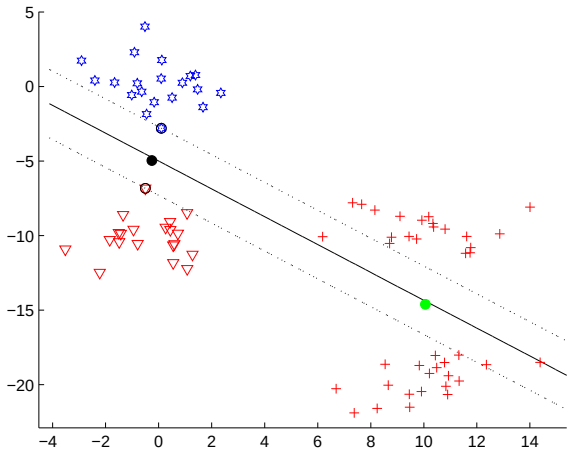


Figure 4.2 – Linearly separable data set using the linear kernel (triangles and stars represent the labeled source data, while "+" symbols represent the unlabeled target data)

In Figure 4.2, the two circles ($\vec{m}_s$ and $\vec{m}_t$) represent the mean of source and target, respectively. As can be seen, the normal to the obtained discriminant function is orthogonal to $\vec{m}_s - \vec{m}_t$, which coincides with the theoretical conception in Section 4.3.3 (for linear kernel $\vec{m}_s$ ($\vec{m}_t$) is equivalent to μ_s (μ_t)).

Obviously, there are simple cases like in Figure 4.3, where SVMMMD with linear kernel does not work : when we force the classification hyperplane parameter w to be orthogonal to $\vec{m}_s - \vec{m}_t$, we might lose the discriminant information for a good classifier. In fact, a *universal kernel* (refer to Section 1.4.4) is necessary when applying MMD criterion, which avoids the former case. When applying SVMMMD to other datasets, we have used *universal kernels*, for example gaussian kernels.

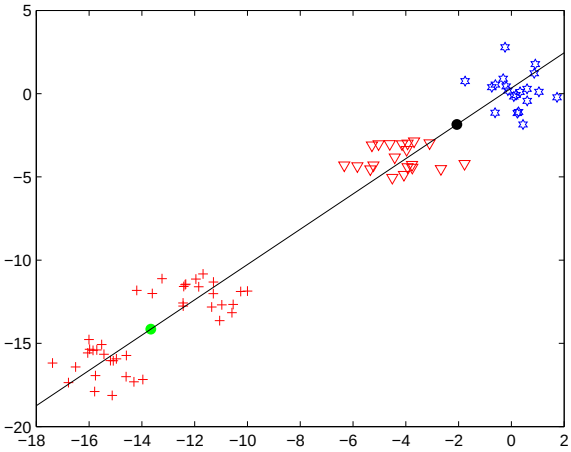


Figure 4.3 – Linearly separable data set using the linear kernel where SVMMMD cannot work (triangles and stars represent the labeled source data, while "+" symbols represent the unlabeled target data)

4.4.2 SVMMMD and LM on Banana-Orange Dataset

Banana-Orange dataset is a synthetic dataset (an example is shown in Figure 4.4(a) and Figure 4.4(c)). We first generate the source banana-orange, then transform source data to generate target data with linear transformations (translation, scaling, rotation) and non-linear transformations. In this way, source and target are different yet related. We first illustrate the classification results obtained by SVMMMD and by LM in a random example of Banana-Orange Dataset (in Figure 4.4).

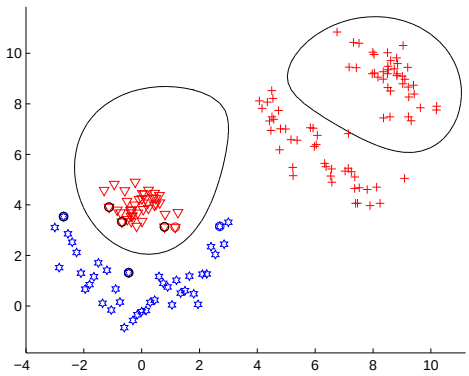
From this example, both SVMMMD and LM can deal with transfer leaning problems that standard SVM cannot. In this example, SVMMMD is better than LM with the optimal value of σ

Then, we generate 50 different banana-orange datasets and show the average performance ($\pm$ standard deviation) of SVMMMD and LM in Figure 4.5.

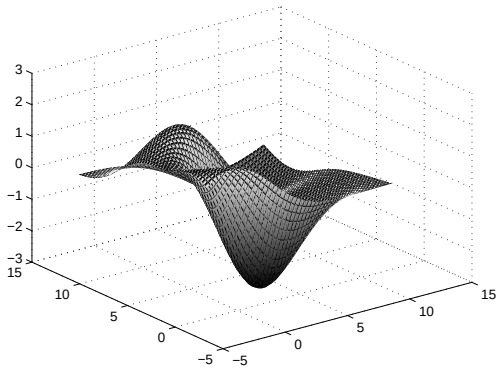
From Figure 4.5, we can conclude that on banana-orange datasets, SVMMMD performs better than LM for a larger range of kernel parameters. Furthermore, the best performance obtained by SVMMMD is generally better than that obtained by LM.

4.4.3 Influence of the Kernel Parameter

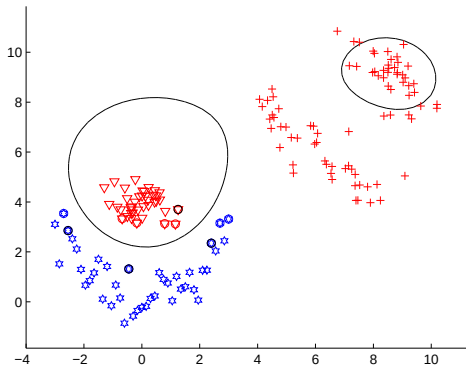
The kernel parameter should be delicately chosen for both methods. From Figure 4.5, an unsuitable kernel parameter can lead to poor performances : for SVMMMD, poor performances can be found from the beginning and a short interval $[0.75, 2]$; then with the augmentation of σ, the performance gets better ; obviously, it is not the larger the σ, the better the performance, but till 4, SVMMMD performs relatively well ; while for LM, the best performance can be found in the interval of $[0.75, 3]$. To set the optimal value of σ, we use cross-validation. A better kernel parameter selection strategy is to be developed.



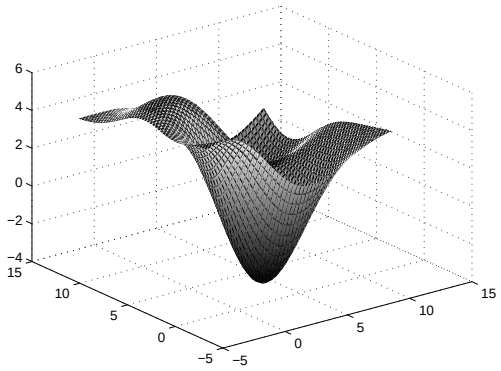
(a) Example of a classifier obtained with our method (for the optimal value of σ)



(b) Decision surface obtained (SVMMMD)



(c) Example of a classifier obtained with LM (for the optimal value of σ)



(d) Decision surface (LM)

Figure 4.4 – Results obtained on the banana-orange data set. In 4.4(a) and 4.4(c), circles and stars represent the labeled source data while "+" symbols are the unlabeled target data. In 4.4(b) and 4.4(d), the decision surfaces are plotted as functions of the input space coordinates. Thresholding these surfaces at 0 level gives the decision curves corresponding to the classifiers in 4.4(a) and 4.4(c), respectively.

4.5 Conclusion

In this chapter, transfer learning approaches based on SVM or on MMD were introduced; followed by the presentation of domain adaptation methods by SVM combined with MMD. Three previous methods were presented, namely Kernel Mean Matching (KMM), Large Margin Transductive Transfer Learning - SVM (LM) and Adaptation Regularization-SVM (ARSVM); one proposed method, Domain Adapatation by SVM subject to a MMD-like constraint (SVMMMD), published in [24].

Different from other SVM and MMD based transfer learning approaches, we are the first to

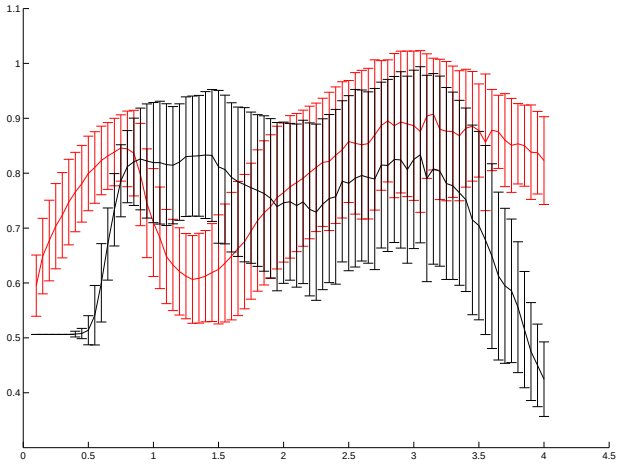


Figure 4.5 – Average performance (good classification rate) ± 1 s.d. as a function of the gaussian kernel parameter. Red line : our method. Black line : LM.

propose to use the projected MMD as a constraint and interpret geometrically the new model. With this constraint, the quadratic nature of SVM keeps unchanged and the optimization is simple and efficient (no matrix inversion required).

However, the projected MMD is no longer the MMD defined in Section 1.4.4. Thus, $< w, \mu_{X_s} - \mu_{X_t} >_{\mathcal{H}} = 0$ cannot guarantee the equality of two distributions ($P_s(X_s)$ and $P_t(X_t)$). Although experimental results (in Section 4.4 and Section 5.6) have been promising, theoretical bounds are to be developed.

Chapter 5

Domain Adaptation by KPCA Alignment

Contents

5.1 Introduction

In Chapter 4, we force $< w, \mu_{X_s} - \mu_{X_t} >= 0$ and w is expected to be orthogonal to $\mu_{X_s} - \mu_{X_t}$. In this way, the separating hyperplane is supposed to work well for both source and target, because $\mu_{X_s} - \mu_{X_t}$ is supposed to be zero after a projection of data by w. However, $< w, \mu_{X_s} - \mu_{X_t} >= 0$ is not equivalent to $\mu_{X_s} - \mu_{X_t} = 0$ ($MMD = 0$). Therefore, in this chapter, we aim at finding a transformation in RKHS that can avoid this limitation. With the same assumption (Assumption 4.1) as SVMMMD, we propose KPCA domain adaptation approaches, which are shown experimentally better than SVMMMD.

In this chapter, as KPCA centers the kernel representations and $\mu_{X_s} = 0 = \mu_{X_t}$, different from SVMMMD in Chapter 4.3.3, KPCA based approaches further need to align other higher order moments of $\phi(X_s)$ and $\phi(X_t)$. Therefore, as shown in Section 5.5 and Section 5.6, in general, the proposed domain adaptation by KPCA alignment approaches (in Section 5.3.2, Section 5.3.3 and Section 5.4) perform better than SVMMMD (in Chapter 4).

Kernel Principal Component Analysis (KPCA) has been widely used in data pre-processing, especially for large-scale data. It generally serves as an unsupervised dimensionality reduction tool. Compared to other unsupervised dimensionality reduction methods (LLE [109], ISOMAP [137], Laplacian eigenmaps [6], Diffusion map [70], etc), KPCA is the most simple approach that takes into consideration the non-linearity of data. To the best of our knowledge, it has hardly been used for the domain adaptation objective. An overview of dimension reduction based domain adaptation will be presented in Section 5.2.1.

Furthermore, with the help of KPCA, we make the implicit RKHS representations explicit. Based on the explicit kernel space representations of source and target data, the alignment between source and target can be easily carried out in their KPCA representations. The KPCA subspace considered here is found by KPCA Coordinate System Alignment, which is similar to [2] but more robust than this approach. Other domain adaptation approaches based on subspace alignment are overviewed in Section 5.2.2. Moreover, the kernel representation alignment complements the existing kernel space alignment domain adaptation approaches.

5.2 Background

5.2.1 Dimension Reduction based Domain Adaptation

Domain Adaptation using Discriminative Dimensionality Reduction

Linear Discriminant Analysis (LDA) is one of the most popular dimension reduction strategies that have been adapted to domain adaptation. In most cases, the principle of traditional LDA

is kept :

$$\begin{aligned}
&\arg\max_w \operatorname{tr}\left(\frac{w^T S_b w}{w^T S_w w}\right) \Leftrightarrow \arg\max_w \operatorname{tr}\left(\frac{w^T S_b w}{w^T S_t w}\right) \\
&\text{s.t.} \quad w^T S_w w = 1 \quad \text{or} \quad w^T S_t w = 1 \\
&\text{with} \quad S_w = \sum_{i=1}^{C} \sum_{j}^{l_i} (x_{ij} - m_i)(x_{ij} - m_i)^T \\
&\qquad S_b = \sum_{i=1}^{C} l_i (m_i - m)(m_i - m)^T \\
&\qquad S_t = S_b + S_w
\end{aligned}$$

where x_{ij} is the jth observation of class i ; $m_i, \forall i = 1, \ldots, C$ is the mean of examples in class i ; m is the mean value of the whole data $m = \frac{1}{N} \sum_{i=1}^{c} n_i m_i$ where $N = \sum_{i=1}^{c} n_i$. LDA is looking for the projection direction w that minimizes the within-class scatter (S_w) and maximizes the between-class scatter (S_b) simultaneously. The optimal w is formed by eigenvectors corresponding to the $C-1$ largest eigenvalues of $S_t^{-1} S_b$ (or $S_w^{-1} S_b$)([47]).

Then, to adapt to the transfer learning context, Wang et al. ([151]) propose to modifying S_b and S_w by taking into consideration the unlabeled target data :

$$\begin{aligned}
&\arg\max_w \operatorname{tr}\left(\frac{w^T S'_b w}{w^T S'_t w}\right) \\
&\text{s.t.} \quad w^T S'_t w = 1 \\
&\qquad S'_b = S_b + S_b^u \\
&\qquad S'_t = S_t + \lambda M
\end{aligned}$$

and S_b^u is the between-class scatter found by clustering unlabeled target data and M is the graph Laplacian term ([25], the detail of matrix M can be found in Annexe 2.1). M represents the intrinsic structure that is shared by source (labeled) and target (unlabeled).

Others integrate a between-domain scatter to penalize the source and target similarity. Then transferred LDA can find an optimum projection w that renders source and target similar while simultaneously encouraging the distinction of classes on the merged source and target data ([141]). The scatter is further extended to kernelized distributional variance (defined in [87]). Then the objective becomes learning a dimension reduction transformation along which the distributional variance between samples is minimized and simultaneously, the functional relationship between samples and their labels is preserved. The source knowledge is then transferred by finding a kernel matrix that is maximally domain invariant.

Domain Adaptation using Unsupervised Dimensionality Reduction

Generally, for unsupervised dimensionality reduction based domain adaptation methods, some similarity measures should be added to regularize the dimension reduction criteria. In [93], dimension reduction projections are found by minimizing the distance between $P(\phi(X_s))$ and $P(\phi(X_t))$ while maximally preserving the data variance (similarly to KPCA). Si et al. ([125]) proposed a general framework that incorporates many dimensionality reduction methods into

domain adaptation transfer learning :

$$w = \arg\min_{w} F(w) + \lambda D_w(P_s||P_t)$$
$$\text{subject to specific constraint(s)}$$

where $D_w(P_s||P_t)$ is the Bregman divergence that measures some distance between P_s and P_t in the projected subspace ; $F(w)$ represents any suitable objective function, based on PCA, Locality Preserving Projection (LPP [58]), Marginal Fisher Analysis (MFA [161], source labels considered), LDA (in previous subsection), etc.

Although some literatures have applied regularized PCA or KPCA, their objective is to find the optimum transformation ($\mathcal{R}^d \rightarrow \mathcal{R}^l$, where $d \gg l$) where source and target are aligned. Our works, instead, aim at aligning the KPCA coordinate systems (or in other words, the KPCA basis of source and target data) where the source and target *kernel* representations are as similar as possible.

5.2.2 Domain Adaptation by Subspace Alignment

Subspace alignment aims at finding a common subspace where source and target share as much similarity as possible. Then, the label information of transformed source data can be transferred to transformed target data. This is also our objective of KPCA domain adaptation.

In [54], source and target subspaces are represented by two points in the Grassmann manifold. Along the smooth geodesic between these two points, new points are generated to represent the intermediate subspaces. Let S'_c denotes the concatenation of all subspaces (of dimension $d_c \times 1$). Then we project source and target data onto S'_c, each source data x_i^s is now represented by $x_i^{s'}$ (of dimension $d_c \times 1$) ; similar for target data ($x_j^t \rightarrow x_j^{s'}$). Then, the classifier trained on transformed source $x_i^{s'}$ (labeled) is supposed to perform well on target transformed data $x_j^{s'}$ (unlabeled). Thus, via geodesic flow, source and target subspaces are aligned. Further research is related to avoid parametrization (the number of intermediate subspaces that are suitable for learning) (GFK [53]).

Others suppose that the target data can be linearly represented by source data in a common subspace ([158]) :

$$\exists P, Z : P^T X_t \approx P^T X_s Z \quad \text{s.t. constraints on } P \text{ and } Z \tag{5.1}$$

Their objective is to find the optimum P and Z that :

$$\min_{P,Z} ||P^T X_t - P^T X_s Z||_F^2$$

The above problem is ill-posed, so extra regularization terms have been introduced w.r.t Z and P.

$$\min_{P,Z,E} ||P^T X_s - Y||_F^2 + \lambda||P||_F + \gamma||Z||_* + \alpha||Z||_1 + \beta||P^T X_t - P^T X_s Z||_1$$

For example, low-rank constraint on Z and subspace learning function ($F(P, X_s) = ||P^T X_s - Y||_F^2 + \lambda||P||_F$) w.r.t P ([120]).

[133] combines PCA with previously overviewed methods and proposes two learning strategies :

- Strategy 1 : this strategy is under Assumption 5.1 ; first, source and target data are transformed into principal components by PCA. Then, through two successive alignment steps : first alignment of basis and then alignment of distributions, source and target data are rendered similar. Similar subspace alignment can be found in [43] without the alignment of basis step. To the best of our knowledge, the alignment of basis is novel and similar to our approach. However, we align basis and distributions at the same time. Moreover, there is no assumption of a linear transformation between either basis or distributions.

- Strategy 2 : this strategy combines PCA with GFK (Geodesic Flow Kernel [53]). The original source and target data are preprocessed by PCA. Then, the respective principal components are used to replace original source and target data. After finding the common subspace, a distribution alignment between projected original source and target data is further used to achieve better performances. Still, this alignment is realized under Assumption 5.1.

5.2.3 Domain Adaptation by Kernel Space Alignment

Kernel feature map ($\phi(.)$) generally transforms low-dimensionality data into a higher-dimensional space. Given a kernel, the explicit expression of $\phi(.)$ is generally impossible to find. However, to study the similarity among samples, explicit expression of $\phi(.)$ is not necessary. We can apply inner-product between their mapped kernel features, $k(x_i, x_j) =< \phi(x_i), \phi(x_j) >$ (More details can be found in Section 1.3). Therefore, previously proposed alignments in RKHS generally use the alignment of kernel matrices K ($K(i,j) =< \phi(x_i), \phi(x_j) >$). Until now, there have not been a lot of related works on domain adaptation.

[169] deals with the covariate shift problems (in Section 3). Authors assume that if two kernel matrices are the same, then their corresponding empirical feature maps ($\phi_{emp} = K^{\frac{1}{2}}$ [112] and [93]) will also be the same, so are the distributions w.r.t the corresponding kernel induced features. They first define a surrogate kernel $\mathcal{K}_{\mathcal{T}\leftarrow\mathcal{S}}$ (Definition in Annexe 2.2). The surrogate kernel is estimated by projecting a given kernel matrix (on $\mathcal{S}$) to an arbitrary sample (from $\mathcal{T}$) while preserving the key eigen-structures (found by Nystrom Kernel Approximation in Annexe 2.3). The the transfer learning problem becomes :

$$\min_{\mathcal{P}\in\mathcal{R}^{|\mathcal{T}|\times|\mathcal{T}|}} ||P^T K_{TT} P - \mathcal{K}_{\mathcal{T}\leftarrow\mathcal{S}}||_F^2 + \lambda||P||_F^2$$
$$\tilde{K}_{TT} = P^T K_{TT} P =< \phi(X_t)P, \phi(X_t)P >=< \phi(\tilde{X}_t), \phi(\tilde{X}_t) >$$

The objective is to find the optimum P and then cross-domain similarity (G) becomes :

$$G = \begin{bmatrix} < \phi(\tilde{X}_t), \phi(\tilde{X}_t) > & < \phi(\tilde{X}_t), \phi(X_s) > \\ < \phi(X_s), \phi(\tilde{X}_t) > & < \phi(X_s), \phi(X_s) > \end{bmatrix} = \begin{bmatrix} P^T K_{TT} P & P^T K_{TS} \\ K_{ST} P & K_{SS} \end{bmatrix}$$

Then, in [76] the assumption is relaxed for domain adaptation context. They relax the eigenspectrum presented in previous paper (eigenspectrum in the original Nystrom Kernel Approximation see Annexe 2.3) to a learnable parameter. Then this learnable parameter can be transferred to get cross-domain similarity.

Different from [169] and [76], we apply KPCA to explicitly estimate kernel representations and the cross-domain similarity in a RKHS. Then the similarity is maximized.

5.2.4 Robust Transfer using Principal Component Analysis

To the best of our knowledge, Robust Transfer using Principal Component Analysis (RTPCA [57]) is the first parameter-based transfer learning using PCA. It provides an orientation for our future work.

RTPCA first considers PCA as a regression problem :

$$\min_{Z,B} ||X - ZB||^2 \quad \text{s.t.} \quad BB^T = I_k \tag{5.2}$$

where I_k is the identity matrix of dimension $k \times k$; Z is the low-dimensional encoding matrix ; B is the basis matrix and $X = ZB + E$ with E representing the error.
Following a similar idea as in Eq 5.2, scalable robust PCA is proposed ([153]) :

$$\min_{M,E} ||M||_* + \lambda||E||_1 + \frac{\alpha}{2}||M + E - X||_F^2$$

where $M = ZB$; $||.||_*$ is the nuclear norm ($||M||_* = trace(\sqrt{M^T M})$) ; $||.||_1$ is the l_1-norm ; λ and α are trade-off parameters.

For source and target data, we can have :

$$X_s = N_s B_c + Z_s B_s + E_s$$
$$X_t = N_t B_c + Z_t B_t + E_t$$

where $B_c \in \mathcal{R}^{k_c \times d}$ is the orthogonal basis shared by source and target ; $B_s \in \mathcal{R}^{k_s \times d}$ and $B_t \in \mathcal{R}^{k_t \times d}$ are specific orthogonal bases for source and target ; $N_s \in \mathcal{R}^{n_s \times k_c}$, $N_t \in \mathcal{R}^{n_t \times k_c}$, $Z_s \in \mathcal{R}^{n_s \times k_s}$, $Z_t \in \mathcal{R}^{n_t \times k_t}$ are matrices containing coefficients to be optimized ; let $Z_c = [N_s; N_t]$ and $Z_c \in \mathcal{R}^{(n_s+n_t) \times k_c}$; $A_s = [I_{n_s}, 0_{n_s,n_t}]$ and $A_t = [I_{n_t}, 0_{n_t,n_s}]$ (I_n represents the $n \times n$ identity matrix and $0_{n_s,n_t}$ represents a $n_s \times n_t$ matrix of all zeros) ; thus, $N_s = A_s Z_c$ and $N_t = A_t Z_c$. We integrate all elements into the formulation of scalable robust PCA :

$$\min_{M_c,M_s,M_t,E_s,E_t} \frac{\alpha_s}{2}||A_s M_c + M_s + E_s - X_s||_F^2 + \frac{\alpha_t}{2}||A_t M_c + M_s + E_t - X_t||_F^2 + \beta_s||E_s||_1 + \beta_t||E_t||_1 + \lambda_c||M_c||_* + \lambda_s||M_s||_* + \lambda_t||M_t||_*$$

where $M_{\{c,s,t\}} = Z_{\{c,s,t\}} B_{\{c,s,t\}}$.

Optimization is implemented as proposed in [57]. Unfortunately, in [57], the application is only image-denoising : restoring reliable target image using common basis and target specific basis ($\hat{X}_t = A_t M_c + M_t$).

The transformation of PCA to a regression form inspires another way of transferring a common basis B_c. For perspectives, we can also transfer the KPCA problem to some regression form and take into consideration the common basis and the specific basis.

5.3 Domain Adaptation by KPCA Coordinate System Alignment

5.3.1 KPCA Subspace Transformation

Recently, a kernel subspace alignment by KPCA was proposed ([2]) :

- Step 1 - Landmark Selection : In order to construct a reliable common subspace, landmarks are selected from source and target data. They are representative for original data (source and target) as well as similar one to another. The selection criterion used in [2] is the distribution overlap criterion :

$$\begin{aligned} &\mathcal{S} \sim \mathcal{N}(\mu_{\mathcal{S}}, \sigma_s^2) \text{ and } \mathcal{T} \sim \mathcal{N}(\mu_{\mathcal{T}}, \sigma_t^2); \\ C &= \frac{\int \mathcal{N}(x|\mu_{\mathcal{S}}, \sigma_s^2)\mathcal{N}(x|\mu_{\mathcal{T}}, \sigma_t^2)dx}{\mathcal{N}(0|0, \sigma_{sum}^2)} \\ &= \frac{\mathcal{N}(\mu_{\mathcal{S}} - \mu_{\mathcal{T}}|0, \sigma_s^2 + \sigma_t^2)}{\mathcal{N}(0|0, \sigma_{sum}^2)} \end{aligned}$$

  where $\mathcal{N}$ is the gaussian distribution and $\mathcal{N}(x|\mu, \sigma)$ is the gaussian distribution with mean μ and variance σ^2 ; C represents the distribution overlap criterion. However, the validity of the above criterion is only proved experimentally with the proposed approach. Further analytical demonstration is to be developed.

  Algorithm 1 presents the detail of landmark selection.

Algorithm 1 Selection of Landmarks

Input : Source data $\mathcal{S}$, Target data $\mathcal{T}$, Threshold th, kernel parameter s
Output : A that contains all selected landmarks

```
A ← {}
for c in S ∪ T do
    KV_s ← exp(−||c − p||²/(2s²)), p ∈ S
    KV_t ← exp(−||c − p||²/(2s²)), p ∈ T
    if C(KV_s, KV_t) > th then
        A = A ∪ {c}
    end if
end for
return A
```

- Step 2 - KPCA : source and target landmarks are transformed by KPCA.

- Step 3 - Subspace Alignment : in [2], subspace alignment is done under the Assumption 5.1. An alignment matrix can be found by using projected source and target landmarks.

- Step 4 - Classification : finally, a classifier trained on source output of Step 3 can be used on target (target output of Step 3).

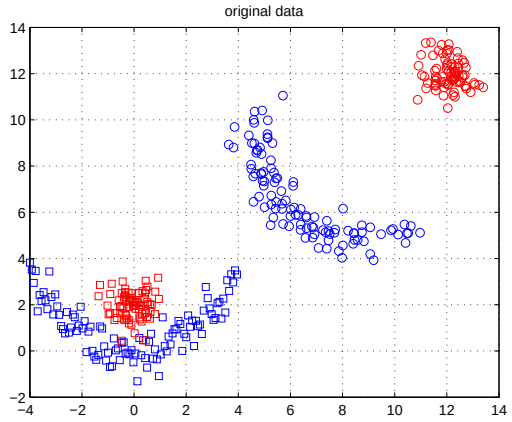
In this paper, authors align source and target data in their kernel subspace, thus extending [43] by taking into account the non-linearity of data domain-shift. However, a limitation of [2] is : the selection of landmarks may not guarantee that a linear transformation (in Step 3, Assumption 5.1) can well align source and target even when transforming both data in a kernel subspace.

5.3.2 KPCA Coordinate System Alignment (KPCA-TL)

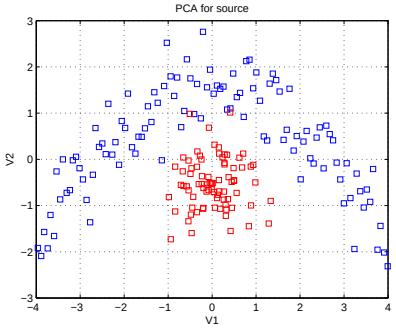
In this section, we propose a new method to align source and target KPCA coordinate systems and simultaneously their distributions therein. Furthermore, the assumption given in Eq 5.1 is no longer necessary. We start by presenting possible problems ; then our approach is proposed aiming at solving these problems.

Possible KPCA Problems

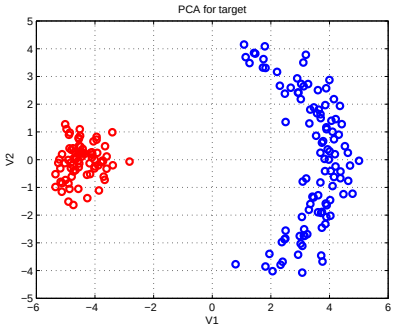
According to the presentation of KPCA (and PCA) in Section 1.4.3, when projecting source and target data that maximally preserving their variance, respectively, there may be permutation or/and inversion of KPCA axes.



(a) Original data. *Squares* represent the source and *circles* represent the target.



(b) Source data after PCA.



(c) Target data after PCA.

Figure 5.1 – Illustration of permutation of first and second eigenvectors after PCA. In both 5.1(b) and 5.1(c), abscissa corresponds to the eigenvector V1 associated to the largest eigenvalue, while ordinate corresponds to eigenvector V2 associated to the second largest eigenvalue.

We first take PCA as an example. In Figure 5.1(a), original source and target data are presented. The transformed data after PCA are presented in Figure 5.1(b) and Figure 5.1(c) for source and target respectively. We can observe that there is a permutation between eigenvectors of PCA of source and PCA of target. As the similarity between source and target is usually

intrinsic, permutation often appears, especially in real applications. Moreover, as eigenvectors are defined up to the multiplicative constant ± 1, the inversion of principal components might occur. Obviously, the problem of inversion and/or the problem of permutation of eigenvectors might also appear in KPCA.

Linearity Assumption Problems

KPCA can transform data to a higher-dimensional kernel space. According to the extended version of Assumption 5.1, in this kernel space, there exists a linear transformation that can align source and target data. We shows a counterexample in Figure 5.2. From Figure 5.2(b), we can observe that even with a well aligned coordinate system, even in a high-order subspace , the relationship between source and target is nonlinear. Furthermore, if we reduce the dimension of KPCA kernel space, it is possible that source and target become less similar than before.

Although kernel feature map can render a nonlinearly separable problem to a linearly separable one, we cannot guarantee that the best classifier remains linear after projection onto a subspace of the initial KPCA. Thus, the linear transformation assumption in KPCA kernel subspace is a limitation to kernel subspace alignment.

Model of KPCA Transfer Learning (KPCA-TL)

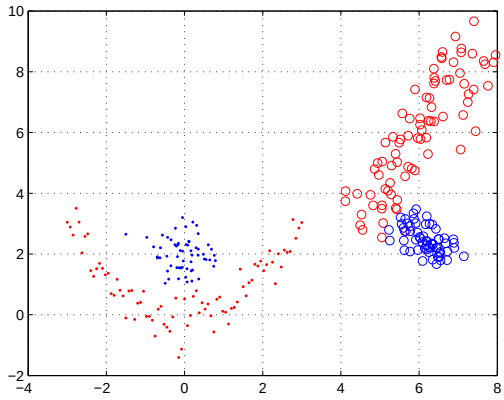
Taking into consideration the above possible problems, we propose KPCA Transfer Learning (KPCA-TL).

- Step 1 - KPCA is applied to source and target separately. But instead of reducing source and target data to a common subspace, we reduce source data to a KPCA subspace with a dimensionality that is selected smaller than that of target KPCA subspace. Let the source KPCA subspace be l_1-dimensional and let the target KPCA subspace be l_2-dimensional ($l_1 < l_2$ [6]). Then, we aims to align source and target KPCA coordinate systems by choosing l_1 suitable axes (from a total of l_2 axes) of target that match maximally the l_1 axes of source (with possible inversions and permutations taken into account).

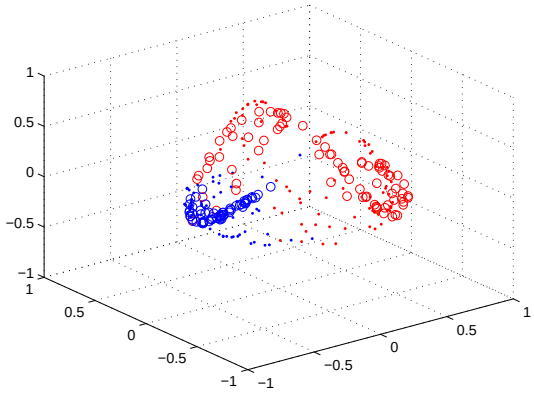
- Step 2 - Alignment of both coordinate systems and distributions : here, we apply Maximum Mean Discrepancy (MMD in Section 1.4.4). In general, MMD is applied to original source and target data, which is equivalent to align the first-order moments of $\phi(X_s)$ and $\phi(X_t)$. But as KPCA centers the data in the kernel subspace, $E[\phi(X_s)] = 0 = E[\phi(X_t)]$ after KPCA even when source and target data are not similar in the original space. Thus we choose to apply MMD on principal components obtained by KPCA, so that the distributions of source and target after KPCA are aligned, which is expected to lead to good alignment in the input space. It is important that we do not suppose the existence of any linear transformation between kernel representations of source and target. From Step 1, there are $C_{l_2}^{l_1} 2^{l_1}$ possible bases (of dimension l_1) formed by target KPCA subspace axes. Thus, we propose to calculate the MMD for all $C_{l_2}^{l_1} 2^{l_1}$ possibilities [7] to select the best KPCA common subspace for source and target. The best alignment is achieved when the

6. If we suppose $l_1 > l_2$, the reasoning remains similar and it just needs to inverse the role of source KPCA subspace and target KPCA subspace

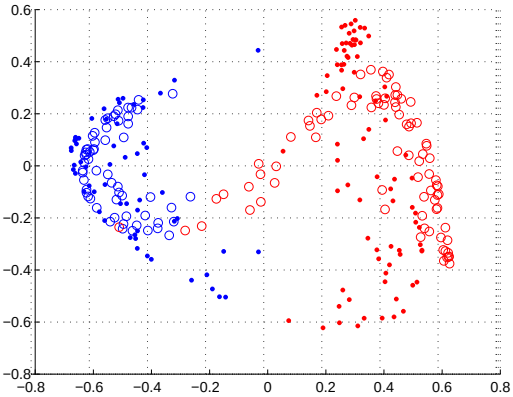
7. Usually, we use relatively small l_1 and l_2.



(a) Original data. Points represent the source and circles represent the target (supposed unlabeled).



(b) Data in KPCA kernel space (of a dimensionality 3) with the KPCA coordinate systems already well aligned. Points are source and circles are target.



(c) Data in a KPCA kernel space (of a dimensionality 2), a possible case obtained from Figure 5.2(b). Points are source and circles are target.

Figure 5.2 – Illustration of non-linear relationship between source and target after KPCA.

MMD value is minimum.

– Step 3 - Classification : finally, we use SVM on source data in the aligned KPCA subspace and it is supposed to perform well on unlabeled target data.

5.3.3 Posterior Linear Transformation to further improve the Alignment

After KPCA-TL, source and target data are assumed to be well aligned. However, according to the experimental results (in Section 5.5 and Section 5.6), sometimes the performance of SVM classifier on target data is not fully satisfactory. Therefore, we consider to apply a linear transformation between source and target in the KPCA-TL subspace to further improve the performance. Here, a linear transformation matrix is learned by minimizing the residual MMD (KPCA-TL-LT). Sometimes, KPCA-TL-LT can improve the performance. We think it is because the corresponding optimization problem is non convex.

5.4 Domain Adaptation by Kernel Space Alignment After a Linear Transformation in the Input Space and its Kernel Representations

In the previous section, we have to calculate $C_{l_2}^{l_1} 2^{l_1}$ different MMDs, which is tedious when l_1 and/or l_2 augment. Therefore, we propose a fast KPCA based RKHS alignment approach : as we shall see, by finding a proper linear transformation (M) in the original input space, the KPCA principal components and the distributions are aligned simultaneously in a RKHS, while avoiding the combinatorial aspects appearing in the previous section. Again, the similarity measure is MMD.

5.4.1 Step 1 : Linear Transformation in the Original Input Space

Let M be a linear transformation matrix such that target data (X_t) is transformed to $X_t M$. If we apply the alignment directly by minimizing an estimate of the squared MMD, we obtain :

$$\begin{aligned} &\arg\min_M || \frac{1}{n_s} \sum_{i=1}^{n_s} \phi_s(x_i^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi_s(x_j^t M) ||^2 \\ &\arg\min_M \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} < \phi_s(x_i^s), \phi_s(x_j^s) > + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} < \phi_s(x_i^t M), \phi_s(x_j^t M) > \\ &\qquad - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} < \phi_s(x_i^s), \phi_s(x_j^t M) > \end{aligned} \tag{5.3}$$

Obviously, the optimization seems infeasible : the objective function is generally non-convex w.r.t M ; and there will be heavy computation burden. Thus, there is no guarantee to find a suitable linear transformation M to best align source and target data. Therefore, we try to find the kernel representations of X_s and $X_t M$ explicitly.

We first transform the metric learning (finding the best transformation matrix M) to kernel learning (finding the best kernel matrix parameter M).

In this section, we use the gaussian kernel. Then, if we express the inner product of $\phi(x_i^t M)$ and $\phi(x_j^t M)$ with the gaussian kernel, we obtain :

$$\begin{aligned} &< \phi_s(X_t^i M), \phi_s(X_t^j M) > \\ &\qquad = \exp(-\frac{(x_i^t - x_j^t)MM^T(x_i^t - x_j^t)^T}{\sigma^2}) \end{aligned}$$

Applying the above relation to target data, we can define a new gaussian kernel such as :

$$< \phi_t(x_i^t), \phi_t(x_j^t) > = < \phi_s(x_i^t M), \phi_s(x_j^t M) >$$

Then, problem 5.3 has an equivalent form :

$$\arg\min_M ||\frac{1}{n_s}\sum_{i=1}^{n_s}\phi_s(x_i^s) - \frac{1}{n_t}\sum_{i=1}^{n_t}\phi_t(x_i^t)||^2 \tag{5.4}$$

Therefore, finding a good linear transformation M is equivalent to finding a good kernel parameter M of $< \phi_t(X_t), \phi_t(X_t) >$ (where $< \phi_t(x_i^t), \phi_t(x_j^t) >= \exp(-\frac{(x_i^t - x_j^t)MM^T(x_i^t - x_j^t)^T}{\sigma^2})$) However, to solve Eq. 5.4, we need to know the inner product between two different kernel transformations ($\phi_t(.)$ and $\phi_s(.)$). But defining such inner product is odd. Therefore, we represent these two kernel transformations explicitly in their common RKHS with the help of KPCA.

5.4.2 Step 2 : KPCA

By KPCA, we can obtain :

$$\begin{aligned} \phi_s(X_s) &= X_{ps}^T \overline{V}_s, \text{where } X_{ps} = \alpha_s^T K_{SS} \\ \phi_t(X_t) &= X_{pt}^T \overline{V}_t, \text{where } X_{pt} = \alpha_t^T K_{TT} \\ \text{with } K_{SS}(i,j) &= < \Phi_s(x_i^s), \Phi_s(x_j^s) >, i,j = 1,\ldots,n_s \\ \text{and } K_{TT}(i,j) &= < \Phi_t(x_i^t), \Phi_t(x_j^t) >, i,j = 1,\ldots,n_t \end{aligned}$$

where $\Phi_{\{s,t\}}$ representing the centered kernel transformation for source or target data ($\Phi_s(x_j^s) = \phi_s(x_j^s) - \frac{1}{n_s}\sum_{i=1}^{n_s}\phi_s(x_i^s)$, similar formula for $\Phi_t(.)$) ; $\alpha_s(\alpha_t)$ are the eigenvectors of $K_{SS}(K_{TT})$. Now X_{pt} is a function of the linear transformation M applied in the input space.

With proper kernel parameters (σ and M), if we apply KPCA to $\phi_s(X_s)$ and $\phi_t(X_t)$ respectively, we should get rather similar data in the new common coordinate system spanned by $\overline{V}_{\{s,t\}}$. If we retain the same number l of principal axes (corresponding to the l first principal components), $\overline{V_s'}$ represents the same coordinate system as $\overline{V_t'}$ while there is no problem of inversion or permutation of axes [8]. We denote the common coordinate system by $\overline{V}_c(\overline{V}_c = \{\vec{V}_{c1}, \vec{V}_{c2}, \ldots, \vec{V}_{cl}\})$, then we get :

$$\phi_s(X_s) \approx X_{ps}^T \overline{V}_c, \quad \text{and} \quad \phi_t(X_t) \approx X_{pt}^T \overline{V}_c \tag{5.5}$$

8. This will be explained in Section 5.4.5.

Then, Eq. 5.4 becomes :

$$\arg\min_{M} ||\frac{1}{n_s}\sum_{i=1}^{n_s} X_{ps}^T(i,:)\overline{V}_c - \frac{1}{n_t}\sum_{j=1}^{n_t} X_{pt}^T(j,:)\overline{V}_c||^2 \tag{5.6}$$

After some further developments :

$$\begin{aligned}\arg\min_{M} &\frac{1}{n_s^2}1X_{ps}^T X_{ps}1^T + \frac{1}{n_t^2}1'X_{pt}^T X_{pt}1'^T \\ &- \frac{2}{n_s n_t}1X_{ps}^T X_{pt}1'^T\end{aligned}$$

where $1(1')$ is the row vector composed of ones and of dimension $1 \times n_s(n_t)$.

As X_{ps} is independent of M, the above optimization problem is equivalent to :

$$\arg\min_{M} \frac{1}{n_t^2}1'X_{pt}^T X_{pt}1'^T - \frac{2}{n_s n_t}1X_{ps}^T X_{pt}1'^T$$

As our objective is to determine the "best" kernel space ($\phi_t(X_t)$), which is equivalent to find the "best" X_{pt}, we could minimize MMD w.r.t X_{pt} rather than w.r.t M, we then optimize with regards to X_{pt} :

$$\begin{aligned}\mathcal{L} &= \min_{X_{pt}} \frac{1}{n_t^2}(X_{pt}1'^T)^T(X_{pt}1'^T) - \frac{2}{n_s n_t}(1X_{ps}^T)(X_{pt}1'^T) \\ \frac{\partial\mathcal{L}}{\partial X_{pt}} &= \frac{2}{n_t^2}(X_{pt}1'^T1') - \frac{2}{n_s n_t}(1X_{ps}^T)^T1' \\ &= \frac{2}{n_t^2}(X_{pt}1'^T1') - \frac{2}{n_s n_t}(X_{ps}1^T1')\end{aligned}$$

However, there are several constraints on X_{pt} that have not been taken into consideration :

- X_{pt} must be of the form $\alpha_t^T K_{TT}$ where α_t is the matrix containing l eigenvectors of K_{TT} that correspond to the l largest eigenvalues. K_{TT} should always be symmetric, definite positive. Thus, there should be constraints on X_{pt}, which are not obvious.
- There must exist M that :

$$\begin{aligned}K_{TT}(i,j) = exp(-\frac{(x_i - x_j)MM^T(x_i - x_j)^T}{\sigma^2}),& \\ i,j = 1,\ldots,n_t; \text{and } \forall x_i, x_j \in X_t&\end{aligned}$$

  In other words, the possible domain of K_{TT} depends also on M and X_t ; and even if we alternatively optimize $\mathcal{L}$, in every iteration we need to guarantee the positive definiteness of K_{TT}. It leads to an intractable optimization problem.

Therefore, even with explicit kernel representations of source and target data, finding the best alignment in the original space is challenging and is not of major interest. So we propose to align source and target data's kernel representations by MMD.

5.4.3 Step 3 : Kernel Representation Alignment

In order to align X_{ps} and X_{pt}, we measure the the MMD in another RKHS space induced by $\varphi(.)$ on X_{ps} and X_{pt}. The optimization problem can be formulated as follows :

$$\begin{aligned}
&\arg\min_M \| \frac{1}{n_s}\sum_{i=1}^{n_s}\varphi(X_{ps}^T(i,:)) - \frac{1}{n_t}\sum_{j=1}^{n_t}\varphi(X_{pt}^T(j,:))\|^2 \\
&\arg\min_M \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} < \varphi(X_{ps}^T(i,:)), \varphi(X_{ps}^T(j,:)) > \\
&\quad + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} < \varphi(X_{pt}^T(i,:)), \varphi(X_{pt}^T(j,:)) > \\
&\quad - \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} < \varphi(X_{ps}^T(i,:)), \varphi(X_{pt}^T(j,:)) >
\end{aligned} \tag{5.7}$$

where $\varphi(.)$ is the kernel transformation, used to compute MMD between source and target data (after KPCA projection). Note that Eq. 5.7 differs from Eq. 5.4 by the introduction of $\varphi(.)$. Minimizing Eq. 5.7, we expect that X_{ps} and X_{pt} share similar distributions, in other words, after different kernel transformation $\phi_s(.)$ and $\phi_t(.)$, the source and target data become similar. In this way, not only the optimum M is easy to find, also the problem of permutation and inversion of KPCA axes is avoided (see Section 5.4.5).

5.4.4 Step 4 : Linear Classification

We finally apply soft margin SVM on source KPCA representation (X_{ps}), thus avoiding the use of preimage. Further, as KPCA can render nonlinearly separable data linearly separable, in the common RKHS found by KPCA, a linear SVM can perform well. For most of the cases, the decision function is a hyperplane in this space.

5.4.5 Fast Search for Parameter

To simplify the optimization, we limit the range of M to be the class of orthonormal bases. We transform the original data by projecting them on a new basis. Therefore, $MM^T = \gamma^2 I$, where γ is the scaling factor (a scalar value) and I is the identity matrix. As learning the linear transformation M is equivalent to learning a parameter of kernel induced by $\phi_t(.)$, we are looking for a suitable kernel parameter σ_t such that the exact $< \phi_t(x_i^t), \phi_t(x_j^t) >$ can be approximated by $\exp(-\frac{(x_i^t - x_j^t)(x_i^t - x_j^t)^T}{\sigma_t^2})$. Therefore, with this restriction, the optimization problem becomes scalar (non-convex). A simple grid search strategy is sufficient to determine a good value of σ_t. During this grid search, the different choices for σ_t give possible orientations of the eigenvectors resulted from KPCA. So, when evaluating MMD to determine an optimal σ_t^*, within the neighborhood of σ_t^*, we have with high probability a good alignment of KPCA bases of source and target.

Experimental results have shown good performance with this strategy. Therefore, such a linear transformation in the original input space can result in non-linear alignment in a common

KPCA subspace, which can lead to relatively good performances.

It is also possible to extend our approach to heterogeneous transfer learning cases. In these cases, the transformation matrix M_{heter} is no longer a squared matrix, but the idea of limiting M_{heter} to be a change of basis can still be used, satisfying $M_{heter}M_{heter}^T = \gamma_{heter}^2 I$. Therefore, optimizing γ_{heter} can solve the problem. However, further experiments are needed to validate this extension.

Thereafter, we denote this approach by KPCAlin.

5.5 Simulations and Analysis

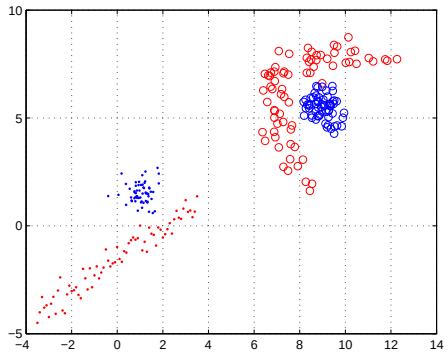
5.5.1 Simulations on Synthetic Datasets

We first illustrate the efficiency of KPCA-TL (in Section 5.3.2) and KPCA-TL-LT (in Section 5.3.3) on two synthetic datasets.

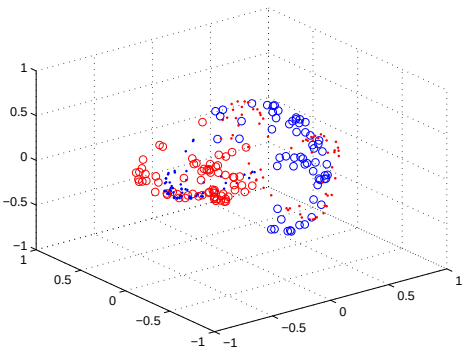
For the first dataset, source data is represented by the lower left points while target data is represented by the upper right circles (in Figure 5.3(a)). Obviously, the source data classification task is a linear one while that of target data is nonlinear. For comparaison, the original data is processed by KPCA without alignment (in Figure 5.3(b)) ; the result after KPCA-TL (after KPCA coordinate systems alignment and distributions alignment) is shown in Figure 5.3(c) ; linear transformation afterwards (KPCA-TL-LT) can further improve the result, shown in Figure 5.3(d) ; in Figure 5.3(e), we apply an optional preimage technique to show that source and target data are aligned when transformed back to the original space ; for this dataset, a linear classifier can classify well source and target data in the original space.

The second synthetic dataset is shown in Figure 5.4. Source data is represented by points, which forms concentric circles in the lower left ; target data is represented by circles, which forms an ellipse and a circle in the upper right. The objective is to find a classifier based on source to well classify target. In Figure 5.4(a), original data is presented. Then, source and target data after KPCA without alignment are shown in Figure 5.4(b). As can be seen, an inversion of the abscissa can be observed. In Figure 5.4(c), KPCA-TL is applied and we have obtained relatively well aligned source and target. Then, in Figure 5.4(d), KPCA-TL-LT has improved the alignment, where a linear classifier can be satisfactory (shown in Figure 5.4(e)). Similarly to the previous simulation, we transform the KPCA space data back to the original space (Figure 5.4(f)) and find that a single classifier can separate both source and target data. However, the original data distribution is not obtained because of the strong dimensionality reduction effect of the KPCA subspace selection and because KPCA axes are selected to align source and target data, not necessarily to give a good representation of them.

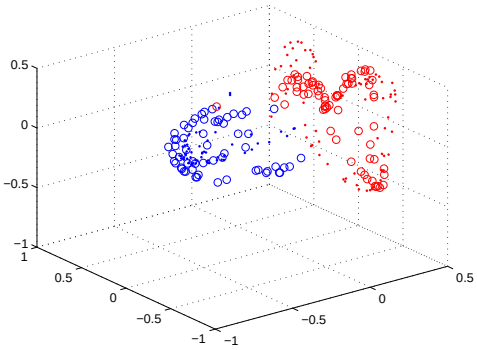
We then compare KPCA-TL, KPCA-TL-LT and KPCAlin. We use similar synthetic datasets. Figure 5.5 illustrates the comparison on the first synthetic data set (similar to Figure 5.3) while Figure 5.6 illustrates the comparison on the second synthetic data (similar to Figure 5.4).



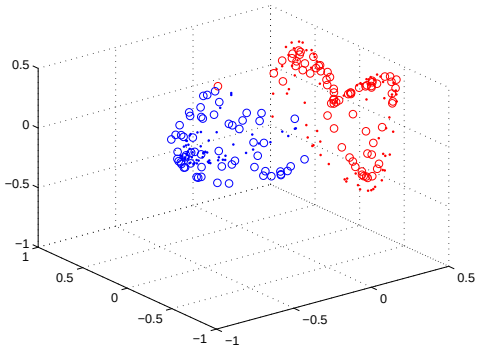
(a) original data



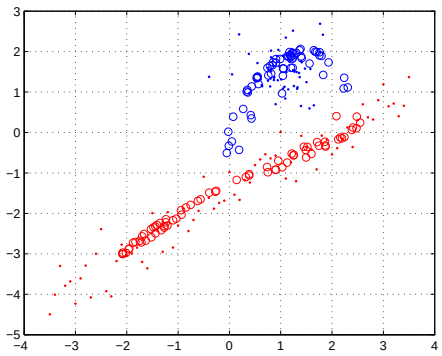
(b) Step 1 : After KPCA



(c) Step 2 : After KPCA-TL

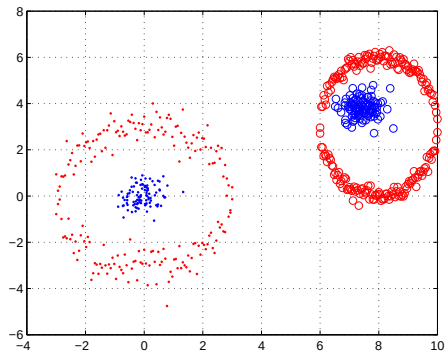


(d) Step 3 : After Linear Improvement (KPCA-TL-LT)

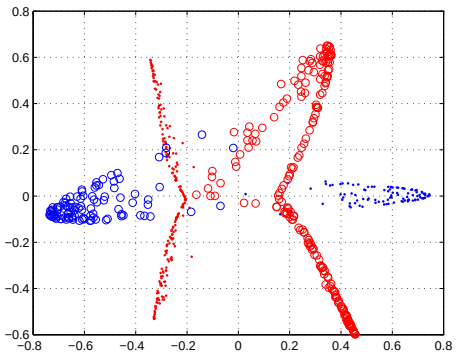


(e) Step 4 : After Preimage

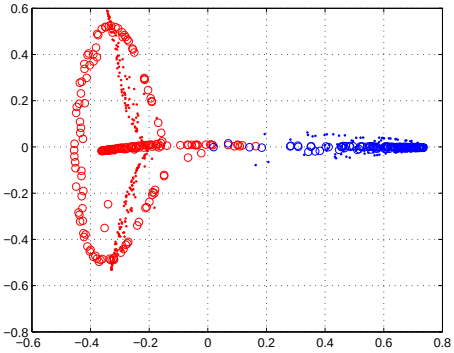
Figure 5.3 – Results obtained on the banana-orange data set. Points represent the labeled source data while circles are the unlabeled target data. Different colors represent different classes. Although target data have different colors, they are assumed to be unlabeled. The matching of colors between circles and points represents the matching results. In the final step, we can see that a linear classifier in the KPCA space that can well separate source data can also well classify target data.



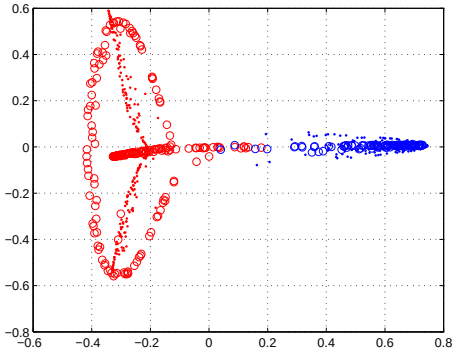
(a) original data



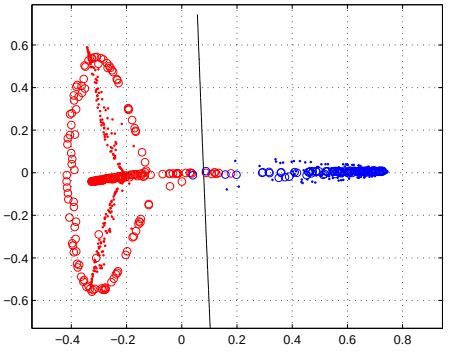
(b) Step 1 : After KPCA



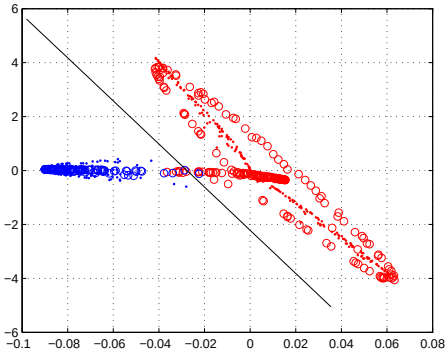
(c) Step 2 : After KPCA Axes Matching



(d) Step 3 : After Linear Alignment

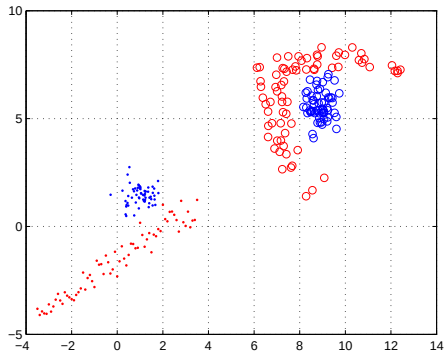


(e) Step 4 : Decision Function (black line)

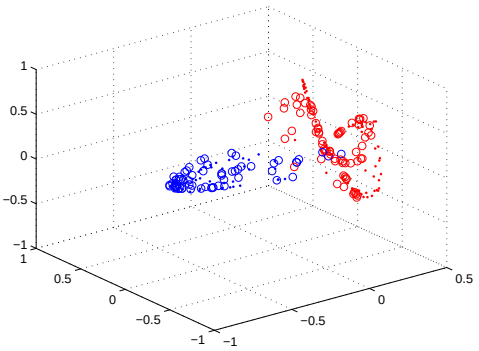


(f) Step 5 : After Preimage

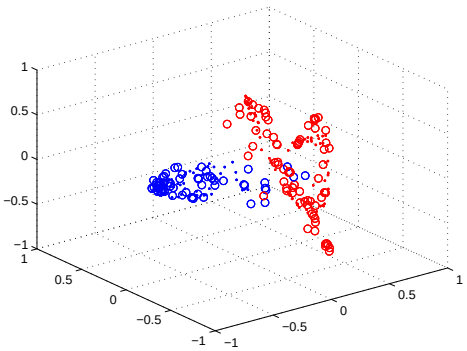
Figure 5.4 – Results obtained on the concentric circle data set. Points represent the labeled source data while circles are the unlabeled target data. Points form concentric circle problem while circles form non concentric circles problem. Different colors represent different classes. Although target data have different colors, they are assumed to be unlabeled. The matching of colors between circles and points represents the matching results. In step 3 and step 4, we can see that a linear classifier that can well separate source data can well classify target data. It is normal that the preimage technique doesn't return the original data because of its dimension reduction effect and because KPCA axes are selected to align source and target data, not necessarily to give a good representation of them.



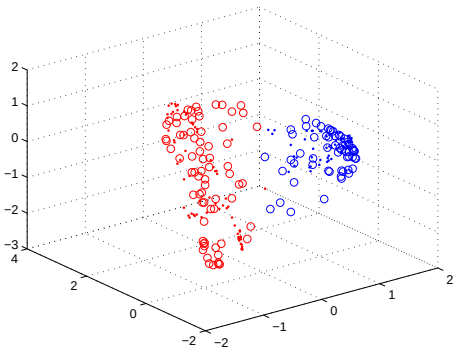
(a) original data



(b) data after KPCA-TL

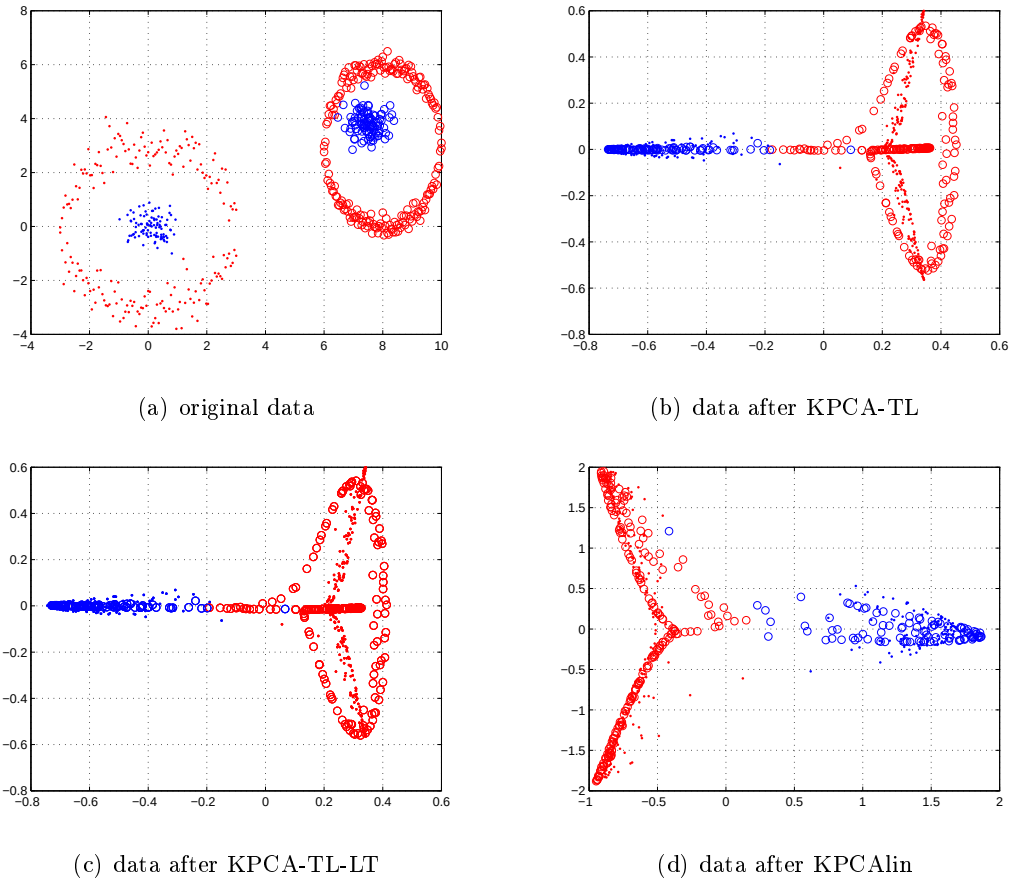


(c) data after KPCA-TL-LT



(d) data after KPCAlin

Figure 5.5 – Comparison among KPCA-LT, KPCA-TL-LT, KPCAlin. Figure 5.5(a) shows the original data : points (lower left) represent source while circles (upper right) represent target.



(a) original data

(b) data after KPCA-TL

(c) data after KPCA-TL-LT

(d) data after KPCAlin

Figure 5.6 – Comparison among KPCA-TL, KPCA-TL-LT, KPCAlin. Figure 5.6(a) shows the original data : points (lower left concentric circles) represent source while circles (upper right the ellipse and the circle) represent target.

5.5.2 Efficiency Comparison

We compare the efficiency and computational time for KPCA-TL-LT and KPCAlin. Because in the synthetic cases, l ($l = 3$ for the first synthetic dataset and $l = 2$ for the second synthetic dataset) is very small and there are 150 points for source in the first synthetic dataset and 200 points for source in the second synthetic dataset (same number of points for target as source data in both synthetic datasets). Here is a table comparing the efficiency of both datasets with the same parameter setting :

	KPCA-TL-LT		KPCAlin	
	Performance	Time	Performance	Time
Dataset in Figure 5.5	0.9542	6.9296	0.9542	5.1347
Dataset in Figure 5.6	0.9702	7.6332	0.9934	8.8084

Table 5.1 – Efficiency on synthetic datasets

In Table 5.1, the computational time for the second dataset (in Figure 5.6) is longer than that for the first dataset (in Figure 5.5). Even though l decreases from the first dataset to the second dataset, there are more points in the second dataset and the time for calculations of kernel matrices is a much more important factor than l. Then we compare the computation time of KPCA-TL-LT and KPCAlin for both datasets : in the first dataset, we have to calculate 160 times MMD for KPCA-TL-LT while 100 times MMD for KPCAlin ; thus, KPCAlin is quicker than KPCA-TL-LT ; however, in the second dataset, we have to calculate 60 times MMD for KPCA-TL-LT while always 100 times MMD for KPCAlin ; thus, KPCA-LT-LT is quicker than KPCAlin. The times of calculation of MMD depend on l_1 and l_2 (defined in Section 5.3.2, $l_2 = 6$ for both datasets and $l_1 = l$) for KPCA-LT-LT while depend on the number of scalar candidates for KPCAlin (scalar optimization for KPCAlin). Note that the time can vary even with the same method on the same dataset, especially for KPCA-TL-LT ; because the computation time of the optimization of the posterior linear transformation can vary ; however, as l is small, the computation time for this optimization usually does not influence much the final result.

Generally, KPCAlin is more efficient when l is large and the results obtained are relatively good compared to KPCA-TL-LT (sometimes better). KPCA-TL-LT is more effective when l is small (2 or 3). In Section 5.6, there will be a comparison of our results with those from the literaturesS.

5.5.3 Tuning of Kernel Parameters

For KPCA-TL and KPCA-TL-LT, the parameters to be tuned are l_2, l_1, σ_{KPCA}, σ_{MMD}. l_1 is the dimensionality of the final KPCA basis ; l_2 is the dimensionality of potential KPCA basis ; σ_{KPCA} is the kernel parameter that is used for KPCA transformation ; σ_{MMD} is the kernel parameter that used for calculating MMD. σ_{KPCA} is usually selected according to the source KPCA results : the positive group and negative group should be well separated. l_2 is usually bigger than l_1. In our simulations, $l_1 = 2$ or 3, while $l_2 = 6$. σ_{MMD} is selected such that every change from the optimum basis will lead to an obvious change in MMD. In real datasets (presented in Section 5.6), we choose σ_{KPCA} and l_1 together, following the strategy below.

For KPCAlin, the parameters are σ_{KPCA} (σ for source KPCA) , l, σ_{MMD} and the interval

for scalar optimization. σ_{KPCA} and l are found together such that the optimum pair (σ_{KPCA}, l) maximizes the information gap : $G = \frac{D_i}{D_{i+1}}, i = 1, \ldots, l-1$, where D_i is the ith eigenvalue, representing the percentage of information kept by using ith principal components. In this way, we consider the data retained to hold the principal information and there is a large gap between this useful data and the data left, which is considered as noise. Then for the scalar optimization, we first fix σ, the kernel parameter used for source data KPCA, then set an interval of $[\frac{\sigma}{10}, 10\sigma]$ and grid search on 100 candidates regularly spaced within this interval. In general, this interval includes the optimum value (for all our simulations and real experiments). To find a suitable value for σ_{MMD}, we use a similar way as above.

5.6 Experiments

In this section, we apply our proposed approaches (SVMMMD in Chapter 4, KPCA-TL, KPCA-TL-LT and KPCAlin in Chapter 5) to real datasets. A brief description of datasets is given in Section 5.6.1. Then, we compare our approaches to other well-known homogeneous transductive transfer learning approaches (no labeled target data is available) : Table 5.2 shows the experimental results and the comparison is given in Section 5.6.3. Finally in Section 5.6.4, we analyse the influence of parameters.

5.6.1 Datasets

For real data sets, we first use the USPS real data set, a handwritten digits data set. There are training and testing subsets, both containing images (16×16 pixels) of handwritten digits 0 to 9. We use the training subset as our source data while the testing subset is our target data. We suppose that there is no label information for target data. As in [142], the objective is to separate digit 4 and digit 7 as the source task and to separate digit 4 and digit 9 as the target task (USPS-Task 1). Other similar transfer learning data set can be formed, for example, the classification of digits 3 and 6 (source) to help classification of digits 3 and 8 (target). Generally, we take advantage of an easier task to help the classification of a harder task (see Fig.5.7, where a t-SNE plot ([80]) illustrates this case).

Some other data sets from the UCI data repository are also used here, namely IRIS and SEED. For the IRIS dataset [9], we know that there are 3 classes (4 attributes), easily separated. For the source, we take iris-setosa and iris-versicolor as source positive and source negative class, while for the target, we consider iris-versicolor as negative class and iris-virginica as positive class (IRIS-Task 1), respectively. We can also form another transfer learning task by using iris-setosa and iris-virginica as source while iris-versicolor and iris-virginica as target (IRIS-Task 2). Figure 5.8 shows the PCA representation of iris data. From Figure 5.8, it is more difficult to separate iris-versicolor and iris-virginica than to separate iris-versicolor and iris-setosa or iris-virginica and iris-setosa.

SEED dataset [10] consists of 3 classes (7 attributes). Similarly to the IRIS dataset, we construct two transfer learning tasks for the SEED data set : SEED-Task 1 - source data are the Canadian

9. https ://archive.ics.uci.edu/ml/datasets/iris
10. https ://archive.ics.uci.edu/ml/datasets/seeds

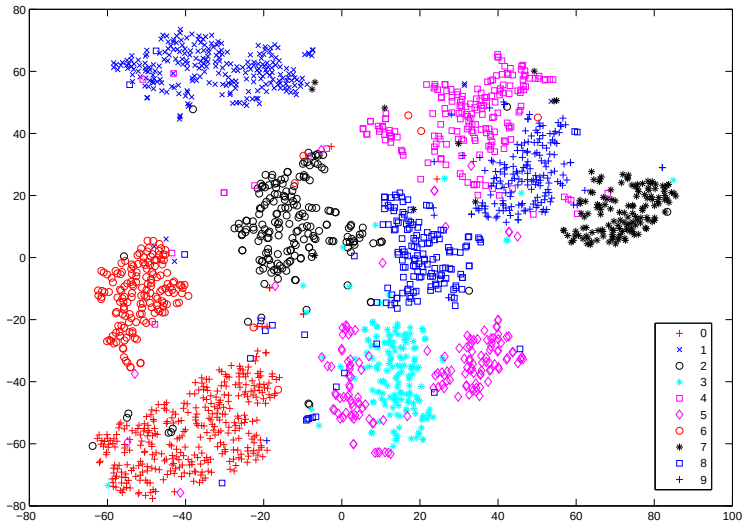


Figure 5.7 – t-SNE plot of USPS testing data.

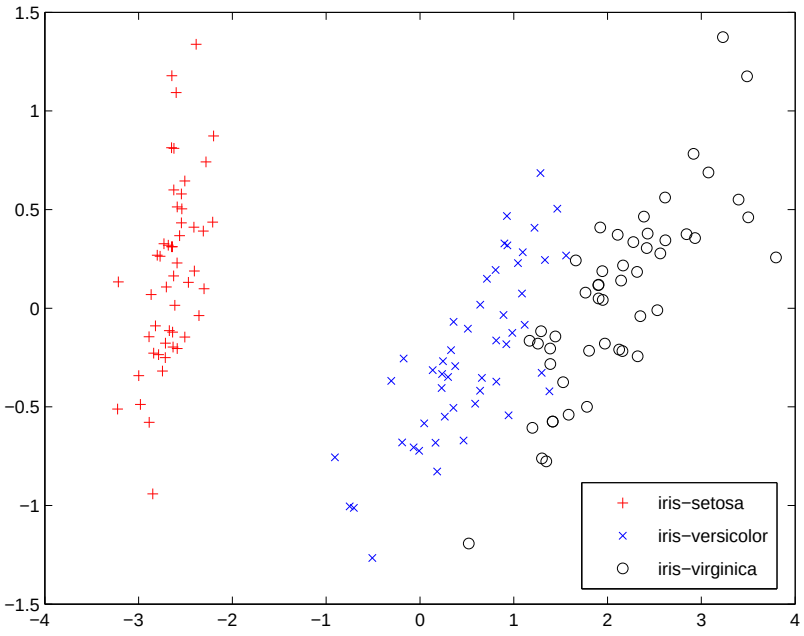


Figure 5.8 – PCA representation of iris dataset.

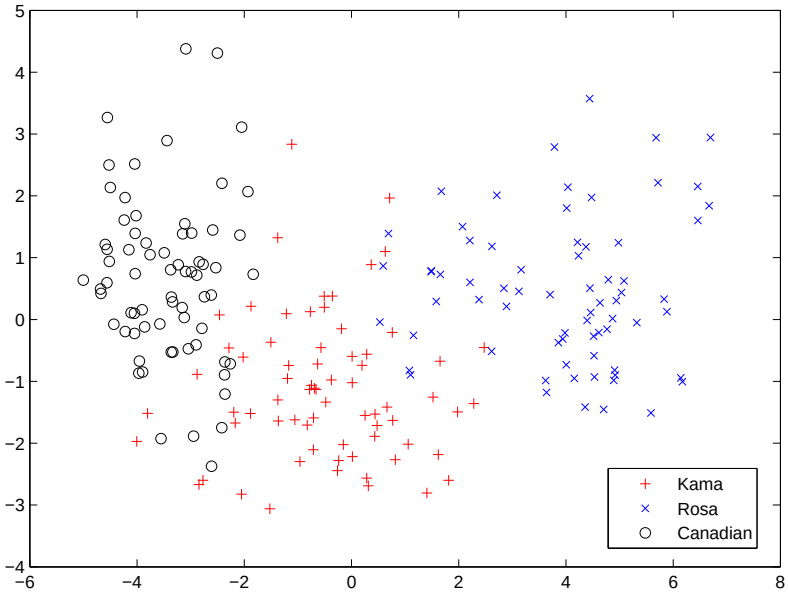


Figure 5.9 – PCA representation of seeds dataset.

wheat variety and the Rosa wheat variety while target are the Canadian wheat variety and the Kama wheat variety ; SEED-Task 2 - the Canadian wheat variety and the Rosa wheat variety are source data while the Kama wheat variety and the Rosa wheat variety are the target. Figure 5.9 shows the PCA representation of seeds dataset. As shown in the figure, we use an easier classification task as source while a more difficult classification task as target.

5.6.2 Experimental Results

In Table.5.2, we compare the results obtained with our methods and that with LM (the method proposed in [102]). In [102], LM has been proved superior to some other related transfer learning methods (T-SVM in [64], CDSC in [73], LWE in [48]), so we omit here the comparison to these methods. Other domain adaptation methods are also included in the comparison, even though some of the methods have little relatedness with ours. From this general comparison, we show readers that, for some data sets, our methods give similar or even better results. Included methods are TCA [93], GFK [53], JDA [75]. Standard SVM is also compared to show the usefulness of transfer learning (or cases when negative transfer happens). All parameters are tuned to be optimal for every method.

5.6.3 Comparison to other state-of-the-art methods

We first compare our methods to standard SVM. All of our methods, namely SVMMMD, KPCA-TL, KPCA-TL-LT, KPCAlin, perform better than standard SVM. In other words, in all real datasets tested, no negative transfer happens. However, we can find that for SEED-Task 1,

11. Results obtained after an inversion of labels, read the comment in the text.

Table 5.2 – Good classification results on different datasets for different transfer learning methods.

	USPS-Task 1	SEED-Task 1	SEED-Task 2	IRIS-Task 1	IRIS-Task 2
SVM	0.6976	0.8000	0.7000	0.5000	0.5000
TCA	0.7347	0.7286	0.8214	0.8800	0.5000
JDA	0.8329	0.7786	0.7714	0.5000	0.5100
GFK	0.7560	0.7857	0.7214	0.5000	0.5000
LM	0.7294	0.8929	0.8571	0.7000	0.7300
SVMMMD	0.9496	0.9071	0.9357	0.8700	0.9500
KPCA-TL	0.8117	0.9357	0.9214	**0.9000** [11]	0.9300
KPCA-TL-LT	0.8143	0.9214	0.9143	**0.9300** [11]	0.9700
KPCAlin	0.8554	0.9143	0.9429	0.9100	0.9200

TCA, JDA, GFK get worse performance than standard SVM, which proves that such methods suffer from negative transfer learning.

We then notice that in IRIS-Task 1, there is an inversion of labels for KPCA-TL and KPCA-TL-LT. With few labeled target data provided, both KPCA-TL and KPCA-TL-LT can achieve excellent performances. The problem probably comes from the fact that positive and negative classes after KPCA are very similar to each other ; thus, the mismatch can happen.

From the results, SVMMMD performs the best in USPS-Task 1. Other USPS transfer learning tasks have also been tested and SVMMMD achieves always the best. However, in other tasks in Table 5.2, SVMMMD performs as well as other KPCA based approaches or worse than some KPCA based approaches. Thus, SVMMMD is probably biased for USPS dataset.

For KPCA based datasets, KPCA-TL, KPCA-TL-LT and KPCAlin generally perform the best among all approaches. In 3/5 tasks, KPCA-TL-LT improves the performance of KPCA-TL. For the other 2 tasks, perhaps, even after KPCA alignment, the relationship between source and target data is nonlinear ; thus applying linear transformation after KPCA-TL sometimes does not work. Another cause may be that the optimization finding the optimum linear transformation is trapped in local minimum. Other better optimization strategies are to be developed. KPCAlin is the most robust and the most efficient (analysed in Table 5.1, in Section 5.5) approach. The mismatch problem in IRIS-Task 1 is avoided by using KPCAlin.

In general, our proposed transfer leaning approaches return better results than other state-of-the-art approaches. Of course, other more complicated real datasets are to be tested.

5.6.4 Analysis of parameters

For SVMMMD, we use cross validation to select the kernel parameter. The parameter should not be too small, or else, overfitting may happen. It should neither be too large, or else, the resulted gaussian kernel performs like a linear kernel and its nonlinearity aspect will be lost. The same observations can be found in other state-of-the-art approaches, for example SVM, TCA, LM.

For KPCA based transductive transfer learning approaches, the tuning of parameters is much more delicate than SVMMMD. Details of tuning parameters can be found in Section 5.5.3, Chapter 5. Importance parameters include the KPCA kernel parameter (σ_{KPCA}), the number of principal axes used(l), the kernel parameter for MMD alignment (σ_{MMD}). By experiments, all KPCA based approaches are not very sensible to σ_{MMD}. However, the selection of pair (σ_{KPCA}, l) is crucial to good experimental results. Followed the strategy proposed in Section 5.5.3, Chapter 5, the selection of (σ_{KPCA}, l) can generally lead to good performances, even for using a relatively small number of principal axes (l).

5.7 Conclusion

In this chapter, we first reviewed related domain adaptation literatures : dimension reduction related domain adaptation (in Section 5.2.1), subspace alignment domain adaptation (in Section 5.2.2) and kernel space alignment related domain adaptation (in Section 5.2.3). We especially detailed *Robust Transfer Principal Component Analysis* (in Section 5.2.4), based on which the perspective of our research is proposed.

Domain adaptation by KPCA coordinate system alignment (KPCA-TL in Section 5.3.2 and KPCA-TL-LT in Section 5.3.3) and domain adaptation by *linearly* kernel space alignment (KPCAlin in Section 5.4) are our main contributions on KPCA domain adaptation. Different from other subspace alignments, we suppose nonlinear relationship between source and target basis as well as their representations therein. We align simultaneously the KPCA basis and the kernel representations by minimizing MMD in a higher-order RKHS. Then, we suppose that a linear transformation can further align source and target data in KPCA subspace. However, taking into consideration the permutations and inversions of KPCA axes greedily is time consuming. We then propose KPCAlin : the optimization problem becomes a scalar optimization w.r.t *kernel parameter* M ; it is also equivalent to finding a linear transformation M (in the original input space) that nonlinearly aligns source and target data in KPCA subspace. In this way, the alignment of KPCA bases and the alignment of the distributions in the RKHS are taken into account implicitly by finding the optimum M. The similarity measure remains the MMD. When the number of KPCA axes augments, KPCAlin becomes much more efficient than KPCA-TL and KPCA-TL-LT.

In Section 5.5 and Section 5.6, experimental results showed the efficiency and the robustness of our approaches. In real datasets, KPCAlin is more robust than KPCA-TL and KPCA-TL-LT. Morevoer, in Section 5.5, we proposed effective strategies for selecting parameters. Obviously,

other new and simple ways of setting parameters are to be developed.

Chapter 6

Conclusion and Perspectives

6.1 Conclusion

This thesis contributes to homogeneous transductive transfer learning. After a presentation of covariate shift, the most simple homogeneous transductive transfer learning approach, we relax the covariate shift assumption to be more and more general. Within these assumptions, several contributions have been proposed :

- In Chapter 3, we propose a relaxed transductive transfer learning approach : it probabilistically solves the transfer learning problem by maximizing a likelihood criterion. It is simple to be implemented. However, its optimization is non convex and there are many local minimums.

- In Chapter 4, we then consider to use a non-parametric criterion to solve the problem. Maximum Mean Discrepancy (MMD) is one of such criteria. It measures the similarity of two distributions with no parameters needed. Furthermore, it can be kernelized and easily calculable. Therefore, we integrate MMD into SVM (Chapter 4 : SVMMMD). But unlike previous literatures, we use MMD as a constraint instead of a regularization term. In this way, we focus on transfer, which can be considered as an extreme case of its regularized counterpart. Experimental results have shown that the constraint SVMMMD is better than its regularized counterpart.

  Another novelty of SVMMMD is our way of manipulating data in a RKHS. As usual, we can transform data into higher dimensional RKHS by kernel methods (kernel based SVM). We then show that our constraint projects data onto a subspace of the RKHS which remains a RKHS.

  Furthermore, the optimization is convex and there is unique solution to the objective function. Compared to other MMD-based SVM transfer learning approaches, our approach avoids the calculation of the inverse of a matrix, which makes our approach more efficient.

  Compared to former proposed approach, SVMMMD can be applied to more general cases and it is more efficient.

- In Chapter 5, as SVMMMD sometimes fails, to avoid the failure, we propose to explicitly

control the alignment in a RKHS : KPCA technique can be applied.

- Our first KPCA based transfer learning approach (KPCA-TL) proposes to align both coordinate systems and distributions of source and target data. Different from other literatures, the alignment is nonlinear : MMD is applied when aligning the KPCA representations of source and target.

  However, to align the coordinate systems of source and target, we need to calculate MMD for every possible match of axes, which is tedious, especially when the number of retained axes augments.

- Our second KPCA based transfer learning approach (KPCA-TL-LT) further aligns the resulting matched KPCA representations of source and target by a linear transformation. The result of KPCA-TL can be considered as a good starting point of the parametric non convex optimization of KPCA-TL-LT. The found linear transformation matrix is expected to be reliable. However, more delicate and reliable optimization strategy is to be developed.

- To avoid the computation burden of KPCA-TL and KPCA-TL-LT, our third approach KPCAlin is proposed. To align the KPCA representations of source and target, KPCA-lin tries to find a linear transformation (a change of representation basis) in the original space that renders source and target to be similar in the KPCA subspace. We have proved that KPCA after such linearly transformed original data is equivalent to projecting the untransformed original data to a different KPCA subspace. In other words, we use different kernel parameters to project source and target into their KPCA subspaces. On modifying the kernel parameter of target KPCA, we are expected to find a suitable target KPCA subspace that aligns with the source KPCA subspace.

  For KPCAlin, the optimization is efficient.

6.2 Perspectives

Extension to Multi-class Problems

All our contributions are designated to solve binary classification tasks. Extension from binary classification to multi-class classification should be considered.

- For SVMMMD, the *one-against-one* or *one-against-all* multi-class strategies seemed to be possible choices. However, we have found that at least for the USPS dataset, both strategies failed. More delicate extension strategy should be developed.

- For KPCA based transfer learning approaches, the most intuitive way of extending binary tasks to multi-class tasks may be to introduce multi-class kernel methods on the KPCA subspace.

Extension to Heterogeneous Transfer Learning

Another interesting orientation may be extending our work to heterogeneous transfer learning. For homogeneous transfer learning, the feature space is shared between source and target. When the feature spaces differ from source to target, our works are not expected to perform well. For example, the difference of feature space can be moderated by taking advantage of the co-occurrence information between source and target. How to use such information and cover the gap in feature space might be one of our future focus.

Evaluation Criteria

In this thesis, we considered that there was no available label information in target domain. We observed possible confusion between classes. How to evaluate the performance on target data remains unclear under such a transfer learning context. Accordingly, new reliable evaluation criteria should be proposed.

Annexes

1 Annexes for Chapter 4

1.1 Dual Form of KMM

Lagrangian of KMM is :

$$\mathcal{L} = \frac{1}{||w||^2} + C\sum_{i=1}^{n_s}\beta_i\xi_i - \sum_{i=1}^{n_s}\gamma_i\xi_i - \sum_{i=1}^{n_s}\alpha_i[y_i(< w, \phi(x_i) > +b) + \xi_i - 1]$$

with γ and α being Lagrangian multipliers. Then we can deduce the dual form :

$$\begin{aligned}
&\min_{\alpha} \sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\alpha_i\alpha_j y_i y_j K_{SS}(i,j) - \sum_{i=1}^{n_s}\alpha_i \\
&\text{s.t.} \sum_{i=1}^{n_s}\alpha_i y_i = 0,\ \forall i = 1,\ldots,n_s \\
&\quad 0 \leq \alpha_i \leq C\beta_i,\ \forall i = 1,\ldots,n_s
\end{aligned}$$

where $K_{SS}(i,j) =< \phi(x_s^i), \phi(x_s^j) >$.

1.2 Final Optimization Problem of LM and its Dual Form

With $w = \sum_{i=1}^{n_s+n_t} \alpha_i \phi(X_i)$, we obtain $Dist(P_s, P_t)^2$ w.r.t α as :

$$
\begin{aligned}
Dist(P_s,P_t)^2 =& || \frac{1}{n_s} \sum_{i=1}^{n_s} < w, \phi(x_i^s) > - \frac{1}{n_t} \sum_{j=1}^{n_t} < w, \phi(x_j^t) > ||^2 \\
=& \frac{1}{n_s^2} (\sum_{i=1}^{n_s} < w, \phi(x_i^s) >)^2 + \frac{1}{n_t^2} (\sum_{j=1}^{n_t} < w, \phi(x_j^t) >)^2 \\
& - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} < w, \phi(x_i^s) >< w, \phi(x_j^t) > \\
=& \frac{1}{n_s^2} \alpha^T [\sum_{i,j=1}^{n_s} \phi(X)^T \phi(x_i^s) \phi(x_j^s)^T \phi(X)] \alpha + \frac{1}{n_t^2} \alpha^T [\sum_{i,j=1}^{n_t} \phi(X)^T \phi(x_i^t) \phi(x_j^t)^T \phi(X)] \alpha \\
& - \frac{2}{n_s n_t} \alpha^T [\sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \phi(X)^T \phi(x_i^s) \phi(x_j^t)^T \phi(X)] \alpha \\
=& \frac{1}{n_s^2} \alpha^T < \phi(X), \phi(x_s) > [1]^{n_s \times n_s} < \phi(X), \phi(x_s) >^T \alpha \\
& + \frac{1}{n_t^2} \alpha^T < \phi(X), \phi(x_t) > [1]^{n_t \times n_t} < \phi(X), \phi(x_t) >^T \alpha \\
& - \frac{1}{n_s n_t} \alpha^T (< \phi(X), \phi(x_s) > [1]^{n_s \times n_t} < \phi(X), \phi(x_t) >^T \\
& \qquad + < \phi(X), \phi(x_t) > [1]^{n_t \times n_s} < \phi(X), \phi(x_s) >^T) \alpha \\
=& \alpha^T \Omega \alpha
\end{aligned}
$$

where $[1]^{n \times m}$ denotes a $n \times m$ matrix of all ones and Ω is a symmetric positive semidefinite matrix of $(n_s + n_t) \times (n_s + n_t)$.

The objective function then becomes :

$$
\begin{aligned}
& \min_{\alpha,\xi,b} \alpha^T (\frac{1}{2} < \phi(X), \phi(X) > + \lambda \Omega) \alpha + C \sum_{i=1}^{n_s} \xi_i \\
& \text{s.t. } \xi_i \geq 0 \text{ and } y_i (\alpha^T < \phi(X), \phi(x_s^i) > + b) \geq 1 - \xi_i, \ \forall i = 1, \ldots, n_s
\end{aligned}
$$

Dual form of the above LM problem is :

$$
\begin{aligned}
& \min_{\gamma} \frac{1}{2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \gamma_i \gamma_j y_i y_j < \phi(x_s^i), \phi(X) > (\frac{1}{2} < \phi(X), \phi(X) > + \lambda \Omega)^{-1} < \phi(x_s^j), \phi(X) >^T - \sum_{i=1}^{n_s} \gamma_i \\
& \text{s.t. } \sum_{i=1}^{n_s} \gamma_i y_i = 0, \text{ and } 0 \leq \gamma_i \leq C, \ \forall i = 1, \ldots, n_s
\end{aligned}
$$

2 Annexes for Chapter 5

2.1 Graph Laplacian M

There are many ways to associate a matrix with a graph. Here, we introduce the commonly used unnormalized Laplacian matrix in transfer learning literatures.

Let $\mathcal{G} = (V, E)$ be the graph, with V being the ensemble of vertex and E being the ensemble of edges. Given data points $x_i, x_j \in \mathcal{X}$, x_i, x_j can be considered as vertex. Let w_{ij} denote a weight of edge (that connects x_i and x_j) : it can represent the similarity between x_i and x_j ; if $w_{ij} = 0$, there is no edge between x_i and x_j. The degree of a vertex x_i is $d_i = \sum_{j=1}^{n} w_{ij}$. A graph Laplacian matrix M is a matrix $M = D - W$ ($W(i,j) = w_{ij}$ and $D(i,i) = d_i$, $D(i,j) = 0, i \neq j$) that satisfies the following properties (from [147]) :

- *For every vector $f \in \mathbb{R}^n$, we have*

$$f^T M f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

- *M is symmetric and positive semi-definite*
- *The smallest eigenvalue of M is 0, the corresponding eigenvector is the constant one vector.*
- *M has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$*

In transfer learning context, by applying $f^T M f$ as a regularization term, certain structural properties of source data (source graph) can be more or less preserved and transferred to target domain.

2.2 Definition of Surrogate Kernel

Definition ([169]) : Let $\mathcal{X}$ and $\mathcal{Z}$ be two samples, and $k(.,.)$ be a kernel function. $K_{\mathcal{X}} \in \mathcal{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ and $K_{\mathcal{Z}} \in \mathcal{R}^{|\mathcal{Z}|\times|\mathcal{Z}|}$ denotes the kernel matrix defined on $\mathcal{X}$ and $\mathcal{Z}$, respectively. $K_{\mathcal{X}\mathcal{Z}} \in \mathcal{R}^{|\mathcal{X}|\times|\mathcal{Z}|}$ denotes the kernel matrix defined among $\mathcal{X}$ and $\mathcal{Z}$. Then, the surrogate kernel of $K_{\mathcal{X}}$ on sample $\mathcal{Z}$, denoted by $\mathcal{K}_{\mathcal{Z}\leftarrow\mathcal{X}}$ is defined as :

$$\mathcal{K}_{\mathcal{Z}\leftarrow\mathcal{X}} = K_{\mathcal{Z}\mathcal{X}} K_{\mathcal{X}}^{-1} K_{\mathcal{X}\mathcal{Z}}$$

2.3 Nystrom Kernel Approximation ([113])

If the conditions of Mercer's theorem are satisfied, then Mercer kernel k can be diagonalized as :

$$k(x, x') = \sum_{j=1}^{l} \lambda_j \psi_j(x) \psi_j(x')$$

where $\psi_j, j = 1, \ldots, l$ are orthonormal and satisfies the eigenvalue equation :

$$\sum_{i=1}^{n} \frac{1}{n} k(x, x') \psi_j(x) \approxeq \lambda_j \psi_j(x')$$

Résumé en français

1 Introduction

L'apprentissage automatique a pour objectif de faire apprendre automatiquement un modèle prédictif à partir de données existantes. Ces données peuvent être associées à une valeur de la fonction à prévoir (étiquette dans le cas de la classification qui nous concerne ici) ou non. En fonction de la disponibilité de l'étiquette, l'apprentissage automatique est appelé supervisé, non-supervisé ou semi-supervisé. Les tâches à apprendre peuvent être très diverses : classification, regression, réduction de dimension etc.

Concernant la classification, l'apprentissage automatique a beaucoup évolué avec l'introduction des réseaux de neurones artificiels et des méthodes à noyau dont les SVM (Machines à Vecteurs de Support) sont le représentant essentiel. Pour les méthodes classiques d'apprentissage, les données futures sont supposées issues de la même distribution de probabilité que les données utilisées lors de l'apprentissage.

Nous nous intéressons ici à un cas différent : le transfert d'apprentissage, dont l'objet est de traiter des données futures qui possèdent des caractéristiques différentes de celles des données ayant servi à l'apprentissage.

1.1 Aperçu du transfert d'apprentissage

Le transfert d'apprentissage a suscité une grande attention de la part de la communauté scientifique à partir de 1995. Il diffère de l'apprentissage automatique traditionnel par sa capacité à traiter des données issues de différentes lois de probabilité, ne possédant pas le même espace de représentation etc. Ainsi, l'objectif du transfert d'apprentissage est d'utiliser au mieux les données disponibles et les relations entre données disponibles et données futures pour réaliser l'apprentissage en assurant la capacité à généraliser aux données futures. L'ensemble des données d'apprentissage est appelé Source et l'ensemble des données futures est appelé Cible ; les caractéristiques possibles de chacun de ces jeux sont :

- leur domaine $\mathcal{D} : (\mathcal{X}, P(\mathcal{X}))$ où $\mathcal{X}$ désigne l'espace des caractéristiques and $P(\mathcal{X})$ désigne la distribution marginale des observations.

- leur tâche $\mathcal{T} : (\mathcal{Y}, P(\mathcal{Y}|\mathcal{X}))$ où $\mathcal{Y}$ désigne les étiquettes et $P(\mathcal{Y}|\mathcal{X})$ désigne la distribution des étiquettes conditionnellement aux observations.

La différence entre Source et Cible peut se trouver dans les domaines, dans les tâches ou les deux simultanément. A partir du type de différence, une taxonomie de l'apprentissage peut être proposée :

- *transductive transfer learning* : la différence se trouve dans les domaines, les données Source et les données Cible partagent le même espace des caractéristiques et diffèrent dans leur distribution marginale (*homogeneous transfer learning*) ou elles possèdent des caractéristiques différentes (*heterogeneous transfer learning*).

- *inductive transfer learning* : la différence se trouve dans les tâches, les données partagent le même type d'étiquette et les distributions de celles-ci conditionnellement aux observations diffèrent ou les types d'étiquettes diffèrent. On ne parle pas ici de la disponibilité ou non des étiquettes.

- *implicit transfer learning* : la différence se trouve à la fois dans les domaines et dans les tâches, mais il existe des liens entre Source et Cible. En fait, si l'on apprend à partir de Sources n'ayant aucun rapport avec les Cibles, on ne peut espérer transférer l'apprentissage effectué.

Le transfert d'apprentissage peut aussi faire l'objet d'une taxonomie reposant sur le contenu à transférer : les observations, les paramètres d'un algorithme...

Dans le cadre de ces travaux, nous nous focalisons sur le *homogeneous transductive transfer learning*, en particulier pour la classification lorsqu'il n'y a aucune étiquette pour les Cibles.

1.2 Apprentissage Homogène (homogeneous transductive transfer learning)

Pour ce type d'apprentissage, les domaines Ssource et Cible ne diffèrent que par la loi marginale des observations et les tâches des données Source et Cible sont identiques (elles partagent le même $\mathcal{Y}$). Elles peuvent différer par la loi des étiquettes conditionnellement aux observations (*domain adaptation*) ou non (*covariate shift*).

Covariate shift

Reposant sur l'égalité des lois des étiquettes conditionnellement aux observations, le *covariate shift* peut se traiter, au moins conceptuellement, très facilement.

$$\begin{aligned}
\theta^* &= \arg\min_{\theta\in\Theta} E_{P_T(x,y)}[l(x,y,\theta)] \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} l(x,y,\theta)P_T(x,y)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} P_T(y|x)P_T(x)l(x,y,\theta)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} P_S(y|x)P_T(x)l(x,y,\theta)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} \frac{P_T(x)}{P_S(x)}(P_S(y|x)P_S(x))l(x,y,\theta)dxdy \\
&= \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} \frac{P_T(x)}{P_S(x)}P_S(x,y)l(x,y,\theta)dxdy
\end{aligned}$$

Il convient de noter que le support de $P_T(.)$ et celui de $P_S(.)$ doivent permettre de définir le quotient de ces grandeurs en tout point de calcul. Nous remplaçons le risque par le risque empirique :

$$\hat{\theta}^* \sim \arg\min_{\theta\in\Theta} \frac{1}{n_s} \sum_{(x_i,y_i)\in\mathcal{S}} \frac{P_T(x_i)}{P_S(x_i)} l(x_i,y_i,\theta)$$

On voit que la solution est obtenue simplement par pondération des coûts individuels sur les données Source (par $\frac{P_T(x_i)}{P_S(x_i)}$). Cette approche est appelée *importance sampling*.

L'apprentissage du détecteur sur les données Cible repose donc sur l'apprentissage du détecteur sur les Source et le transfert se réalise par la pondération $\frac{P_T(x_i)}{P_S(x_i)}$ du coût associé à chaque observation des données Source.

Toutefois, pour une majorité d'applications réelles, l'égalité de la loi des étiquettes conditionnellement aux observations n'est pas toujours vérifiée. Nous relâchons cette contrainte dans la Section 2 de ce chapitre.

Domain adaptation

Par comparaison avec le *covariate shift*, le *domain adaptation* ne fait pas l'hypothèse d'égalité des lois des étiquettes conditionnellement aux hypothèses. Pour aborder ce type de problème, l'idée générale repose sur la recherche d'un espace de caractéristiques où les données Source et Cible deviennent similaires.

Certains auteurs proposent d'augmenter le nombre de caractéristiques en prenant en compte des caractéristiques spécifiques aux données Source et Cible et des caractéristiques communes.

Certains auteurs proposent de construire un dictionnaire qui relie les Source et Cible par les "mots" utilisés dans tous les deux domaines.

D'autres projettent les données Source et Cible dans un sous-espace en commun où la similitude entre Source et Cible est maximale.

Dans la Section 3 et la Section 4 de ce chapitre, nous cherchons à améliorer les méthodes existantes en s'inspirant de cette dernière idée.

2 Covariate Shift Étendu

Nous proposons tout d'abord de traiter une variante du problème de *covariate shift*, l'approche proposée ne nécessite pas la condition très stricte d'égalité des lois des étiquettes conditionnellement aux observations. Nous émettons l'hypothèse plus large suivante :

$$\exists A, b \ : \ p_S(y|x, x \in \mathcal{S}) = p_T(y|Ax + b, x \in \mathcal{T}) \tag{1}$$

où $\mathcal{S}$ ($\mathcal{T}$) représente les données Source (Cible) et A, b sont les paramètres qui contrôlent la transformation.

Sous cette condition, si les lois marginales $p_S(x, x \in \mathcal{S})$ et $p_T(Ax+b, x \in \mathcal{T})$, deviennent semblables (après avoir transporté la distribution des données Cible vers celle des données Source), et sous l'hypothèse précédente, la détection sera possible sur les données Cible. Pour estimer A et b nous utilisons la méthode du Maximum de Vraisemblance (MV) :

$$(A^*, b^*) = \arg\max_{A,b} \prod_{x_t^i \in \mathcal{D}_T} p_S(Ax_t^i + b|A, b)$$

où $\mathcal{D}_T$ représente le domaine Cible.

En utilisant un estimateur de densité (de type Parzen, par exemple), l'implémentation est simple. Malheureusement, l'optimisation est non-convexe et les expériences menées ont montré que la recherche de l'optimum global était irréalisable. Néanmoins, l'idée du transport de la densité des observations Cible pour la rendre similaire à celle des données Source reste l'idée générale de la suite de nos travaux, sachant qu'il reste à proposer un critère de similitude entre deux lois de probabilité permettant une optimisation aisée.

3 Domain Adaptation avec SVM sous contrainte basée sur la MMD

Nous avons vu qu'il était possible de relâcher la contrainte sur l'égalité des distributions des étiquettes conditionnellement aux observations dans la Section 2. Toutefois la transformation $Ax + b$ ne permet pas de rendre compte d'une modification non linéaire des données Cible par rapport aux Source. Nous allons donc considérer une transformation non linéaire entre Source et Cible et émettre l'hypothèse plus générale suivante :

$$\exists g(.) : p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$$

où $g(.)$ est une fonction suffisamment régulière. Transférer aux données Cible l'apprentissage réalisé sur les données Source devient le problème de la recherche d'une fonction $g(.)$ permettant de rendre similaires les distributions marginales $p_S(x, x \in \mathcal{S})$ et $p_T(g(x), x \in \mathcal{T})$. Il convient de noter que si l'hypothèse précédente n'est pas respectée, le transport de la loi marginale des observations Cible vers celle des Source ne permet pas une bonne détection sur les Cible. La figure

1 montre un cas où l'hypothèse précédente n'est pas respectée et nos approches ne peuvent plus fonctionner.

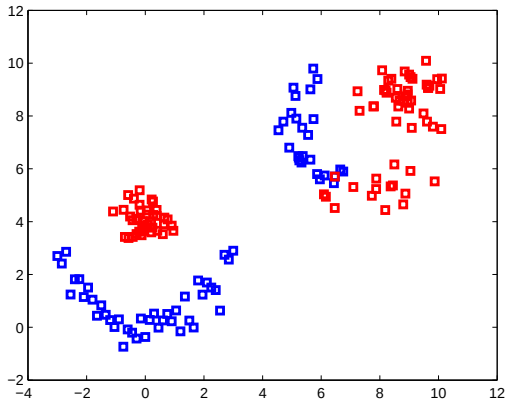


Figure 1 – Un exemple où l'hypothèse $\exists g(.) : p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$ n'est pas vérifiée. Les Source se trouvent en bas à gauche et les Cible en haut à droite. Les couleurs des observations représentent la classe. Même si nous concevons aisément l'existence d'une transformation permettant de superposer les distributions, l'application aux donnée Cible du détecteur obtenu sur les Source ne fournira pas les résultats escomptés car l'hypothèse émise en début de section n'est pas vérifiée

Avant de présenter cette approche et son évolution au cours de nos travaux, nous rappelons les fondamentaux qui seront utilisés. Nous présentons ensuite l'état de l'art et nous en extrairons quelques méthodes de référence. Enfin, nous comparerons nos travaux avec ceux issus de la littérature.

3.1 Outils fondamentaux

Machine à Vecteurs de Support (SVM)

Une Machine à Vecteurs de Support (SVM) est un outil de classification supervisée, en général présenté dans un premier temps pour traiter le cas de la détection linéaire, qui sépare les données selon le critère de marge maximum, garantissant de bonnes propriétés en généralisation. Grâce à l'introduction des méthodes à noyaux, les SVM se généralisent aisément au cas non linéaire. L'optimisation d'une SVM à marge souple (permettant de traiter le cas avec mélange) se formalise de la manière suivante :

$$\begin{aligned} \min_{w,\epsilon,b} \ & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\epsilon_i \\ \text{s.t. } & y_i(< w, \phi(x_i) > +b) \geq 1 - \epsilon_i, \forall i = 1, \ldots, n \\ & \epsilon_i \geq 0, \forall i = 1, \ldots, n \end{aligned}$$

où w est un paramètre de marge ; ϵ représente l'erreur commise ($\epsilon \geq 0$), C est le paramètre qui contrôle le compromis entre l'erreur de classification et la marge ; x_i est une observation dont l'étiquette est y_i ; $\phi(.)$ représente la transformation non linéaire des données, b représente le biais.

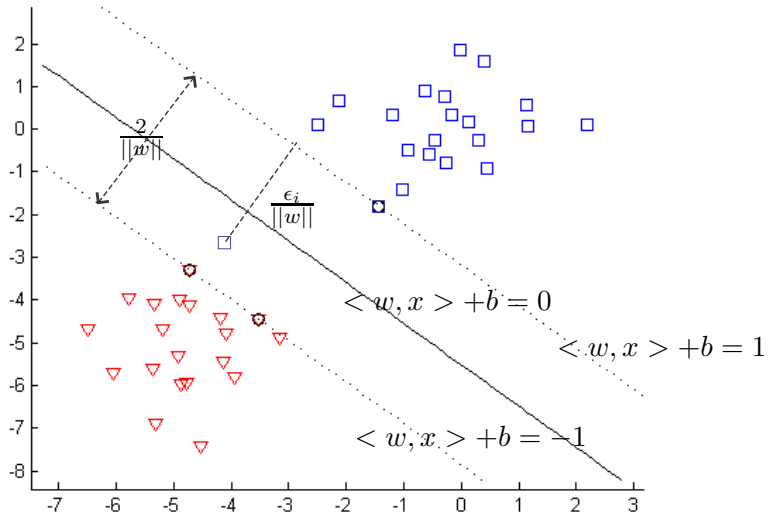


Figure 2 – Illustration d'une SVM à marge souple avec une transformation linéaire. Les triangles rouges et les carrés bleus représentent des différentes classes. Les droites vérifiant, $< w, x > +b = 1$ et $< w, x > +b = -1$ représentent les marges et $< w, x > +b = 0$ est la courbe de décision.

Pour une illustration géométrique, voir Figure 2.

Pour prédire l'appartenance d'une nouvelle observation (x), il suffit de calculer $sign(< w, x > +b)$. L'utilisation de l'astuce du noyau permet aisément l'extension des SVM au cas d'un détecteur non linéaire.

Maximum Mean Discrepancy (MMD)

La *Maximum Mean Discrepancy* (MMD) est une mesure de la similitude entre deux distributions. Elle est définie ([55]) par :

Soit $\mathcal{F}$ est un ensemble des fonctions $f : \mathcal{X} \to \mathbb{R}$ et x (z) une observation tirée de la distribution p (q), la Maximum Mean Discrepancy (MMD) est :

$$MMD[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} (E_p[f(x)] - E_q[f(y)])$$

où $E_{\{p,q\}[.]}$ représente l'espérance par rapport à la distribution p (q).

$MMD[\mathcal{F}, p, q] = 0$ si et seulement si les distributions p et q sont égales.

Dans [14], il est démontré que la MMD peut être aisément évaluée dans un Espace de Hilbert à Noyau Reproduisant (RKHS : $\mathcal{H}$) :

$$MMD = ||\mu_p - \mu_q||_{\mathcal{H}}$$

où $\mu_{\{p,q\}} = E_{\{p,q\}}[k(x,.)]$ et $k(x,.)$ est la représentation de l'observation x (iid tirée de p ou q) dans un RKHS ($\mathcal{H}$). Pour ce faire, le noyau $k(x,.)$ doit être *universel*[12].

Théorème ([130] et [127]) $MMD[\mathcal{F},p,q] = 0$ si et seulement si $p = q$ lorsque $\mathcal{F} = f : ||f||_{\mathcal{H}} \leq 1$ où $k(.,.)$ est universel.

Le transport de la densité des observations Cible vers celle des Source peut être réalisé par la recherche d'un espace de représentation des données où la MMD devient nulle (ou est minimale). Ceci nous permet de poursuivre.

En outre, avec l'astuce de noyau, le carré de la MMD peut s'écrire :

$$\begin{aligned} MMD^2[\mathcal{F},p,q] &= ||\mu_p - \mu_q||^2_{\mathcal{H}} \\ &= E_{p,p}[k(x,x')] - 2E_{p,q}[k(x,y)] + E_{q,q}[k(y,y')] \end{aligned}$$

où x et x' (y et y') sont des observations iid issues de la distribution p (q) et k désigne un noyau universel.

Il nous faut maintenant rendre la MMD calculable. Il est aisé de proposer un estimateur sans biais de celle-ci [118] :

$$\widehat{MMD}^2_u[\mathcal{F},X,Y] = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m} k(x_i,x_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(y_i,y_j) - \frac{2}{nm}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i,y_j)$$

où $x_i, i = 1,\ldots,m$ et $y_i, i = 1,\ldots,n$ sont des observations iid issues de p et q, respectivement.

3.2 *Domain Adaptation* par SVM sous contrainte de nullité de la MMD (SVMMMD)

Nous pouvons coupler les SVM et la MMD pour réaliser le transfert par la résolution du problème suivant :

$$\begin{aligned} &\min_{w,b,\epsilon} ||w||^2 + C\sum_{i=1}^{n_s}\epsilon_i \\ &\text{s.t.} y_i(< w,\phi(x^i_s) > +b) \geq 1-\epsilon_i, \forall i = 1,\ldots,n_s \\ &\quad \epsilon_i \geq 0, \forall i = 1,\ldots,n_s \\ &\quad < w, \mu_s - \mu_t >= 0 \end{aligned} \tag{2}$$

où la contrainte supplémentaire exprime le fait que le sous espace $\mathcal{H}'$ (sous espace de $\mathcal{H}$ orthogonal à w, qui reste un RKHS [97]), dans lequel est recherchée la solution, garantit que $||\mu_p - \mu_q||_{\mathcal{H}'} = 0$.

Nous pouvons donc trouver un sous-espace où Source et Cible sont rendus similaires. Sous l'hypothèse $\exists g(.) : p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$, la SVM apprise sur les données Source fonctionnera sur les données Cible. L'intérêt de cette approche est qu'elle conduit à un problème d'optimisation quadratique sur un domaine convexe, tout comme les SVM standards, et que l'optimum obtenu est unique. Cette démonstration fait l'objet de la section suivante.

12. Voir [129] pour la définition d'un noyau universel.

Optimization

Pour résoudre le Problème 2, nous utilisons le théorème du représentant. w s'exprime alors :

$$w = \sum_{k=1}^{n_s} \beta_k^s \phi(x_k^s) + \sum_{l=1}^{n_t} \beta_l^t \phi(x_l^t)$$

$< w, \mu_{X_s} - \mu_{X_t} >_{\mathcal{H}} = 0$ devient :

$$\begin{aligned}
< w, \mu_{X_s} - \mu_{X_t} >_{\mathcal{H}} =& < \sum_{k=1}^{n_s} \beta_k^s \phi(x_k^s) + \sum_{l=1}^{n_t} \beta_l^t \phi(x_l^t), \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_j^t) >_{\mathcal{H}} \\
&= \frac{1}{n_s} \sum_{k=1}^{n_s} \beta_k^s \sum_{i=1}^{n_s} < \phi(x_k^s), \phi(x_i^s) >_{\mathcal{H}} - \frac{1}{n_t} \sum_{k=1}^{n_s} \beta_k^s \sum_{j=1}^{n_t} < \phi(x_k^s), \phi(x_j^t) >_{\mathcal{H}} \\
&+ \frac{1}{n_s} \sum_{l=1}^{n_t} \beta_l^t \sum_{i=1}^{n_s} < \phi(x_l^t), \phi(x_i^s) >_{\mathcal{H}} - \frac{1}{n_t} \sum_{l=1}^{n_t} \beta_l^t \sum_{j=1}^{n_t} < \phi(x_l^t), \phi(x_j^t) >_{\mathcal{H}} \\
&= (\begin{bmatrix} K_{SS} & K_{TS} \\ K_{ST} & K_{TT} \end{bmatrix} [\underbrace{\frac{1}{n_s}, \ldots, \frac{1}{n_s}}_{n_s}, \underbrace{-\frac{1}{n_t}, \ldots, -\frac{1}{n_t}}_{n_t}]^T)^T [\beta^s, \beta^t]^T \\
&= (K\tilde{1})^T \beta
\end{aligned}$$

où $K_{SS} = < \phi(x_k^s), \phi(x_i^s) >_{\mathcal{H}}$, $K_{TS} = < \phi(x_j^t), \phi(x_k^s) >_{\mathcal{H}}$, $K_{ST} = < \phi(x_i^s), \phi(x_l^t) >_{\mathcal{H}}$, $K_{TT} = < \phi(x_j^t), \phi(x_l^t) >_{\mathcal{H}}$.

Nous pouvons donc exprimer $||w||^2$:

$$\begin{aligned}
||w||^2 =& < \sum_{k=1}^{n_s} \beta_k^s \phi(x_k^s) + \sum_{l=1}^{n_t} \beta_l^t \phi(x_l^t), \sum_{k=1}^{n_s} \beta_k^s \phi(x_k^s) + \sum_{l=1}^{n_t} \beta_l^t \phi(x_l^t) >_{\mathcal{H}} \\
&= \sum_{k=1}^{n_s} \sum_{k'=1}^{n_s} \beta_k^s < \phi(x_k^s), \phi(x_{k'}^s) >_{\mathcal{H}} \beta_{k'}^s + 2 \sum_{k=1}^{n_s} \sum_{l=1}^{n_t} \beta_k^s < \phi(x_k^s), \phi(x_l^t) >_{\mathcal{H}} \beta_l^t \\
&+ \sum_{l=1}^{n_t} \sum_{l'=1}^{n_t} \beta_l^t < \phi(x_l^t), \phi(x_{l'}^t) >_{\mathcal{H}} \beta_{l'}^t \\
&= [\beta^s, \beta^t] \begin{bmatrix} K_{SS} & K_{TS} \\ K_{ST} & K_{TT} \end{bmatrix} [\beta^s, \beta^t]^T \\
&= \beta^T K \beta
\end{aligned}$$

Le Problème 2 devient :

$$\begin{aligned}
&\min_{\beta, \xi, b} \frac{1}{2} \beta^T K \beta + C \sum_{i=1}^{n_s} \xi_i \\
&\text{s.t.} (K\tilde{1})^T \beta = 0 \\
&\quad y_i(\beta < \phi(X), \phi(x_i^s) > + b) \geq 1 - \xi_i, \forall i = 1, \ldots, n_s \\
&\quad \xi_i \geq 0, \forall i = 1, \ldots, n_s
\end{aligned}$$

où X représente l'ensemble des X_s et X_t.

L'optimisation de la forme duale du Lagrangien s'écrit :

$$\max_{\mu,\eta} \sum_{i=1}^{n_s} \mu_i - \frac{1}{2}(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T K^{-1} (\sum_{j=1}^{n_s} \mu_j y_j K_{j.}) - \frac{1}{2}\eta^2 \tilde{1}^T K \tilde{1} - \eta(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T \tilde{1}$$
$$\text{s.t.} \quad 0 \leq \mu_i \leq C \ \text{ and} \sum_{i=1}^{n_s} \mu_i y_i = 0 \tag{3}$$

où μ et η sont des multiplicateurs de Lagrange et $K_{.i} =< \phi(X), \phi(x_i^s) >$

Nous pouvons observer qu'à μ fixé, le Problème 3 est quadratique par rapport à η. Le terme en facteur de η^2 étant négatif, l'optimum est le maximum unique recherché et ce maximum par rapport à η peut être obtenu analytiquement : $\eta = -\frac{(\sum_{i=1}^{n_s} \mu_i y_i K_{.i})^T \tilde{1}}{\tilde{1}^T K \tilde{1}}$

Finalement, le problème devient :

$$\max_{\gamma} \gamma^T Y - \frac{1}{2}\gamma^T (K_{SS} - \frac{K_{S.}\tilde{1}\tilde{1}^T K_{S.}^T}{\tilde{1}^T K \tilde{1}})\gamma$$
$$\text{s.t.} \quad \sum_{i=1}^{n_s} \gamma_i = 0 \ \text{ and } \ min(0, Cy_i) \leq \gamma_i \leq max(0, Cy_i)$$

où $\gamma_i = \mu_i y_i$ et $K_{S.} = \sum_{i=1}^{n_s} K_{i.}$.

La contrainte $< w, \mu_s - \mu_t >= 0$, ne modifie pas la nature semi définie positive de matrice de Gram calculée dans le sous espace défini par cette contrainte et le problème d'optimisation est convexe.

3.3 Domain Adaptation avec SVM régularisé par MMD (LM)

Dans [102], les auteurs ont proposé une autre solution pour introduire la MMD dans les SVM pour le transfert d'apprentissage. De manière plus précise, ils ont utilisé

$$|| < w, \mu_s - \mu_t > ||^2$$

comme un terme de régularisation au lieu de la contrainte que nous avons proposée. La formulation du problème est la suivante :

$$\min_{w,\xi,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n_s} \xi_i + \lambda|| < w, \mu_s - \mu_t > ||^2$$
$$\text{s.t.} \ \ y_i(< w, \phi(x_i^s) > +b) \geq 1 - \xi_i \ \text{ and } \ \xi_i \geq 0$$

Cette approche réalise un compromis entre l'erreur de classification des données Source et la similitude entre les données Cible et Source.

SVMMMD peut être considéré comme un cas extrême de LM. Il arrive que LM sacrifie sa capacité de transfert pour une meilleure performance sur les données Source. Les résultats expérimentaux prouvent que SVMMMD fournit des performances supérieures à celles de LM dans la majorité des cas étudiés.

4 Domain Adaptation avec alignement dans un sous-espace de la KPCA

Pour SVMMMD, $< w, \mu_s - \mu_t >= 0$ n'est pas équivalent à $||\mu_s - \mu_t||_{\mathcal{H}} = 0$. Il n'y a aucune raison que $< w, \mu_s - \mu_t >= 0$ et que μ_s coincide avec μ_t. Dans cette section, nous proposons d'aligner explicitement Source et Cible dans un RKHS en utilisant la Kernel Principal Component Analysis (KPCA). Nous verrons que cette approche fournit en général de meilleures performances que SVMMMD.

Avant de présenter notre approche, nous rappelons brièvement les idées de base de la KPCA, qui est une extension de l'Analyse en Composantes Principales (PCA) dans un RKHS.

4.1 Fondamentaux

L'Analyse en Composantes Principales (PCA)

L'Analyse en Composantes Principales (PCA) est une approche non-supervisée qui transforme les données dans un nouveau repère dont les axes sont déterminés pour maximiser la variance de la projection obtenue. Compte tenu de la formulation du problème, les axes obtenus sont orthogonaux et préservent une proportion décroissante de la variance totale des données initiales.

Supposons que X est l'ensemble des observations, l'objectif de la PCA est de trouver un vecteur u, sur lequel nous allons projeter les données, de manière à ce que Xu garde l'information la plus importante, ici mesurée par la variance de Xu.

Après centrage de X : $X_c = X - E(X)$ (et éventuellement réduction), la PCA s'écrit :

$$\begin{aligned} \arg\max_u ||X_c u||^2 &= \arg\max_u u^T X_c^T X_c u \\ \text{s.t. } u^T u &= 1 \end{aligned}$$

La contrainte permet d'éviter que le problème soit mal posé. En résolvant ce problème par optimisation Lagrangienne, nous obtenons : $X_c^T X_c u = \lambda u$ où l'on voit que u est le vecteur propre de la matrice de variance covariance des données $(X_c^T X_c)$ associé à la valeur propre λ maximale. Les axes factoriels suivants sont donnés par les vecteurs propres de la matrice de variance covariance associés aux valeurs propres triées par ordre décroissant.

KPCA

Avec l'introduction des méthodes à noyaux, la PCA est facilement étendue à la KPCA pour prendre en compte des non linéarités potentielles entre les variables :

Dans un premier temps, les données X sont transformées par la transformation $(\Phi(.))$. Nous

obtenons la matrice de variance covariance par :

$$\begin{aligned} C_K &= \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)\Phi(x_i)^T \\ &= \frac{1}{n}\sum_{i=1}^{n}(\phi(x_i)-\frac{1}{n}\phi(x_i))(\phi(x_i)-\frac{1}{n}\phi(x_i))^T \end{aligned}$$

où $x_i \in X, \forall i = 1, \ldots, n$.

La KPCA peut s'exprimer comme :

$$\arg\max_{V} VC_KV^T \quad \text{s.t. } V^TV = I$$

où I est la matrice d'identité et V représente le vecteur recherché. V s'écrit nécessairement $V = \sum_{i=1}^{n}\alpha_i\Phi(x_i)$. La solution de la KPCA est :

$$\tilde{\lambda}V = C_KV \quad \Leftrightarrow \quad n\tilde{\lambda}\alpha = M\alpha$$

où $M = (I_n - \frac{1}{n}1_n1_n^T)K(I_n - \frac{1}{n}1_n1_n^T)$ et $K =< \phi(X), \phi(X) >$; I_n est la matrice d'identité $n \times n$; 1_n est un vecteur colonne de taille $n \times 1$ ne contenant que de 1. En considérant la contrainte $VV^T = I$, nous avons $\alpha^TM\alpha = I$ qui est aussi équivalent à $n\tilde{\lambda}\alpha^T\alpha = I$. Donc α est le vecteur propre de M et $\tilde{\lambda}$ est sa valeur propre associée qui vérifient $n\tilde{\lambda}\alpha^T\alpha = I$.

Pour exprimer une observation (z) dans la base obtenue, $V\Phi(z) = \alpha^T < \Phi(X), \Phi(z) >_{\mathcal{H}}$. Donc pour $\Phi(z)$:

$$\Phi(z) = \alpha^T < \Phi(X), \Phi(z) >_{\mathcal{H}} \overline{V} \tag{4}$$

où $\overline{V}$ représente l'ensemble des axes du repère KPCA.

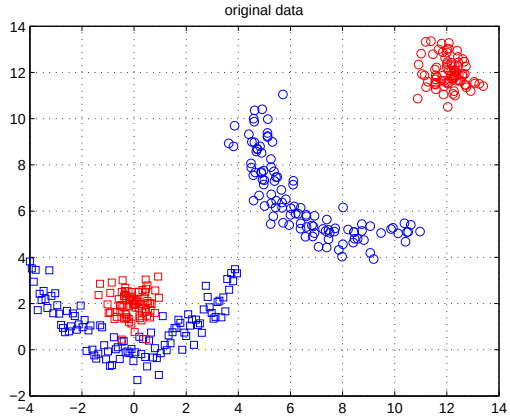
Comme la PCA, la KPCA est souvent utilisée pour la réduction de dimension en éliminant les axes factoriels associés aux valeurs propres faibles de la matrice de variance covariance, supposées correspondre au bruit.

4.2 Alignement des repères et des données par KPCA (KPCA-TL)

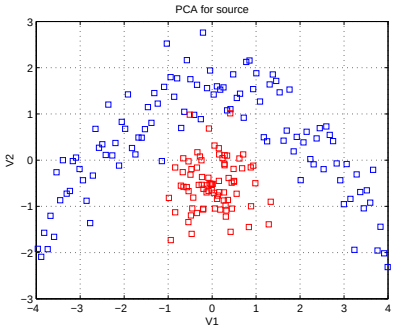
Nous proposons d'aligner Source et Cible après KPCA. L'alignement s'effectue via choix d'un repère qui rend les distributions Source et Cible les plus semblables, après transformation non linéaire. Nous présentons dans un premier temps la problématique de l'alignement après KPCA.

Problème de l'Alignement après KPCA

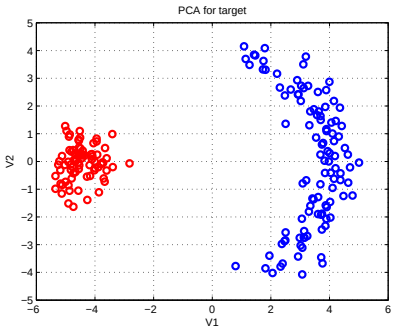
Le repère obtenu après PCA (KPCA) n'est pas unique. Les vecteurs propres de la matrice de variance covariance sont définis au signe près. Par ailleurs, selon la nature des données, il est possible que la PCA (KPCA) fasse apparaître une permutation des axes factoriels, gênante pour mesurer la similitude des observations. La figure suivante illustre ce problème :



(a) Données initiales. Les *carrés* représentent les Source et les *cercles* représentent les Cible.



(b) Source après PCA.



(c) Cible après PCA.

Figure 3 – Illustration de la permutation du premier et du deuxième vecteur propre après PCA. Dans 6.3(b) et 6.3(c), l'abscisse correspond au vecteur propre V1 ,qui est associé à la plus grande valeur propre ; l'ordonnée correspond au vecteur propre V2, qui est associé à la deuxième plus grande valeur propre.

Transfert d'apprentissage par KPCA (KPCA-TL)

Pour aligner Source et Cible, il faut éviter ces permutations et/ou inversions. KPCA-TL prend en compte ces problèmes (par une approche combinatoire) et aligne à la fois les repères et les distribution des Source et des Cible.

– Étape 1 - KPCA sur Source et Cible individuellement : KPCA-TL projette Source et Cible dans leur sous espace "optimal" respectif. La dimension du sous-espace Source (l_1) est tout d'abord choisie plus petite que celle du sous-espace Cible (l_2) [13]. L'objectif de cette étape est de permettre d'aligner le repère des Cible et le repère des Source en prenant en compte les permutations et inversions potentielles : nous sélectionnons parmi toutes les possibilités le repère des Cible qui permet d'aligner au mieux (MMD minimale) les données Source et Cible.
Pour l'étape 1, il y a $C_{l_2}^{l_1} 2^{l_1}$ combinaisons possibles à étudier pour choisir le sous-espace

13. Si $l_1 > l_2$, le raisonnement reste similaire et il faut seulement inverser la rôle des sous-espace Source et Cible.

des Source.

- Étape 2 - Nous alignons les données Cible avec les Source en considérant que leurs coordonnées (obtenues après KPCA des Cible) ont été obtenues dans le repère de la KPCA des Source.

- Étape 3 - la classification : finalement, nous utilisons une SVM sur les données Source après alignement. Sous l'hypothèse $\exists g(.) : p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$, le classifieur est censé fonctionner sur les Cible maintenant plongées dans le même repère.

4.3 Transformation Linéaire a Posteriori (KPCA-TL-LT)

Après KPCA-TL, nous avons pu observer que l'alignement des Source et Cible pouvait encore être amélioré. Pour ce faire, nous introduisons une transformation linéaire a posteriori :

$$\begin{aligned} &\arg\min_{A,b} MMD^2(A,b) \\ &\arg\min_{A,b} ||\frac{1}{n_s}\sum_{i=1}^{n_s}\phi(z_s^i) - \frac{1}{n_t}\sum_{i=1}^{n_t}\phi(Az_t^i + b)||^2 \end{aligned} \tag{5}$$

où $z_s^i \in \mathcal{Z}_s, \forall i = 1, \ldots, n_s$ ($z_t^i \in \mathcal{Z}_t, \forall i = 1, \ldots, n_t$) représente une observation Source (Cible) obtenue après KPCA-TL.

Le problème d'optimization 5 est non-convexe. Un bon choix initial de A et b est important. $A = I$ et $b = 0$ permettent d'améliorer localement la solution.

4.4 Transformation Linéaire a Priori (KPCAlin)

Les approches précédentes fournissent de très bons résultats mais requièrent une approche combinatoire pour la sélection des repères optimaux. En général, peu d'axes factoriels sont suffisants pour obtenir un bon transfert, sur l'ensemble des jeux de données simulées et réelles. Néanmoins, il est possible que, pour certains jeux de données, le nombre d'axes factoriels nécessaires devienne important et rende le temps calcul prohibitif. Dans cette partie, nous proposons une alternative basée aussi sur la KPCA. Cette approche semble efficace et robuste.

Étape 1 : Transformation Linéaire dans l'Espace d'origin

Soit M une transformation linéaire des Cible dans l'espace initial. Si nous utilisons la MMD (dans cet espace) pour aligner au mieux les données Source et Cible, nous avons :

$$\begin{aligned} &\arg\min_M ||\frac{1}{n_s}\sum_{i=1}^{n_s}\phi_s(x_s^i) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi_s(x_t^j M)||^2 \\ &\arg\min_M \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} < \phi_s(x_s^i), \phi_s(x_s^j) > + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} < \phi_s(x_t^i M), \phi_s(x_t^j M) > \\ &\quad - \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} < \phi_s(x_s^i), \phi_s(x_t^j M) > \end{aligned} \tag{6}$$

La recherche de la solution est délicate : l'optimisation est non-convexe par rapport à M. Il est difficile de déterminer de manière efficace M dans un temps raisonnable. Nous reformulons le problème d'optimisation en établissant une équivalence entre *metric learning* (trouver la meilleure transformation M) et *kernel learning* (trouver le meilleur paramètre M du noyau).

Dans ces travaux, nous utilisons le noyau gaussien. Pour $< \phi_s(x_t^i M), \phi_s(x_t^j M) >$ nous avons donc :

$$< \phi_s(x_t^i M), \phi_s(x_t^j M) >= \exp(-\frac{(x_t^i - x_t^j)MM^T(x_t^i - x_t^j)}{\sigma^2})$$

Á partir de l'équation ci-dessus, nous pouvons définir un autre noyau gaussien :

$$< \phi_s(x_t^i M), \phi_s(x_t^j M) >=< \phi_t(x_t^i), \phi_t(x_t^j) > .$$

Le problème 6 devient :

$$\arg\min_M ||\frac{1}{n_s}\sum_{i=1}^{n_s}\phi_s(x_s^i) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi_t(x_t^j)||^2 \tag{7}$$

M est maintenant un paramètre du noyau utilisé pour les données Cible. Pour résoudre le problème 7, il nous faut connaître le produit scalaire entre $\phi_s(.)$ et $\phi_t(.)$. Dans la suite, nous proposons d'exprimer explicitement ces deux transformations avec l'aide de la KPCA, pour que le problème 7 puisse être résolu.

Étape 2 : KPCA

A l'aide de la KPCA, nous obtenons :

$$\begin{aligned}
&\phi_s(X_s) = X_{ps}^T\overline{V}_s, \text{ où } X_{ps} = \alpha_s^T K_{SS} \\
&\phi_t(X_t) = X_{pt}^T\overline{V}_t, \text{ où } X_{pt} = \alpha_t^T K_{TT} \\
&\text{with } \; K_{SS}(i,j) =< \Phi_s(x_s^i), \Phi_s(x_s^j) >, \; \forall i,j = 1,\ldots,n_s \\
&\text{and } \; K_{TT}(i,j) =< \Phi_t(x_t^i), \Phi_t(x_t^j) >, \; \forall i,j = 1,\ldots,n_t
\end{aligned}$$

où $\Phi_{\{s,t\}}(.)$ représente la transformation pour les Source ou Cible, après centrage ($\Phi_s(x_s^j) = \phi_s(x_s^j) - \frac{1}{n_s}\sum_{i=1}^{n_s}\phi_s(x_s^i)$, de même pour $\Phi_t(.)$) ; α_s (α_t) est l'ensemble des vecteurs propres de K_{SS} (K_{TT}). X_{pt} est fonction de M.

Avec des paramètres de noyau (σ et M) bien choisis, si nous utilisions la PCA respectivement sur $\phi_s(X_s)$ et $\phi_t(X_t)$ (KPCA), nous devrions obtenir des données similaires (Source et Cible) dans un sous-espace en commun déterminé par $\overline{V}_{\{s,t\}}$.

Nous ne conservons que les l premières composantes principales pour les données Source ($\overline{V}_s'$) et Cible ($\overline{V}_t'$), nous appelons ce sous-espace "en commun" $\overline{V}_c$ ($\overline{V}_c = \{\vec{V}_{c1},\ldots,\vec{V}_{cl}\}$). Nous souhaitons :

$$\phi_s(X_s) \approx X_{ps}^T\overline{V}_c \;\text{ and }\; \phi_t(X_t) \approx X_{pt}^T\overline{V}_c$$

L'Eq. 7 devient :

$$\arg\min_M ||\frac{1}{n_s}\sum_{i=1}^{n_s}X_{pt}^T(i,:)\overline{V}_c - \frac{1}{n_t}\sum_{j=1}^{n_t}X_{pt}^T(j,:)\overline{V}_c||^2 \tag{8}$$

Après développement

$$\arg\min_M \frac{1}{n_s^2} 1 X_{ps}^T X_{ps} 1^T + \frac{1}{n_t^2} 1' X_{pt}^T X_{pt} 1'^T - \frac{2}{n_s n_t} 1 X_{ps}^T X_{pt} 1'^T$$

où 1 ($1'$) représente le vecteur ligne ne contenant que des 1 (de dimension $1 \times n_s(n_t)$).

Puisque X_{pt} est le seul élément qui dépend de M, l'optimisation par rapport à M devient :

$$\arg\min_M \frac{1}{n_t^2} 1' X_{pt}^T X_{pt} 1'^T - \frac{2}{n_s n_t} 1 X_{ps}^T X_{pt} 1'^T$$

L'objectif est alors de trouver le meilleur sous-espace engendré par $\phi_t(X_t)$, ce qui est équivalent à trouver la meilleure matrice X_{pt}. Nous pouvons imaginer optimiser par rapport à X_{pt}. Dans ce cas, il faut prendre en compte plusieurs contraintes sur X_{pt} :

- X_{pt} doit pouvoir s'écrire sous la forme $\alpha_t^T K_{TT}$ où α_t est la matrice qui contient les l premiers vecteurs propres de K_{TT} ; K_{TT} doit toujours être symétrique semi-definite positive.

- L'élément ij de la matrice K_{TT} est $< \phi_t(x_t^i), \phi_t(x_t^j) > = < \phi_s(x_t^i M), \phi_s(x_t^j M) >$. K_{TT} dépend explicitement de M. En conséquence, la recherche de la solution de

  $$\arg\min_M \frac{1}{n_t^2} 1' X_{pt}^T X_{pt} 1'^T - \frac{2}{n_s n_t} 1 X_{ps}^T X_{pt} 1'^T$$

  est délicate.

L'alignement des données Source et Cible avec cette aproche n'est pas simple. Nous proposons une alternative pour aligner les représentations des données Source et Cible.

Étape 3 : Alignement des Représentation dans le RKHS

Pour aligner X_{ps} et X_{pt}, nous utilisons à nouveau la MMD, évaluée dans un second RKHS (induit par $\varphi(.)$), entre X_{ps} et X_{pt}. Le problème devient :

$$\begin{aligned}
&\arg\min_M \| \frac{1}{n_s} \sum_{i=1}^{n_s} \varphi(X_{ps}^T(i,:)) - \frac{1}{n_t} \sum_{j=1}^{n_t} \varphi(X_{pt}^T(j,:)) \|^2 \\
&\arg\min_M \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} < \varphi(X_{ps}^T(i,:)), \varphi(X_{ps}^T(j,:)) > \\
&\qquad + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} < \varphi(X_{pt}^T(i,:)), \varphi(X_{pt}^T(j,:)) > \\
&\qquad - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} < \varphi(X_{ps}^T(i,:)), \varphi(X_{pt}^T(j,:)) >
\end{aligned} \tag{9}$$

En résolvant le problème 9, X_{ps} et X_{pt} devraient être rendus similaires. La similitude est atteinte par les transformations $\phi_s(.)$ et $\phi_t(.)$ qui s'appliquent sur Source et Cible, respectivement.

Étape 4 : Classification Linéaire

Finalement, une SVM à marge souple est utilisé sur les projections X_{ps}, et devrait fonctionner sur X_{pt}, sous l'hypothèse $\exists g(.) : p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$. En raison des transformations non linéaires $\phi_s(.)$ et $\phi_t(.)$, la KPCA peut rendre linéairement séparables les données qui ne le sont pas initialement. En général, un discriminateur linéaire (SVM à noyau linéaire ici) dans l'espace KPCA suffit.

Optimisation

Pour simplifier l'optimisation, nous limitons la classe de M aux matrices orthogonales : $MM^T = \gamma^2 I$ où γ est le facteur d'échelle et I est la matrice d'identité. Trouver la matrice M optimale est équivalent à chercher un paramètre de noyau optimal pour $< \phi_t(.), \phi_t(.) >$, nous pouvons alors rechercher σ_t avec lequel $< \phi_t(x_t^i), \phi_t(x_t^j) >$ peut être approché par $\exp(-\frac{(x_t^i - x_t^j)(x_t^i - x_t^j)^T}{\sigma_t^2})$. La restriction de la famille de solutions permet alors de rendre scalaire le problème d'optimisation (mais toujours non-convexe). Une simple recherche sur une grille est suffisante pour déterminer une valeur satisfaisante de σ_t. Les différents choix de σ_t sur la grille induisent différents vecteurs propres définissant les axes principaux de la KPCA (des Cible). Statistiquement, pour une grille assez fine, il est probable que le σ_t sélectionné se trouve dans le voisinage de l'optimum, tout en évitant le problème d'inversion des axes principaux.

Extension au cas *heterogeneous transfer learning*

Pour le cas *heterogeneous transfer learning*, M_{heter} n'est plus une matrice carrée, mais l'idée de limiter $M_{heter} M_{heter}^T = \gamma_{heter}^2 I$ peut toujours être utilisée. L'optimisation par rapport à M_{heter} se transforme en optimisation par rapport à γ_{heter}. Les travaux correspondants n'ont pas encore été menés.

4.5 Résultats expérimentaux

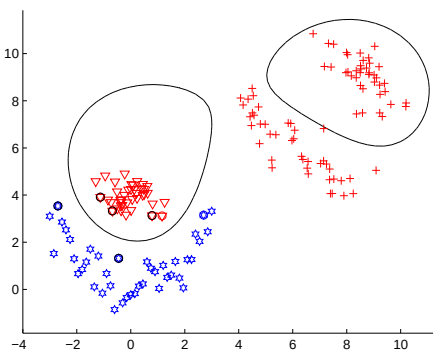
Nous avons, dans un premier temps, validé nos approches sur des données synthétiques (Figure 4). Nous constatons que le problème du transfert d'apprentissage devient de plus en plus difficile de la Figure 6.4(a) à la Figure 6.4(b) et la Figure 6.4(c).

Nous avons bien évidemment utilisé des données réelles. Nous avons utilisé les données USPS, IRIS, SEEDS pour valider nos approches de transfert d'apprentissage. La comparaison des résultats obtenus avec ceux issus d'autres approches récentes (voir Tableau 5.2, Chapitre 5, Section 5.6) démontre la validité de notre contribution.

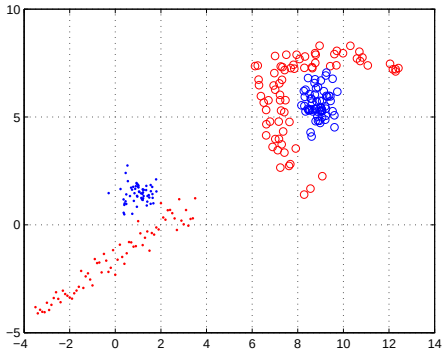
5 Conclusion et Perspectives

5.1 Conclusion

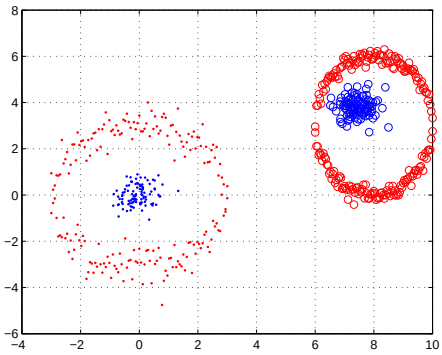
Après une introduction de la problématique et une présentation du transfert d'apprentissage, nous nous sommes intéressés au cas du *homogeneous transfer learning* en relâchant l'hypothèse



(a) Données synthétiques utilisées pour SVMMMD



(b) Données synthétiques utilisées pour les approches basées sur KPCA



(c) Autres données synthétiques utilisées pour les approches basés sur KPCA

Figure 4 – Données synthétiques. Á gauche en bas, se trouvent les données Source et à droite en haut, les données Cible. Pour les Cible, les étiquettes sont supposées inconnues. L'objectif d'utiliser les Source pour réaliser la classification des Cible. Les étiquettes des Cible ne sont utilisées que pour l'évaluation des performances.

classiquement émise. Nous avons tout d'abord présenté le *covariate shift*. Nous avons ensuite relâché l'hypothèse précédente ce qui nous a permis de présenter le *covariate shift* étendu. Nous avons considéré le cas suivant :

$$\exists\ A, b\ :\ p_S(y|x, x \in \mathcal{S}) = p_T(y|Ax + b, x \in \mathcal{T})$$

où A et b sont des paramètres. Nous recherchons donc une transformation linéaire entre Source et Target qui soit optimale au sens où les distributions marginales ($p_S(x, x \in \mathcal{S})$ et $p_T(Ax+b, x \in \mathcal{T})$) soient alignées (semblables). La solution est recherché par la méthode du maximum de vraisemblance, après estimation de densité. Hélas, l'optimisation est non-convexe et difficile.

Ensuite, nous nous sommes intéressés à un cas beaucoup plus général :

$$\exists g(.)\ :\ p_S(y|x, x \in \mathcal{S}) = p_T(y|g(x), x \in \mathcal{T})$$

où $g(.)$ est une fonction régulière, généralement non-linéaire. Sous cette hypothèse, nous avons

proposé quelques approches basées sur les méthodes à noyau.

Nous avons reformulé le problème primal des SVM avec une contrainte additionnelle sur la similitude des distributions. Nous imposons que la solution (le discriminateur) se trouve dans un sous espace dans lequel la Maximum Mean Discrepancy (MMD), aisément calculable à l'aide d'un noyau, projetée soir nulle, ce qui permet d'espérer que les données Source et Cible deviennent semblables. Sous l'hypothèse précédente, le classifieur des Source s'appliquera aux données Cible.

La contrainte satisfaite n'implique pas que la MMD soit nulle (ce qui correspond à l'égalité des distributions). Il existe des cas où notre méthode échoue. Pour résoudre ce problème, nous avons proposé trois approches reposant sur l'Analyse en Composantes Principales à Noyau (KPCA). L'objectif est de rendre similaires les lois marginales des observations Source et Cible dans un espace de Hilbert à noyau reproduisant.

- KPCA-TL aligne les repères de les distributions de Source et de Cible dans un sous-espace de Hilbert à noyau reproduisant déterminé par KPCA. En évaluant la similitude (à l'aide de la MMD) des distributions obtenues après KPCA des Source et des Cible, nous sélectionnons un espace de représentation conjoint "optimal" qui conduit à des meilleures performances que SVMMMD.

- KPCA-TL-LT est un post traitement de KPCA-TL. KPCA-TL-LT applique une transformation linéaire sur les données après KPCA-TL. Cette transformation linéaire permet encore d'améliorer la ressemblance entre données Source et Cible après sélection du meilleur sous espace conjoint.

- Enfin, KPCAlin a pour objectif d'améliorer l'efficacité de KPCA-TL et KPCA-TL-LT. L'idée initiale de KPCAlin est de chercher une transformation linéaire a priori dans l'espace initial qui permette de faire coïncider au mieux Source et Cible dans l'espace obtenu après KPCA. Nous avons montré que la KPCA des données Cible après cette transformation est équivalente la réalisation d'une KPCA des Cible avec un noyau différent de celui des Source. En sélectionnant correctement le paramètre du noyau utilisé sur les données Cible, nous obtenons un sous-espace KPCA des Cible qui s'aligne bien avec le sous-espace KPCA des Source.

Les expériences réalisées sur des données synthétiques et réelles ont montré l'efficacité et la robustesse de nos approches, dont les résultats ont été comparés à ceux des méthodes récentes issues de la littérature.

5.2 Perspectives

Dans le contexte du *homogeneous transfer learning*, nous nous sommes intéressés au cas de la classification binaire. Les résultats sont très encourageants et nous permettent d'envisager l'extension de nos travaux au cas multi classes.

Dans un second temps, il serait opportun d'étendre nos travaux réalisés dans le cadre du *homogeneous transfer learning* au cas du *heterogenous transfer learning*, pour lequel les données Source et Cible ne partagent pas le même espace de représentation. Nous avons identifié quelques

pistes potentielles dans nos travaux.

Dans nos travaux, nous avons supposé que les données Cible n'étaient pas étiquetées. Il serait intéressant d'étudier le gain potentiel lorsque nous disposons d'un nombre réduit d'observations Cible étiquetées.

Bibliography

[1] Mark Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25 :821–837, 1964.

[2] Rahaf Aljundi, Rémi Emonet, Damien Muselet, and Marc Sebban. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 56–63, 2015.

[3] Pablo Arias, Gregory Randall, and Guillermo Sapiro. Connecting the out-of-sample and pre-image problems in kernel methods. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[4] Yusuf Aytar and Andrew Zisserman. Tabula rasa : Model transfer for object category detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2252–2259. IEEE, 2011.

[5] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 769–776, 2013.

[6] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6) :1373–1396, 2003.

[7] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization : A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov) :2399–2434, 2006.

[8] Kristin P Bennett and Olvi L Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, 1(1) :23–34, 1992.

[9] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[10] 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*, pages 81–88. ACM, 2007.

[11] John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders : Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

[12] 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*, pages 120–128. Association for Computational Linguistics, 2006.

[13] Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

[14] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14) :e49–e57, 2006.

[15] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced lectures on machine learning*, pages 169–207. Springer, 2004.

[16] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[17] Lorenzo Bruzzone, Mingmin Chi, and Mattia Marconcini. A novel transductive svm for semisupervised classification of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11) :3363–3373, 2006.

[18] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems : A dasvm classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5) :770–787, 2010.

[19] Bin Cao, Xiaochuan Ni, Jian-Tao Sun, Gang Wang, and Qiang Yang. Distance metric learning under covariate shift. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1204, 2011.

[20] Rich Caruana. Multitask learning. *Machine learning*, 28(1) :41–75, 1997.

[21] Rita Chattopadhyay, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Joint transfer and batch-mode active learning. In *International Conference on Machine Learning*, pages 253–261, 2013.

[22] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4) :18, 2012.

[23] Depin Chen, Yan Xiong, Jun Yan, Gui-Rong Xue, Gang Wang, and Zheng Chen. Knowledge transfer for cross domain learning to rank. *Information Retrieval*, 13(3) :236–253, 2010.

[24] Xiaoyi Chen and Régis Lengellé. Domain adaptation transfer learning by svm subject to a maximum-mean-discrepancy-like constraint. In *ICPRAM*, pages 89–95, 2017.

[25] Fan RK Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[26] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In *International Conference on Algorithmic Learning Theory*, pages 38–53. Springer, 2008.

[27] Gabriela Csurka. Domain adaptation for visual applications : A comprehensive survey. *arXiv preprint arXiv :1702.05374*, 2017.

[28] Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu. Translated learning : Transfer learning across different feature spaces. In *Advances in neural information processing systems*, pages 353–360, 2009.

[29] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 210–219. ACM, 2007.

[30] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007.

[31] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. In *Proceedings of the 25th international conference on Machine learning*, pages 200–207. ACM, 2008.

[32] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv :0907.1815*, 2009.

[33] Jesse Davis and Pedro Domingos. Deep transfer via second-order markov logic. In *Proceedings of the 26th annual international conference on machine learning*, pages 217–224. ACM, 2009.

[34] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[35] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 668–675, 2013.

[36] 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, 2012.

[37] Lixin Duan, Ivor W Tsang, Dong Xu, and Stephen J Maybank. Domain transfer svm for video concept detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1375–1381. IEEE, 2009.

[38] Lixin Duan, Dong Xu, and Shih-Fu Chang. Exploiting web images for event recognition in consumer videos : A multiple source domain adaptation approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1338–1345. IEEE, 2012.

[39] Lixin Duan, Dong Xu, and Ivor Tsang. Learning with augmented features for heterogeneous domain adaptation. *arXiv preprint arXiv :1206.4660*, 2012.

[40] Miroslav Dudík, Steven J Phillips, and Robert E Schapire. Correcting sample selection bias in maximum entropy density estimation. In *Advances in neural information processing systems*, pages 323–330, 2006.

[41] Richard M Dudley. A course on empirical processes. In *Ecole d'été de Probabilités de Saint-Flour XII-1982*, pages 1–142. Springer, 1984.

[42] Eric Eaton, Terran Lane, et al. Modeling transfer relationships between learning tasks for improved inductive transfer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 317–332. Springer, 2008.

[43] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[44] Kyle D Feuz and Diane J Cook. Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (fsr). *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(1) :3, 2015.

[45] R. Fortet and E. Mourier. Convergence de la répartition empirique vers la réparation théorique. *Ann. Scient. École Norm. Sup.*, pages 266–285, 1953.

[46] Shereen Fouad, Peter Tino, Somak Raychaudhury, and Petra Schneider. Incorporating privileged information through metric learning. *IEEE transactions on neural networks and learning systems*, 24(7) :1086–1098, 2013.

[47] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 2013.

[48] 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*, pages 283–291. ACM, 2008.

[49] Jochen Garcke and Thomas Vanck. Importance weighted inductive transfer learning for regression. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 466–481. Springer, 2014.

[50] Liang Ge, Jing Gao, Hung Ngo, Kang Li, and Aidong Zhang. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining : The ASA Data Science Journal*, 7(4) :254–271, 2014.

[51] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks : Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 222–230, 2013.

[52] Boqing Gong, Kristen Grauman, and Fei Sha. Reshaping visual datasets for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1286–1294, 2013.

[53] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[54] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition : An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

[55] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13 :723–773, March 2012.

[56] Lin Gui, Ruifeng Xu, Qin Lu, Jiachen Du, and Yu Zhou. Negative transfer detection in transductive transfer learning. *International Journal of Machine Learning and Cybernetics*, pages 1–13, 2017.

[57] Yuhong Guo. Robust transfer principal component analysis with rank constraints. In *Advances in Neural Information Processing Systems*, pages 1151–1159, 2013.

[58] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[59] Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv :1301.3224*, 2013.

[60] Paul Honeine and Cédric Richard. Solving the pre-image problem in kernel machines : A direct method. In *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on*, pages 1–6. IEEE, 2009.

[61] Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with label and structural consistency. *IEEE Transactions on Image Processing*, 25(12) :5552–5562, 2016.

[62] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2007.

[63] Wei Jiang, Eric Zavesky, Shih-Fu Chang, and Alex Loui. Cross-domain learning methods for high-level visual concept classification. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 161–164. IEEE, 2008.

[64] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.

[65] George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1) :82–95, 1971.

[66] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get : Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011.

[67] Sanjeev R Kulkarni and Gilbert Harman. Statistical learning theory : a tutorial. *Wiley Interdisciplinary Reviews : Computational Statistics*, 3(6) :543–556, 2011.

[68] Sanjeev R Kulkarni, Gábor Lugosi, and Santosh S. Venkatesh. Learning pattern classification-a survey. *IEEE Transactions on Information Theory*, 44(6) :2178–2206, 1998.

[69] JT-Y Kwok and IW-H Tsang. The pre-image problem in kernel methods. *IEEE transactions on neural networks*, 15(6) :1517–1525, 2004.

[70] Stephane Lafon and Ann B Lee. Diffusion maps and coarse-graining : A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE transactions on pattern analysis and machine intelligence*, 28(9) :1393–1403, 2006.

[71] 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*, page 65. ACM, 2004.

[72] Wen Li, Lixin Duan, Dong Xu, and Ivor W Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 36(6) :1134–1148, 2014.

[73] Xiao Ling, Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Spectral domain-transfer learning. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 488–496. ACM, 2008.

[74] Mingsheng Long, Jianmin Wang, Guiguang Ding, Sinno Jialin Pan, and S Yu Philip. Adaptation regularization : A general framework for transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 26(5) :1076–1089, 2014.

[75] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

[76] Mingsheng Long, Jianmin Wang, Jiaguang Sun, and S Yu Philip. Domain invariant transfer kernel learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6) :1519–1532, 2015.

[77] Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang. Transfer learning using computational intelligence : a survey. *Knowledge-Based Systems*, 80 :14–23, 2015.

[78] Gábor Lugosi. Pattern classification and learning theory. In *Principles of nonparametric learning*, pages 1–56. Springer, 2002.

[79] Andy J Ma, Pong C Yuen, and Jiawei Li. Domain transfer support vector ranking for person re-identification without target camera label information. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3567–3574, 2013.

[80] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov) :2579–2605, 2008.

[81] Anna Margolis. A literature review of domain adaptation with unlabeled data. *Tec. Report*, pages 1–42, 2011.

[82] Shahar Mendelson. A few notes on statistical learning theory. *Lecture notes in computer science*, 2600 :1–40, 2003.

[83] Charles A Micchelli. Algebraic aspects of interpolation. In *Proceedings of Symposia in Applied Mathematics*, volume 36, pages 81–102, 1986.

[84] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *The Journal of Machine Learning Research*, 7 :2651–2667, 2006.

[85] Lilyana Mihalkova, Tuyen Huynh, and Raymond J Mooney. Mapping and revising markov logic networks for transfer learning. In *AAAI*, volume 7, pages 608–614, 2007.

[86] Sebastian Mika, Bernhard Schölkopf, Alex J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.

[87] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 10–18, 2013.

[88] Jaechang Nam, Wei Fu, Sunghun Kim, Tim Menzies, and Lin Tan. Heterogeneous defect prediction. *IEEE Transactions on Software Engineering*, 2017.

[89] Balas K Natarajan. *Machine learning : a theoretical approach*. Morgan Kaufmann, 2014.

[90] Michael K Ng, Qingyao Wu, and Yunming Ye. Co-transfer learning via joint transition probability graph based method. In *Proceedings of the 1st international workshop on cross domain knowledge discovery in web and social network mining*, pages 1–9. ACM, 2012.

[91] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 692–699, 2013.

[92] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[93] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2) :199–210, 2011.

[94] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10) :1345–1359, 2010.

[95] Xiaoxi Pang. The pagerank citation ranking : Bring order to the web. 2010.

[96] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation : A survey of recent advances. *IEEE signal processing magazine*, 32(3) :53–69, 2015.

[97] Vern I Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152. Cambridge University Press, 2016.

[98] P J Phillips, Jingjing Zheng, and Rama Chellappa. Sparse embedding-based domain adaptation for object recognition. In *The 1st International Workshop on Visual Domain Adaptation and Dataset Bias*, 2013.

[99] Guo-Jun Qi, Charu Aggarwal, and Thomas Huang. Towards semantic knowledge propagation from text corpus to web images. In *Proceedings of the 20th international conference on World wide web*, pages 297–306. ACM, 2011.

[100] Guo-Jun Qi, Charu Aggarwal, and Thomas Huang. Transfer learning of distance metrics by cross-domain metric sampling across heterogeneous spaces. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 528–539. SIAM, 2012.

[101] Qiang Qiu, Vishal M Patel, Pavan Turaga, and Rama Chellappa. Domain adaptive dictionary learning. In *European Conference on Computer Vision*, pages 631–645. Springer, 2012.

[102] Brian Quanz and Jun Huan. Large margin transductive transfer learning. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1327–1336. ACM, 2009.

[103] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[104] 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*, pages 759–766. ACM, 2007.

[105] Jiangtao Ren, Zhou Liang, and Shaofeng Hu. Multiple kernel learning improved by mmd. *Advanced Data Mining and Applications*, pages 63–74, 2010.

[106] Jiangtao Ren, Xiaoxiao Shi, Wei Fan, and Philip S Yu. Type independent correction of sample selection bias via structural discovery and re-balancing. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 565–576. SIAM, 2008.

[107] F Riez and B Nagy. Functional analysis ungar. *New York*, 1955.

[108] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, volume 898, 2005.

[109] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500) :2323–2326, 2000.

[110] Bernhard Scholkopf, Sebastian Mika, Chris JC Burges, Philipp Knirsch, K-R Muller, Gunnar Ratsch, and Alexander J Smola. Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5) :1000–1017, 1999.

[111] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

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

[113] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[114] Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning gaussian process kernels via hierarchical bayes. In *Advances in Neural Information Processing Systems*, pages 1209–1216, 2005.

[115] Chun-Wei Seah, Yew-Soon Ong, and Ivor W Tsang. Combating negative transfer from predictive distribution differences. *IEEE transactions on cybernetics*, 43(4) :1153–1165, 2013.

[116] D. Sejdinovic and A. Gretton. Foundations of reproducing kernel hilbert spaces, advanced topics in machine learning. *www.gatsby.ucl.ac.uk/ dino/teaching*, 2014.

[117] Dino Sejdinovic and Arthur Gretton. What is an rkhs ? 2012.

[118] Robert J Serfling. *Approximation theorems of mathematical statistics*, volume 162. John Wiley & Sons, 2009.

[119] Ling Shao, Fan Zhu, and Xuelong Li. Transfer learning for visual categorization : A survey. *IEEE transactions on neural networks and learning systems*, 26(5) :1019–1034, 2015.

[120] Ming Shao, Dmitry Kit, and Yun Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109(1-2) :74–93, 2014.

[121] Sumit Shekhar, Vishal M Patel, Hien V Nguyen, and Rama Chellappa. Generalized domain-adaptive dictionaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–368, 2013.

[122] Xiaoxiao Shi, Qi Liu, Wei Fan, S Yu Philip, and Ruixin Zhu. Transfer learning on heterogenous feature spaces via spectral transformation. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 1049–1054. IEEE, 2010.

[123] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. *arXiv preprint arXiv :1206.6438*, 2012.

[124] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2) :227–244, 2000.

[125] Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7) :929–942, 2010.

[126] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

[127] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31. Springer, 2007.

[128] Alexander Smola. Maximum mean discrepancy. In *13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006 : Proceedings*, 2006.

[129] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11(Apr) :1517–1561, 2010.

[130] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *The Journal of Machine Learning Research*, 2 :67–93, 2002.

[131] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert MÃžller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May) :985–1005, 2007.

[132] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008.

[133] Baochen Sun and Kate Saenko. Subspace distribution alignment for unsupervised domain adaptation. In *BMVC*, pages 24–1, 2015.

[134] Ben Tan, Etheng Zhong, Michael K Ng, and Qiang Yang. Mixed-transfer : transfer learning over mixed graphs. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 208–216. SIAM, 2014.

[135] Qi Tan, Huifang Deng, and Pei Yang. Kernel mean matching with a large margin. In *ADMA*, pages 223–234. Springer, 2012.

[136] XINLU TAN. Notes on reproducing kernel hilbert space.

[137] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500) :2319–2323, 2000.

[138] Andreui Nikolaevich Tikhonov, Vasiliui Yakovlevich Arsenin, and Fritz John. *Solutions of ill-posed problems*, volume 14. Winston Washington, DC, 1977.

[139] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers : Learning categories from few examples with multi model knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3081–3088. IEEE, 2010.

[140] Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17 :138–155, 2009.

[141] Wenting Tu and Shiliang Sun. Transferable discriminative dimensionality reduction. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 865–868. IEEE, 2011.

[142] Selen Uguroglu and Jaime Carbonell. Feature selection for transfer learning. *Machine learning and knowledge discovery in databases*, pages 430–442, 2011.

[143] Annegreet van Opbroek, M Ikram, Meike Vernooij, and Marleen de Bruijne. Supervised image segmentation across scanner protocols : A transfer learning approach. *Machine Learning in Medical Imaging*, pages 160–167, 2012.

[144] Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[145] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[146] Roger Velásquez, L Enrique Sucar, and Eduardo F Morales. Transfer learning for bayesian networks. *Advances in Artificial Intelligence-IBERAMIA*, pages 14–17, 2008.

[147] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4) :395–416, 2007.

[148] Chang Wang and Sridhar Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1541, 2011.

[149] Xiaogang Wang, Meng Wang, and Wei Li. Scene-specific pedestrian detection for static video surveillance. *IEEE transactions on pattern analysis and machine intelligence*, 36(2) :361–374, 2014.

[150] Xuezhi Wang, Tzu-Kuo Huang, and Jeff Schneider. Active transfer learning under model shift. In *International Conference on Machine Learning*, pages 1305–1313, 2014.

[151] Zheng Wang, Yangqiu Song, and Changshui Zhang. Transferred dimensionality reduction. *Machine learning and knowledge discovery in databases*, pages 550–565, 2008.

[152] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May, 1998.

[153] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis : Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[154] Qingyao Wu, Michael K Ng, and Yunming Ye. Cotransfer learning using coupled markov chains with restart. *IEEE Intelligent Systems*, 29(4) :26–33, 2014.

[155] Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria. Feature ensemble plus sample selection : domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3) :10–18, 2013.

[156] Ge Xie, Yu Sun, Minlong Lin, and Ke Tang. A selective transfer learning method for concept drift adaptation. In *International Symposium on Neural Networks*, pages 353–361. Springer, 2017.

[157] Dikan Xing, Wenyuan Dai, Gui-Rong Xue, and Yong Yu. Bridged refinement for transfer learning. In *PKDD*, pages 324–335. Springer, 2007.

[158] Yong Xu, Xiaozhao Fang, Jian Wu, Xuelong Li, and David Zhang. Discriminative transfer subspace learning via low-rank and sparse representation. *IEEE Transactions on Image Processing*, 25(2) :850–863, 2016.

[159] Yonghui Xu, Sinno Jialin Pan, Hui Xiong, Qingyao Wu, Ronghua Luo, Huaqing Min, and Hengjie Song. A unified framework for metric transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(6) :1158–1171, 2017.

[160] Zhijie Xu and Shiliang Sun. Multi-source transfer learning with multi-view adaboost. In *International Conference on Neural Information Processing*, pages 332–339. Springer, 2012.

[161] Shuicheng Yan, Dong Xu, Benyu Zhang, and Hong-Jiang Zhang. Graph embedding : A general framework for dimensionality reduction. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 830–837. Ieee, 2005.

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

[163] Liu Yang, Liping Jing, Jian Yu, and Michael K Ng. Learning transferred weights from co-occurrence data for heterogeneous transfer learning. *IEEE transactions on neural networks and learning systems*, 27(11) :2187–2200, 2016.

[164] Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. Heterogeneous transfer learning for image clustering via the social web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP : Volume 1-Volume 1*, pages 1–9. Association for Computational Linguistics, 2009.

[165] Shizhun Yang, Ming Lin, Chenping Hou, Changshui Zhang, and Yi Wu. A general framework for transfer sparse subspace learning. *Neural Computing and Applications*, 21(7) :1801–1817, 2012.

[166] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1855–1862. IEEE, 2010.

[167] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114. ACM, 2004.

[168] Xiaohua Zhai, Yuxin Peng, and Jianguo Xiao. Heterogeneous metric learning with joint graph regularization for cross-media retrieval. In *AAAI*, 2013.

[169] Kai Zhang, Vincent Zheng, Qiaojun Wang, James Kwok, Qiang Yang, and Ivan Marsic. Covariate shift in hilbert space : A solution via sorrogate kernels. In *International Conference on Machine Learning*, pages 388–395, 2013.

[170] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827, 2013.

[171] Peng Zhang, Xingquan Zhu, and Li Guo. Mining data streams with labeled and unlabeled training examples. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 627–636. IEEE, 2009.

[172] Yu Zhang and Dit-Yan Yeung. Transfer metric learning by learning task relationships. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1199–1208. ACM, 2010.

[173] Zhihao Zhang and Jie Zhou. Multi-task clustering via domain adaptation. *Pattern Recognition*, 45(1) :465–473, 2012.

[174] Lili Zhao, Sinno Jialin Pan, Evan Wei Xiang, Erheng Zhong, Zhongqi Lu, and Qiang Yang. Active transfer learning for cross-system recommendation. In *AAAI*, 2013.

[175] Wei-Shi Zheng and Jian-huang Lai. Regularized locality preserving learning of pre-image problem in kernel principal component analysis. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 456–459. IEEE, 2006.

[176] Wei-Shi Zheng, JianHuang Lai, and Pong C Yuen. Penalized preimage learning in kernel principal component analysis. *IEEE Transactions on Neural Networks*, 21(4) :551–570, 2010.

[177] Erheng Zhong, Wei Fan, Jing Peng, Kun Zhang, Jiangtao Ren, Deepak Turaga, and Olivier Verscheure. Cross domain distribution adaptation via kernel mapping. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1027–1036. ACM, 2009.

[178] Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 547–562. Springer, 2010.

[179] Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *AAAI*, 2011.

Xiaoyi CHEN

Doctorat : Optimisation et Sûreté des Systèmes

Année 2018

Transfert d'apprentissage et méthodes à noyau

Le transfert d'apprentissage regroupe les méthodes permettant de transférer l'apprentissage réalisé sur des données (appelées Source) à des données nouvelles, différentes, mais liées aux données Source. Ces travaux sont une contribution au transfert d'apprentissage homogène (les domaines de représentation des Source et Cible sont identiques) et transductif (la tâche à effectuer sur les données Cible est identique à celle sur les données Source), lorsque nous ne disposons pas d'étiquettes des données Cible. Dans ces travaux, nous relâchons la contrainte d'égalité des lois des étiquettes conditionnellement aux observations, souvent considérée dans la littérature. Notre approche permet de traiter des cas de plus en plus généraux. Elle repose sur la recherche de transformations permettant de rendre similaires les données Source et Cible. Dans un premier temps, nous recherchons cette transformation par Maximum de Vraisemblance. Ensuite, nous adaptons les Machines à Vecteur de Support en intégrant une contrainte additionnelle sur la similitude des données Source et Cible. Cette similitude est mesurée par la Maximum Mean Discrepancy. Enfin, nous proposons l'utilisation de l'Analyse en Composantes Principales à noyau pour rechercher un sous espace, obtenu à partir d'une transformation non linéaire des données Source et Cible, dans lequel les lois des observations sont les plus semblables possibles. Les résultats expérimentaux montrent l'efficacité de nos approches.

Mots clés : apprentissage automatique - machines à vecteurs de support - noyaux (analyse fonctionnelle) – analyse multivariée.

Transfer Learning with Kernel Methods

Transfer Learning aims to take advantage of source data to help the learning task of related but different target data. This thesis contributes to homogeneous transductive transfer learning where no labeled target data is available. In this thesis, we relax the constraint on conditional probability of labels required by covariate shift to be more and more general, based on which the alignment of marginal probabilities of source and target observations renders source and target similar. Thus, firstly, a maximum likelihood based approach is proposed. Secondly, SVM is adapted to transfer learning with an extra MMD-like constraint where Maximum Mean Discrepancy (MMD) measures this similarity. Thirdly, KPCA is used to align data in a RKHS on minimizing MMD. We further develop the KPCA based approach so that a linear transformation in the input space is enough for a good and robust alignment in the RKHS. Experimentally, our proposed approaches are very promising.

Keywords: machine learning - support vector machines - kernel functions – multivariate analysis.

Thèse réalisée en partenariat entre :

Ecole Doctorale "Sciences pour l'Ingénieur"