CS678A COURSE PROJECT

Domain Invariant Transfer Kernel Learning

Piyush Singla - 12482
Shreesh Laddha - 12679

Abstract

Domain transfer learning generalizes a learning model across training data and testing data with different distributions. A general principle to face this issue is by reducing the distribution difference between the training and the testing data such that the generalization error can be bounded. The technique proposed in our project is through Transfer Kernel Learning, to learn a domain invariant kernel. Specifically, we design a family of spectral kernels by extrapolating target eigensystem on source data points with Mercer's Theorem and Nyström approximation.The spectral kernel minimizing the approximation error to the original source kernel is selected to construct the domain invariant kernel. Several experiments have been performed over text, image and brain data to identify the effectiveness of Transfer Kernel Learning over normal SVM algorithm.

1 Introduction

Most of the machine algorithms out there, are build upon the basic principle that the training and testing data are from same distributions. However, in today's era, generation of data from many heterogeneous sources, in domains of texts, images and videos has created a compelling requirement to build models that generalize well across all these different distributions. So for example, if we wish to build an object detection model which has been trained on datasets from one particular domain(say web cams) and tested on images from some other domain(say dslr's), then the results would definitely not be up to the mark. Similarly a survey conducted in one region, might not necessarily be applicable in some other region because of difference in population.

Added to this is the fact that when the distributions are different, most statistical models have to be remodeled from scratch using the newly collected training data. However, in many real-world applications, it is quite expensive(or even impossible) to recollect the needed training data and update the models. Hence in such cases, devising a method to successfully transfer the knowledge between the two different domains is highly desirable. The domain invariant task that we employ in this paper involves using two distinct types of datasets - a source and a target domain. Source domain basically includes the labeled training data using which a successful classifier can be constructed. The target domain is the testing/unlabeled data from a different yet related distribution. Our goal is to minimize this distribution mismatch, such that our classifier is adaptive across domains.

A lot of prior work has been done is this area. A general approach to tackle this problem involves minimization of distribution difference between the source and target domain. Such distribution discrepancies can be formalized using the KL divergence, Bregman divergence and Maximum Mean Discrepancy (MMD). However, the applicability of KL and Bregman divergence is hindered by the need of a density estimation procedure which is computationally inefficient. Similarly, joint minimization of MMD along with the empirical loss is a Semi Definite Programming(SDP) problem, which is of the order $O(n^{6.5})$. The novelty in our approach, is that the distribution mismatch is corrected using Nyström approximation error, which is both computationally efficient and gives good results.

2 Preliminaries

2.1 Mercer Theorem

Let $k(\mathbf{z}, \mathbf{x})$ be a continuous symmetric non - negative function which is positive semi - definite and square integrable w.r.t. distribution $p(x)$, then,

$$k(\mathbf{z}, \mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{z}) \phi_i(\mathbf{x})$$

The eigenvalues λ_i's and the orthonormal eigenfunctions ϕ_i's are the solutions of the following integral equation,

$$\int k(\mathbf{z}, \mathbf{x}) \phi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_i \phi_i(\mathbf{z})$$

2.2 Spectral Kernel

If a positive semi-definite kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ has the eigensystem $\{\gamma_i, \phi_i\}, \gamma_1 \geq \ldots \geq \gamma_n \geq 0$, then the family of matrices,

$$K_\lambda = \sum_{i=1}^{\infty} \lambda_i \phi_i \phi_i^T, \lambda_1 \geq \cdots \geq \lambda_n \geq 0$$

will produce PSD kernels with K_λ as kernel matrices.

3 Transfer Kernel Learning

3.1 Problem Formulation

Definition 1 : A domain D is composed of a d-dimensional feature space F and a marginal probability distribution $P(\mathbf{x})$, i.e, $D = \{F, P(\mathbf{x})\}, \mathbf{x} \in F$

Definition 2 : Given domain D, a task T is composed of a c - cardinality label set Y and a classifier $f(\mathbf{x})$, i.e, $T = \{Y, f(\mathbf{x})\}$, where $y \in Y$ and $f(\mathbf{x}) = P(y \mid \mathbf{x})$ can be interpreted as the conditional probability distribution.

Problem Statement (Transfer Kernel Learning)
Given a labeled source domain $Z = \{(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_m, y_m)\}$ and an unlabeled target domain $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $F_Z = F_X, Y_Z = Y_X, P(\mathbf{z}) \neq P(\mathbf{x})$ and $P(y \mid \mathbf{z}) = P(y \mid \mathbf{x})$, learn a domain invariant kernel $k(\mathbf{z}, \mathbf{x}) =< \phi(\mathbf{z}), \phi(\mathbf{x}) >$ such that $P(\phi(\mathbf{z})) \simeq P(\phi(\mathbf{x}))$.

3.2 Learning Approach

As discussed in the problem formulation above, we wish to come up with a mapped feature space where the distributions are similar i.e $P(\phi(\mathbf{x})) \simeq P(\phi(\mathbf{z}))$. To have the distributions to be similar, it suffices to have the kernel matrices of the source and target domain to be similar, i.e $\mathbf{K}_X \simeq \mathbf{K}_Z$. To observe this, note that a kernel matrix K, can be written as $K = (KK^{-1/2})(K^{-1/2}K)$, and it's corresponding empirical kernel map can be written as $\phi_{emp} = K^{1/2}$. Therefore, if two kernel matrices are the same, i.e., $\mathbf{K}_X \simeq \mathbf{K}_Z$, then their corresponding empirical feature maps would also be the same, i.e, $\phi(\mathbf{x}) \simeq \phi(\mathbf{z})$ and as a result the empirical distributions of the data in the mapped feature space would be the same, i.e, $P(\phi(\mathbf{x})) \simeq P(\phi(\mathbf{z}))$.

But there is no certainty that the two kernel matrices would have the same dimensions, making them unable to be compared. Hence we use **Nyström approximation** to come up with an extrapolated source kernel $\overline{\mathbf{K}}_Z$, to compare it to the original source kernel, using the eigensystem of the target kernel $\mathbf{K}_X$

3.3 Nyström Kernel Approximation and Extrapolation

We start off with approximating the integral equation mention under 2.1, by its emperical estimate:

$$\sum_{j=1}^{n} \frac{k(\mathbf{z}, \mathbf{x}_j)\phi_i(\mathbf{x}_j)}{n} \simeq \lambda_i \phi_i(\mathbf{z}) \tag{1}$$

As we vary $\mathbf{z}$ over X in the above equation, we come up with its matrix form:

$$\mathbf{K}_X \mathbf{\Phi}_X^{'} = \mathbf{\Phi}_X^{'} \mathbf{\Lambda}_X^{'} \tag{2}$$

Also since $\mathbf{K}_X$ is a kernel matrix, using the standard eigendecomposition we can write it in the form :

$$\mathbf{K}_X \mathbf{\Phi}_X = \mathbf{\Phi}_X \mathbf{\Lambda}_X \tag{3}$$

By comparing the two equations we can easily find out the values of the eigenfunction $\phi_i(\mathbf{x}_j)$'s and λ_i's. Formally, it would be equal to :

$$\frac{\phi_i(\mathbf{x}_j)}{\sqrt{n}} = [\mathbf{\Phi}_X]_{ij}, n\lambda_i = [\mathbf{\Lambda}_X]_{ii} \tag{4}$$

Now we can use equation (1) to extrapolate the eigenfunction ϕ_i at any arbitrary point $\mathbf{z}$ by $\phi_i(\mathbf{z}) = \sum_{j=1}^{n} \frac{k(\mathbf{z},\mathbf{x}_j)\phi_i(\mathbf{x}_j)}{n}$. So, evaluating eigenfunction ϕ_i on a new dataset $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ with distribution $p(\mathbf{z})$ leads to the approximation :

$$\overline{\mathbf{\Phi}}_Z \simeq \mathbf{K}_{ZX} \mathbf{\Phi}_X \mathbf{\Lambda}_X^{-1} \tag{5}$$

where $\mathbf{K}_{ZX} \in \mathbb{R}^{m \times n}$, is the cross domain similarity matrix between Z and X evaluated using our chosen kernel k. Further using the Mercer's theorem explained above,

$$\mathbf{K}_Z \simeq \mathbf{\Phi}_Z \mathbf{\Lambda}_X \mathbf{\Phi}_Z^T = \mathbf{K}_{ZX} \mathbf{K}_X^{-1} \mathbf{K}_{XZ} \tag{6}$$

This entire procedure is the method of Nyström Kernel Approximation. Note that this approximation is valid only if the probability distributions of the source and target domain are similar, i.e, $p(\mathbf{z}) \simeq p(\mathbf{x})$. If the distributions are substantially different, then the Nyström approximation error would be very large. So, in a sense the Nyström approximation error embodies the distribution difference across domains. Hence minimizing this error is essentially equivalent to reducing the distribution divergence between our source and target domains. Finally to reduce the error, we relax the the eigenvalues and use the idea of Spectral Kernel Design to create a family of kernels using the eigensystem of the extrapolated source kernel $\overline{\mathbf{\Phi}}_\mathbf{Z}$. This eigensystem, built using the target kernel $\mathbf{K}_X$, preserves its key structure and simultaneously allows itself to be reshaped to minimize the error. As a result, our extrapolated source kernel becomes :

$$\overline{\mathbf{K}}_Z = \overline{\mathbf{\Phi}}_Z \mathbf{\Lambda} \overline{\mathbf{\Phi}}_Z^T \tag{7}$$

3.4 Nyström Approximation Error Minimization

Now we reduce the distribution difference between the source and target domains by minimizing the Nyström approximation error between the extrapolated source kernel and the original source kernel under Frobenius Norm :

$$\begin{gathered} \min_{\Lambda} ||\overline{\mathbf{K}}_Z - \mathbf{K}_Z||_F \\ \lambda_i \geq \zeta \lambda_{i+1}, i = 1, \ldots, n-1 \\ \lambda_i \geq 0, i = 1, \ldots, n \end{gathered} \tag{8}$$

where $\mathbf{\Lambda} = diag(\lambda_1, ..., \lambda n)$ are the non-negative eigen spectrum parameters. Here note that $\zeta \geq 1$ is taken to increase the contribution of the topmost eigenvalues. By linear algebra, the above norm reduces to :

$$\begin{gathered} \min_{\lambda} \lambda^T \mathbf{Q} \lambda - 2\mathbf{r}^T \lambda \\ \mathbf{C}\lambda \geq \mathbf{0} \\ \lambda \geq \mathbf{0} \\ \text{where } \mathbf{Q} = (\overline{\mathbf{\Phi}}_Z^T \overline{\mathbf{\Phi}}_Z) \odot (\overline{\mathbf{\Phi}}_Z^T \overline{\mathbf{\Phi}}_Z) \\ \mathbf{r} = diag(\overline{\mathbf{\Phi}}_Z^T \mathbf{K}_Z \overline{\mathbf{\Phi}}_Z) \\ \mathbf{C} = \mathbf{I} - \zeta \bar{\mathbf{I}} \end{gathered} \tag{9}$$

Here $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix and $\bar{\mathbf{I}} \in \mathbb{R}^{n \times n}$, is the 1th-diagonal matrix with the nonzero elements $\bar{I}_{i,i+1} = 1, i = 1, \ldots, n-1$. Instead of using all the eigenvalues and vectors of $\mathbf{K}_X$, we choose only the top r eigenvalues to make the calculations faster. So $\overline{\mathbf{\Phi}}_Z \in \mathbb{R}^{m \times r}, \lambda \in \mathbb{R}^{r \times 1}$ and $\mathbf{Q} \in \mathbb{R}^{r \times r}$

3.5 Domain - Invariant Kernel

After solving the optimization problem formulated above, it is straight forward to construct the domain invariant kernel $\overline{\mathbf{K}}_A$ on the source and target data $A = Z \cup X$. Based on the spectral design, the invariant kernel can be generated from the domain-invariant eigensystem $\{\mathbf{\Lambda}, \overline{\mathbf{\Phi}}_A\}$ as:

$$\overline{\mathbf{K}}_A = \begin{bmatrix} \overline{\mathbf{\Phi}}_Z \mathbf{\Lambda} \overline{\mathbf{\Phi}}_Z^T & \overline{\mathbf{\Phi}}_Z \mathbf{\Lambda} \mathbf{\Phi}_X^T \\ \mathbf{\Phi}_X \mathbf{\Lambda} \overline{\mathbf{\Phi}}_Z^T & \mathbf{\Phi}_X \mathbf{\Lambda} \mathbf{\Phi}_X^T \end{bmatrix} = \overline{\mathbf{\Phi}}_A \mathbf{\Lambda} \overline{\mathbf{\Phi}}_A^T \tag{10}$$

where $\overline{\mathbf{\Phi}}_A \stackrel{\Delta}{=} [\overline{\mathbf{\Phi}}_Z; \mathbf{\Phi}_X]$ are the extrapolated eigenvectors over the entire data of the source and target domain. Now the extrapolated source kernel which minimized the approximation error, $\overline{\mathbf{K}}_Z = \overline{\mathbf{\Phi}}_Z \mathbf{\Lambda} \overline{\mathbf{\Phi}}_Z^T$, is directly fed to the SVM for model building. Finally our classifier turns out to be :

$$\mathbf{y}_X = \overline{\mathbf{K}}_{XZ}(\alpha \odot \mathbf{y}_Z) + b \tag{11}$$

where $\overline{\mathbf{K}}_{\mathbf{XZ}} = \mathbf{\Phi}_X \Lambda \overline{\mathbf{\Phi}}_Z^T$ is obtained from the domain invariant kernel. For any out of sample prediction on a dataset X_0, a similar approximation, that we did earlier, of the eigensystem and cross domain kernel matrix is performed.

$$\mathbf{\Phi}_{X_0} = \mathbf{K}_{X_0 X} \mathbf{\Phi}_X \mathbf{\Lambda}_X^{-1} \tag{12}$$

$$\overline{\mathbf{K}}_{X_0 Z} = \mathbf{\Phi}_{X_0} \Lambda \overline{\mathbf{\Phi}}_Z^T \tag{13}$$

4 Experiments

4.1 Text Dataset

We tested on two text datasets namely - 20newsgroup and reuters. Both of these were collection of documents, each having the frequency of words in the documents, which was further preprocessed using TF-IDF. The newsgroup data was composed of four main categories : rec, sci, main and talk. Each of these categories were further subdivided into 4-5 subcategories. The reuters dataset was composed of five main categories : orgs, people, places, exchanges and topics. We carried out a binary classification task separately between two random categories, with both of the datasets. So, for the newsgroup data, we randomly picked two categories and used two subcategories each (totalling four), from both of those categories to form the source domain. The rest were put in the target domain. Such a construction leads to formation of $\sim$200 datasets. For feasibility purposes

we tried only on 12 datasets. A similar construction was done for the reuters dataset(without any subcategories). For comparative study, the cost parameter was empirically chosen to be 10.0 and ζ to be 2.0 and a linear kernel was used. The accuracies are shown in the table. As is clearly visible, TKL outperformed SVM in all the instances. Since all the subcategories had some difference in distribution, TKL was successfully able to transfer it and design a better model.

Dataset	SVM	TKL
NEWSGROUP		
comp vs talk	87.19	**94.00**
rec vs talk	84.80	**94.58**
sci vs talk	70.39	**87.03**
REUTERS		
org vs places	72.26	**77.64**
people vs places	55.52	**67.40**
places vs org	70.27	**77.08**

4.2 Image Dataset

For this experiment, we used a corpus of images obtained from a variety of sources, namely : Amazon(A), DSLR(D), Webcam(W) and Caltech(C). The features were scaled and an rbf kernel was used along with a cost of 10.0 and ζ as 1.1. Here the task at hand was a multilabel classification problem. All the different sources of images had the same categories of objects like : TV's, backpack's, monitor's etc. We selected two domains randomly and used them as source and target domain respectively. A similar trend as with the text dataset was observed, wherein TKL was able to give more accurate results.

Dataset	SVM	TKL
D - W	63.05	**86.10**
A - W	38.30	**43.08**
C - A	53.13	**56.10**
A - D	43.94	**47.13**

4.3 fMRI Dataset

We were motivated to try out TKL on an fmri dataset primarily because the brain image that is captured for different subjects varies in distribution. There are multiple reasons behind it. Firstly, the anatomical size of the brain and the level of brain activity varies across people. Secondly, even a slight movement of the head generates noise in the captured brain images. For those reasons, designing a generalized classifier across different subjects is a difficult task. The dataset we had was of six subjects. Each of them was shown a picture or a sentence and their corresponding brain activity was recorded. The classification task was to successfully classify, by a subject's brain image, whether the subject was looking at a picture or a sentence. We performed three experiments where we had equal number of subjects in both the target and the source domain, with a cost of 10 and ζ as 2 with a linear kernel. The number of subjects in those domains were varied from one to three .The accuracies are listed below in the table. Here, TKL gave either better or almost similar results, which suggests that perhaps with better feature engineering one might be able to get better results. However, nothing can be claimed conclusively.

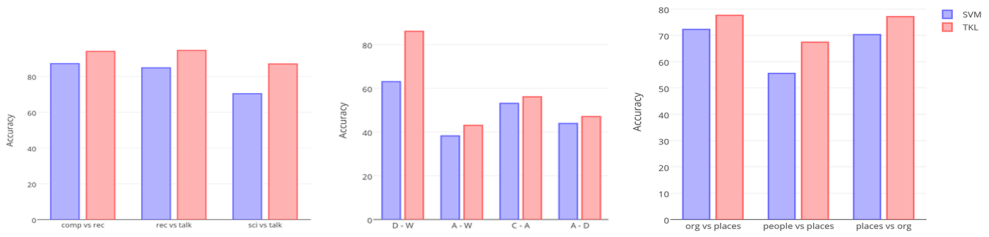

Figure 1: Accuracies obtained over text and image datasets

Dataset	SVM	TKL
1S - 1T	48.45	**53.00**
2S - 2T	50.00	**52.00**
3S - 3T	**54.00**	53.00

5 Scalable Implementation

Complexity of the proposed method is of $O(n^2)$, hence it is not viable for large data sets. But the Nyström method also allows us to approximate large kernel matrices. We start of with choosing a random subset of data from both the target and the source domain, so that we have the matrices, $\mathbf{K}_{X\hat{X}} \in \mathbb{R}^{n\times\hat{n}}$ and $\mathbf{K}_{Z\hat{Z}} \in \mathbb{R}^{m\times\hat{m}}$. Eigensystem of the matrix $\mathbf{K}_{\hat{X}}$ can be computed using eigendecomposition and can be used with equation (5) to find the target domain's eigensystem.

$$\mathbf{\Phi}_X \simeq \mathbf{K}_{X\hat{X}}\mathbf{\Phi}_{\hat{X}}\mathbf{\Lambda}_{\hat{X}}^{-1} \tag{14}$$

Similarly using equations (5) and (6), the extrapolated eigensystem of the source kernel, $\overline{\mathbf{\Phi}}_Z$, and the kernel matrix $\mathbf{K}_Z$, used in the QP, can be found out as :

$$\begin{aligned}\mathbf{\Phi}_{\hat{Z}} &\simeq \mathbf{K}_{\hat{Z}X}\mathbf{\Phi}_X\mathbf{\Lambda}_{\hat{X}}^{-1}\\ \overline{\mathbf{\Phi}}_Z &\simeq \mathbf{K}_{Z\hat{Z}}\overline{\mathbf{\Phi}}_{\hat{Z}}\mathbf{\Lambda}_{\hat{X}}^{-1}\\ \mathbf{K}_Z &\simeq \mathbf{K}_{Z\hat{Z}}\mathbf{K}_{\hat{Z}}^{-1}\mathbf{K}_{\hat{Z}Z}\end{aligned} \tag{15}$$

The overall complexity of our earlier approach comes out to be $O((r+p)(m+n)^2)$ where p is the dimension of the dataset. The scalable approach reduces this complexity to $O((r+p)(m+n)(\hat{m}+\hat{n}))$

6 Conclusion

The purpose of the project was to learn better models by reducing the distribution mismatch between training and testing data. This was achieved by using minimization of the Nyström appoximation error. Experiments that were conducted over text and image datasets showed that the domain invariant kernel was indeed able to capture the information and gave particularly better results than SVM. The accuracies obtained for text domain classification were significantly higher than those for image and brain data. We believe this happened due to the larger distribution difference between images as compared to text. For the fmri data, TKL showed either better or almost similar results, which suggests that perhaps with better feature engineering one might be able to get better results. Although TKL failed to perform as good on out of sample data, as SVM. Owing to this, the high accuracies for newsgroups dataset might actually be due to overfitting. More experiments and tests need to be conducted to better understand the behaviour. Further work might include using a regularizer in the optimization problem to keep a check on overfitting.

References

[1] M.Long, J.Wang, J.Sun, Philip S, "Domain Invariant Transfer Kernel Learning,"in *IEEE Transactions on Knowledge and Data Engg .,* 2014 ,pp. 1519-1532.

[2] K. Zhang, I.W. Tsang J.T. Kwok, "Improved Nyström low rank approximation and error analysis," in *Proc 25th Int. Conf. Mach. Learn.,* 2008 , pp. 1232-1239.

[3] K. Zhng, V.W. Zheng, Q. Wang, J.T. Kwok, Q.Yang and I Marsic, "Covariate shift in Hilbert Space : A solution via surrogate kernels",in *Proc 30th Int. Conf. Mach. Learn.,*2013,pp.388-395

[4] Image Data Source - K.Saenko, B.Kulis, M.Fritz, and T.Darrell,"Adapting visual category models to new domains," in *Proc. 11th Eur. Conf. Comput. Vis.,* 2010,pp.213-216