SIMPLIFIED DOMAIN TRANSFER MULTIPLE KERNEL LEARNING FOR LANGUAGE RECOGNITION

Jiaming Xu[1,2]*, Jia Liu*[3]*, Shanhong Xia*[1]

[1] State Key Laboratory on Transducing Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100190, China
[3] National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
Email:xujiaming09@gmail.com

ABSTRACT

Distribution mismatch between training and test data can greatly deteriorate the performance of language recognition. Some effective methods for compensation have been proposed, such as nuisance attribute projection (NAP). In real-world applications, there are often sufficient training samples from a different domain and only a limited number of labeled training samples from target domain, performance of a system will be degraded and needs to be further improved. In this paper, we introduce transfer learning to solve this problem. We propose a novel transfer learning algorithm referred to as simplified domain transfer multiple kernel learning (SDTMKL). Our aim is to discover a good representation of feature space that minimizes the distribution mismatch between samples from the source and target domains. Robust models can be learned in this suitable feature space. Results on a NIST language recognition task show that the SDTMKL method is quite effective and can further improve system performance when combined with NAP.

Index Terms— Transfer Learning, Language Recognition, Support Vector Machine, Multiple Kernel Learning.

1. INTRODUCTION

A major assumption in many conventional machine learning algorithms is that the training and test data are drawn from the same feature space and the same distribution. When the distribution changes the model needs to be rebuilt from scratch using newly collected training data, otherwise the performance of the algorithm will be greatly deteriorated by the distribution mismatch. At the same time, it is nearly impossible to get robust models using only a limited number of newly labeled training samples which match the distribution of the test data. In real-world applications, however, it is quite expensive and time consuming to collect sufficient newly labeled training data.

In language recognition, we face the same problem. Mismatch between train and test conditions is usually due to variability from changes in channel, environment and other factors. It is quite crucial to reduce this mismatch. Recent years, some effective algorithms have been applied to this task and have considerably improved the performance of language recognition systems. One of the most popular methods is nuisance attribute projection (NAP) [1] which is a typical subspace method for model compensation. It has been widely used in language and speaker recognition and shown promising results. Usually, we need a large amount of labeled samples having the same feature space and distribution as that of test data to train NAP projection in order to get good performance. However, this demand can not always be met. In real-world applications, there are often sufficient training samples from a different domain and only a limited number of newly labeled training samples from the target domain, performance of a system will be degraded and needs to be further improved.

Recently, transfer learning (or cross-domain learning) has emerged as a new learning framework to address this kind of problems [2]. It can train relatively robust models with only a limited number of labeled data from target domain and a large amount of labeled training data from source (or auxiliary) domain. There is increasing research interest in it and some new methods have been proposed and successfully used in some real-word applications, such as object category recognition [3], WiFi localization [4]. Very recently, a novel algorithm referred to as domain transfer multiple kernel learning (DTMKL) [5] has been proposed. It simultaneously learns a kernel function and a robust classifier by minimizing both the structural risk function and the distribution mismatch between the samples from the source and target domains.

In this paper, we introduce transfer learning into language

This work was supported by the NSFC under grant No. 61273268, No. 61005019, No. 90920302 and project KZ201110005005 supported by Beijing Natural Science Foundation Program.

recognition to solve the problem mentioned. We simplify the DTMKL algorithm and name it as simplified domain transfer multiple kernel learning (SDTMKL). It takes advantage of the form of multiple kernel learning (MKL) [6] in order to discover a suitable higher feature space minimizing the distribution mismatch between data from different domains. We effectively combine it with NAP and robust models are learned under this framework. The performance of system is further improved.

The outline of the paper is as follows. In section 2, we briefly introduce the criteria referred to as Maximum Mean Discrepancy (MMD) which is used to compare data distributions. In section 3, we briefly describe the MKL. Section 4 illustrates the SDTMKL method. In section 5, we demonstrate the potential of the approach by applying it to language recognition task.

2. MAXIMUM MEAN DISCREPANCY

Let us denote the labeled data from source domain D^s as (x_i^s, y_i^s), $i = 1, 2, ...N_s$, where y_i^s is the label of x_i^s and N_s is the size of D^s. Similarly, D^t, (x_i^t, y_i^t) and N_t represent the target domain. Given samples $\{\mathrm{x}_i^s\}$ and $\{\mathrm{x}_i^t\}$, there exist many criteria (such as the Kullback-Leibler (KL) divergence) that have been used to estimate their distribution distance. However, many of these criteria are parametric or require intermediate density estimate. Recently, Borgwardt and Gretton et al. [7] have designed an effective nonparametric criterion, referred to as Maximum Mean Discrepancy (MMD) for comparing data distributions based on Reproducing Kernel Hilbert Space (RKHS) distance. Let us denote the kernel-induced feature map as ϕ. The empirical estimate of MMD between D^s and D^t is

$$\mathrm{MMD}(D^s, D^t) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \phi(\mathrm{x}_i^s) - \frac{1}{N_t} \sum_{i=1}^{N_t} \phi(\mathrm{x}_i^t) \right\|_{\mathcal{H}}^2 \quad (1)$$

The feature map ϕ transforms the samples into a higher or even infinite feature space, and the inner product of $\phi(\mathrm{x}_i)$ and $\phi(\mathrm{x}_j)$ equals to the kernel function $K(\cdot,\cdot)$, namely, $K(\mathrm{x}_i, \mathrm{x}_j) = \phi(\mathrm{x}_i)^T \phi(\mathrm{x}_j)$. Note that when MMD asymptotically approaches zero, two distributions of high-dimensional feature space are the same or close to each other [8]. So it is quite critical to minimize the MMD in order to reduce the mismatch between data from source and target domains.

3. MULTIPLE KERNEL LEARNING

An support vector machine (SVM) [9] is a two-class classifier constructed from sums of a kernel function $K(\cdot,\cdot)$,

$$f(\mathrm{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathrm{x}, \mathrm{x}_i) + b \quad (2)$$

where the y_i are labels, $\sum_{i=1}^{N} \alpha_i y_i = 0$, and $\alpha_i > 0$. The vector x_i are support vectors trained by an optimal algorithm.

In SVMs, it is usually uncertain which kernel is the most suitable for the task at hand. Intuitively, combining several kernels available can be a good choice. So was born the multiple kernel learning. Learning a optimal set of kernel weights $\boldsymbol{\beta} = [\beta_1, ..., \beta_M]^T$ for kernel combination is referred to as MKL problem. Given a set of M kernels, $\{K_1(\cdot,\cdot), ..., K_M(\cdot,\cdot)\}$, a combined kernel $K(\cdot,\cdot;\boldsymbol{\beta})$ can be defined as the weighted sum of the individual kernels, namely

$$K(\cdot,\cdot;\boldsymbol{\beta}) = \sum_{i=1}^{M} \beta_i K_i(\cdot,\cdot) \quad (3)$$

where β_i are the weights of each kernel function and meet the constraint, $\sum_{i=1}^{M} \beta_i = 1$ and $\beta_i > 0$. And then an SVM classifier is learned using the kernel defined in (3). During test stage, the score defined in equation (2) is adjusted to

$$f(\mathrm{x}) = \sum_{i=1}^{N} \alpha_i y_i \sum_{j=1}^{M} \beta_j K_j(\mathrm{x}, \mathrm{x}_i) + b \quad (4)$$

4. SIMPLIFIED DOMAIN TRANSFER MULTIPLE KERNEL LEARNING

4.1. DTMKL

The kernel defined in equation (3) is a linear combination of a set of base kernels, and we should note that the combination is equivalent to a weighted concatenation of the associated feature spaces, namely

$$\boldsymbol{\phi}(\mathrm{x};\boldsymbol{\beta}) = \begin{bmatrix} \sqrt{\beta_1}\phi_1(\mathrm{x}) \\ \cdot \\ \cdot \\ \cdot \\ \sqrt{\beta_M}\phi_M(\mathrm{x}) \end{bmatrix} \quad (5)$$

As has been mentioned, the MKL problem is to learn a optimal set of kernel weights by some complex methods. It also means to discover a weighted higher feature space defined in (5) which is the most suitable for the task at hand. Inspired by this motivation, we hope to find a 'good' representation of feature space that minimizes the distribution mismatch of data from different domains by using multiple kernels. In [5], a novel method called domain transfer multiple kernel learning was proposed. It simultaneously considers minimizing the structural risk function and the distribution mismatch, namely

$$\underset{K,f}{\operatorname{argmin}} \left(\mathrm{MMD}_K(D^s, D^t) + \theta R(K, f, D) \right) \quad (6)$$

where the first term represents the distribution mismatch described by multiple kernel K and the second term is the structural risk function, $\theta > 0$ is the tradeoff parameter and f is the SVM classifier defined in equation (2).

4.2. SDTMKL

Our goal is to find a 'good' feature space in which the MMD defined in equation (1) is minimized, so we take only the first term of (6) to discover this space and then a SVM classifier is used to learn robust models. The details are as follows.

First, the most important thing is to explicitly represent the MMD in the form of MKL. We denote a vector l with $N_s + N_t$ entries, in which the first N_s entries are set as $1/N_s$ and the rest are set as $-1/N_t$. Given a feature map ϕ, let $\Phi = [\phi(\mathrm{x}_1^s), ...\phi(\mathrm{x}_{N_s}^s), \phi(\mathrm{x}_1^t)...\phi(\mathrm{x}_{N_t}^t)]$, the MMD can be written in terms of the kernel matrix defined by ϕ, as:

$$\mathrm{MMD}(D^s, D^t) = trace(KL) \tag{7}$$

where

$$K = \Phi^T\Phi = \begin{bmatrix} K_{s,s} & K_{s,t} \\ K_{t,s} & K_{t,t} \end{bmatrix} \in \mathbb{R}^{(N_s+N_t)\times(N_s+N_t)}$$

K is a composite kernel matrix with $K_{s,s}$, $K_{t,t}$ and $K_{s,t}$ being kernel matrices on the data from the source domain, the target domain and the cross domain, respectively, and $L = ll^T \in \mathbb{R}^{(N_s+N_t)\times(N_s+N_t)}$

$$L_{ij} = \begin{cases} \frac{1}{N_s^2} & \mathrm{x}_i, \mathrm{x}_j \in D_s, \\ \frac{1}{N_t^2} & \mathrm{x}_i, \mathrm{x}_j \in D_t, \\ -\frac{1}{N_s N_t} & otherwise. \end{cases}$$

Using multiple kernel, the equation (7) can be adjusted to

$$\mathrm{MMD}(D^s, D^t) = trace(\sum_{i=1}^{M} \beta_i K_i L) \tag{8}$$

Now, the problem becomes minimizing the MMD defined in (8). In order to facilitate the calculation, here we optimize the following objective function

$$\begin{aligned} \frac{1}{2}\mathrm{MMD}(D^s, D^t)^2 &= \frac{1}{2}(trace(\sum_{i=1}^{M} \beta_i K_i L))^2 \\ &= \frac{1}{2}\boldsymbol{\beta}^T \boldsymbol{p}\boldsymbol{p}^T \boldsymbol{\beta} \end{aligned} \tag{9}$$

where $\boldsymbol{p} = [p_1, ..., p_M]^T, p_i = trace(K_i L)$

It is obvious that (9) is a typical convex quadratic problem and many mathematical methods can be applied to figure out an optimal set of weights $\boldsymbol{\beta}$. Here the gradient decent method is used. The gradient of h in (9) is

$$\nabla h = \boldsymbol{p}\boldsymbol{p}^T\boldsymbol{\beta} \tag{10}$$

So we can update the weight of multiple kernels as follow

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \lambda_t \nabla h \tag{11}$$

where ∇h is the updating direction and λ_t is the learning rate at tth iteration.

The whole procedure of the SDTMKL Algorithm is summarized in Algorithm 1.

Algorithm 1 Framework of SDTMKL Algorithm

1: Given a set of M kernels, initialize the weight $\boldsymbol{\beta} = [\frac{1}{M}, ...\frac{1}{M}]$
2: Calculate the kernel matrix $K_i, i = 1, ..., M$, and the matrix L
3: **for** $t = 1, ..., T$ **do**
4: Calculate the updating direction ∇h using (10) and update the weight $\boldsymbol{\beta}$ using (11);
5: **end for**
6: Combine the kernel matrices using the updated weight $\boldsymbol{\beta}$ and a SVM classifier is used to learn robust models.

5. EXPERIMENTS

5.1. Experimental Setup

The experimental setup for this work is based on the NIST 2009 Language Recognition Evaluation(LRE). The 2009 LRE consisted of 23 linguistic classes. Here we choose 8 of them, including Cantonese, Farsi, Hindi, Korean, Mandarin, Russian, Urdu, and Vietnamese. The reason why we choose these 8 languages is that both conversational telephone speech utterances(CTS) and narrow band telephone segments from Voice of America broadcasts(VOA) are available to them as data sources.

Two data set are used in this work: "train" and "test". Here the CTS set is referred to as the target domain. Similarly, the VOA set is referred to as source domain. The "test" set only consists of all the CTS segments from the 2009 LRE task. For each language, 1000 segments from VOA (a large amount of samples from source domain) and 100 segments from CTS (a limited number of samples from target domian) are used for training models. The VOA samples for "train" consist of narrow band segments from VOA broadcasts. The CTS samples for "train" set come from previous NIST LRE (2007) and the CallFriend, CallHome, OGI-22 collections. And we randomly choose 1000 segments from the VOA "train" set and 100 segments from the CTS "train" set for each language, respectively. So we have 8800 samples for in "train" date set.

Experiments are performed on the 8 languages closed-set task. The criterion for evaluation is EER.

For feature extraction, we first extract 13-dimensional MFCC features and the cepstral features are processed with RASTA filtering. Then SDCC features are used with a 7-1-3-7 parameterization.

A language and gender independent UBM is trained using all of the training data with 8 iterations of EM adapting all parameters-means, mixture weights and diagonal covariances. The number of mixture components is 512. For GMM MAP training, only means are adapted. Then, a GMM Supervector (GSV) is extracted for each segment.

In our systems, we use the GSVs extracted in advance as input features and take LIB-SVM as our classifier.

System	30s	10s	3s
GSV-SVM	12.14	19.16	28.74
NAP	9.73	18.00	29.54
SDTMKL	9.75	18.38	28.64
NAP + SDTMKL	8.50	16.63	28.40

Table 1. Comparison of EERs (%) for different training methods at different durations for the 8 proposed languages of LRE09.

For the baseline system, the 8800 GSVs for training are directly sent into the SVM with linear kernel (i.e., $K(\mathrm{x}_i, \mathrm{x}_j) = \mathrm{x}_i^T \mathrm{x}_j$). And a score is calculated according to equation (2). This is the common method referred to as GSV-SVM system which has been widely used.

For the NAP estimation, all the 800 samples from CTS are used to train the NAP projection. All the GSVs from "train" and "test" set are processed with NAP. Then, the GSV-SVM system is used.

For our SDTMKL system, two types of base kernels are used: linear kernel and polynomial kernel (i.e., $K(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i^T \mathrm{x}_j + 1)^a$), we set a = 1.1, 1.2, 1.5, 1.8, 2.0, 2.3, 2.5 empirically. So we get eight kernels and then the proposed SDTMKL algorithm is used to learn an ideal set of weights of the base kernels. A weighted higher feature space defined in equation (5) is discovered and theoretically, it is the most suitable feature space in which the distribution mismatch between data from the VOA and CTS set is minimum. At last, the SVM classifier is used to learn robust models in this optimal feature space. During the test stage, a score is calculated according to equation (4).

At last, we combine the NAP and our SDTMKL framework. All the GSVs processed with NAP are sent into the STDMKL system.

5.2. Experimental Result

Results for the various systems are shown in Table 1. First, we should note that the performance of the baseline GSV-SVM system is greatly deteriorated because only a limited number of samples which match the target domain and a large number of samples from a different domain are used to train models. Second, both NAP and the SDTMKL algorithm are quite effective. They outperform the GSV-SVM system at 30s and 10s durations. Third, the SDTMKL can effectively combine with NAP and further improve the performance. Last, note that 3s is a quite difficult task and none of the methods can perform well at this duration.

6. CONCLUSION AND FUTURE WORK

In this paper, we introduced transfer learning into language recognition to solve the problem of mismatch between train and test conditions. We presented a novel transfer learning technique SDTMKL to find a feature space that minimizes the mismatch. Our experiment demonstrated that our method can reduce the sensitivity to the mismatch. In addition, it can combine with NAP to further effectively improve the performance of the original system. To our knowledge, this is the first time such an approach is applied to language recognition. In the future, we plan to apply our method to other exiting algorithms for compensation, such as i-vector. Besides, we will further explore some other kinds of base kernels.

7. REFERENCES

[1] W.M. Campbell, D.E. Sturim, D.A. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, may 2006, vol. 1, p. I.

[2] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345 –1359, oct. 2010.

[3] Yi Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, june 2010, pp. 1855 –1862.

[4] Zhuo Sun, Yiqiang Chen, Juan Qi, and Junfa Liu, "Adaptive localization through transfer learning in indoor wi-fi environment," in *Machine Learning and Applications, 2008. ICMLA '08. Seventh International Conference on*, dec. 2008, pp. 331 –336.

[5] Lixin Duan, I.W. Tsang, and Dong Xu, "Domain transfer multiple kernel learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 465 –479, march 2012.

[6] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan, "Multiple kernel learning, conic duality and the smo algorithm," in *Proc. ICML*, 2004.

[7] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B.Schölkopf, and A.J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 4, pp. 49 –57, 2006.

[8] A. J. Smola, A. Gretton, L. Song, and B. Schölkopf, "A hilbert space embedding for distributions," in *Proc. 18th Int. Conf. Algorithmic Learn.Theory, Sendai, Japan*, Oct 2007, pp. 13–31.

[9] Nello Cristianini and John Shawe-Taylor, "Support vector machines," *Cambridge University Press, Cambridge*, 2000.