Domain Transfer via Multiple Sources Regularization

Shaofeng Hu[1], Jiangtao Ren[1], Changshui Zhang[2,3,4], and Chaogui Zhang[1]

[1] School of Software, Sun Yat-sen University, Guangzhou, P.R. China
[2] Department of Automation, Tsinghua University, Beijing, P.R. China
[3] State Key Lab. of Intelligent Technologies and Systems, Beijing, P.R. China
[4] Tsinghua National Laboratory for Information Science and Technology (TNList)
{hugoshatzsu,daguizhang}@gmail.com, issrjt@mail.sysu.edu.cn,
zcs@mail.tsinghua.edu.cn

Abstract. The common assumption that training and testing samples share the same distribution is often violated in practice. When this happens, traditional learning models may not generalize well. To solve this problem, domain adaptation and transfer learning try to employ training data from other related source domains. We propose a multiple sources regularization framework for this problem. The framework extends classification model with regularization by adding a special regularization term, which penalizes the target classifier far from the convex combination of source classifiers. Then this framework guarantees the target classifier minimizes the empirical risk in target domain and the distance from the convex combination of source classifier simultaneously. By the way, the weights of the convex combination of source classifiers are embedded into the learning model as parameters, and will be learned through optimization algorithm automatically, which means our framework can identify similar or related domains adaptively. We apply our framework to SVM classification model and develop an optimization algorithm to solve this problem in iterative manner. Empirical study demonstrates the proposed algorithm outperforms some state-of-art related algorithms on real-world datasets, such as text categorization and optical recognition.

Keywords: domain adaptation, multiple sources regularization.

1 Introduction

The common assumption that training and testing samples share the same distribution is often violated in practice. When this happens, traditional learning models may not generalize well even with abundant training samples. *Domain Adaptation* is one of these situations where little labeled data is provided from target domain, but large amount of labeled data from source domains are available. Domain adaptation methods [1,2] learn robust decision function by leveraging labeled data both from target and source domains which usually don't share the same distributions. This problem involves in many real world application such as natural language processing[3], text categorization[4], video concept detection[5], WiFi localization[4], remote sensor network[2], etc.

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 522–533, 2014.

Most of domain adaptation methods can be classified into two classes according to their strategies of adapting source information: either with sources labeled data or with sources classifier. The former strategy selects source labeled samples that match target distribution to overcome distribution discrepancy. For example, [6] predicts unlabel samples via an ensemble method in local region including labeled samples of sources. [7] iteratively draws sources labeled samples that are in the same cluster with target labeled data in projected subspace. Alternately, the latter strategy try to get the final target classifier by weighted sum of target classifier f_T trained from target domain data and multiple source classifiers $\{f_{S_1}, f_{S_2}, \ldots, f_{S_m}\}$ trained from source domain data. [8] seeks a convex combination of f_T and $\frac{1}{m}\sum_k f_{S_k}$ by cross validation. [9] proposes Adaptive Support Vector Machine (ASVM) to learn f_T by incorporating the weighted sum of source classifiers $\sum_k \lambda_k f_{S_k}$ into the objective function of SVM, where λ_k is evaluated by a meta-learning procedure. [10] obtains the final f_T by maximizing output consensus of source classifiers. [11] modifies f_T and penalizes the output difference between f_T and each f_{S_k} on unlabeled data.

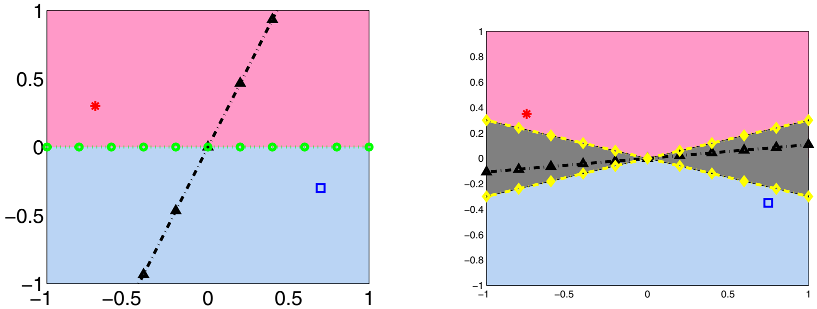
We focus on the strategy of adapting source classifiers in this paer. Based on the related works, it can be summarized one of the simplest methods to adapt source classifiers is treating their weighted sum as a single classifier. However, performance of this strategy is dependent on the weights for target and sources classifiers. It would be appropriate to assign higher weights to sources that are more similar with target domain. To our best of knowledge, although a few works have been addressed on domain weights assignment, little of them try to learn the appropriate weights automatically. [8] weights each source equally. [9] evaluates weights by meta-learning algorithm which is not promising since features of meta-learning are only dependent on the output of source classifiers. [11] determines domain weights by estimating the distribution similarity by MMD.

In this paper, we propose a novel way of adapting source classifiers by considering multiple source classifiers as prior information. Instead of learning the combination weights of target and source classifiers explicitly, we learn the target classifier directly from target domain data while keeping the target classifier approximates a convex combination of source classifiers as closely as possible, and the convex combination weights of source classifiers will be learned jointly with the learning of target classifier through optimization methods.

To illuminate the motivation of our paper, let us consider an example in Figure 1. Because of the rareness of labeled data in target domain, it is hard to learn a good target classifier directly. For example, in Figure 1 (a), only one labeled sample of each class is provided, denoted by $\Box/*$ respectively. There exists a very large classifier space in which every classifier can separate the training samples well with high uncertainty on test samples however. As depicted in Figure 1(a), the horizontal hyperplane (solid line with circle) generalizes best based on the real classes distribution indicated by different colors. But we will get a bad hyperplane (dotted line with triangular) by large margin principle[12]. However, by the introduction of some useful prior information contained in related source domains, we can improve the target classifier performance on test samples.

We can restrict the target classifier approximates the convex combination of source classifiers, because we think the convex combination of the source classifiers is a compact version of the source classifiers. In this way, we can exploit every source classifier with high confidence. Further, when we add the convex combination of the source classifiers as a regularization term to object function, it will shrink the search space of target classifier greatly and provide a good way to optimize it. For example, Figure 1(b) presents two source classifiers (dotted line with diamond), and a gray region which represents the convex combination space of the two source classifiers. It is clear that if the target classifier is in or near to the convex combination space of source classifiers, the target classifier (dotted line with triangular in Figure 1 (b)) will have better generalization performance than the one learned by large margin principle.

Therefore, we propose a multiple sources regularization framework based on the above motivation. The framework extends general classification model with regularization by adding a special regularization term, which penalizes the target classifier far from the convex combination of source classifiers. Then this framework make sure the target classifier minimizes the empirical risk in target domain and the distance from the convex combination of source classifier simultaneously. By the way, the weights of the convex combination of source classifiers are embedded into the learning model as parameters, and will be learned through optimization algorithm automatically, which means our framework can identify similar or related domains adaptively. we propose an iterative algorithms to solve this optimization problem efficiently.



(a) No Transfer

(b) Transfer with multiple sources regularization

Fig. 1. Intuitive example about multiple sources regularization

2 Multiple Sources Regularization Framework

To solve multiple sources domain adaptation problem, we propose a multiple sources regularization framework. Supposed there exist m source domain data sets, denoted by $S = \{S_1, S_2, \ldots, S_m\}$. We assume that all the samples in source data sets are labeled, etc. $S_k = \{X_{s_k}, y_{s_k}\}$ and $|X_{s_k}| = |y_{s_k}|$ for all $k \in \{1, \ldots, m\}$. y_{s_k} is output variable, which can be either continuous or discrete. Correspondingly, target domain is unique and is divided into labeled training set and unlabel testing set, etc. $T = \{(X_L, y_L), X_U\}$. A multiple sources domain adaptation problem can be summarized as: (a) each source domain has a different but similar distribution with target domain, $Pr_{S_k}(X, Y) \neq Pr_T(X, Y)$. (b) scale of training set of target domain is much smaller than that of test set, $|X_L| \ll |X_U|$. (c) source and target domain share the same output variable. (d) the objective of multiple sources domain adaptation is to utilize source data S to improve learning performance of target domain T. We firstly discuss our regularization framework under general linear form. Then the framework is extended to RKHS (Reproducing Kernel Hilbert Space) with SVM hinge loss function for classification problem. Thirdly, an iterative optimization algorithm is proposed to efficiently solve multiple sources regularization SVM.

2.1 Multiple Sources Regularization Framework

In this section, multiple source regularization (MSR) framework is introduced for classification. We start from the linear classification model. Linear model is more intuitive and geometrically interpretable. Denote linear predictive function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, where $\mathbf{w}$ is feature weights and b is bias term of separating hyperplane. Learning algorithms seek to find optimum $\mathbf{w}$ and b that minimize structural risk, such as hinge loss. Generally, structural risk trade off between empirical risk and regularization:

$$\min_{\mathbf{w},b} \sum_{i=1}^{l} L(x_i, y_i; \mathbf{w}, b) + \lambda \Phi(\mathbf{w}) \tag{1}$$

$L(\cdot)$ is loss function while $\Phi(\cdot)$ is regularization term. $\Phi(\cdot)$ penalizes function complexity to avoid overfitting. When labeled data number l is large enough, Eq(1) is a tight upper bound of expected risk. However, under domain adaptation setting, training samples of T would be scarce. Therefore, structural risk will be too loose to be used as upper bound under supervised learning setting. As we know, loose bound created by Eq(1) will be tightened by introducing unlabeled data which is referred as semi-supervised learning. Alternately, our framework alleviates this problem by including multiple source classifiers trained from source domain labeled data. To do this, we modify Eq(1) by adding an extra regularization term as following:

$$\begin{aligned} \min_{\mathbf{w},b,\beta} \quad & \sum_{i=1}^{l} L(x_i, y_i; \mathbf{w}, b) + \lambda \Phi(\mathbf{w}) + \rho \|\mathbf{w} - W_s \beta\|_2^2 \\ s.t. \quad & \sum_{k=1}^{m} \beta_k = 1, \beta \geq 0 \end{aligned} \tag{2}$$

where $W_s = [w_{s1}, w_{s2}, \ldots, w_{sm}] \in R^{d\times m}$. w_{sk} is a feature weight vector learned from the k-th source domain S_k. Learning model of each source domain should be consistent with that of target domain in order to maintain homogeneity of model coefficients. The last regularization term of Eq(2) penalizes w far away from the convex combination of m source classifiers. $\rho > 0$ trade off between structural risk and multiple source regularization. Moreover, $\boldsymbol{\beta} \in \varDelta$ denotes the weight vector that determines convex combination of source classifiers. $\varDelta$ represents the m dimensional simplex: $\varDelta = \{\boldsymbol{\beta} : \sum_{k=1}^{m} \beta_k = 1, \beta_k \geq 0\}$. Our method of determining $\boldsymbol{\beta}$ also differs from other state-of-art multiple source domain adaptation methods: other than manual setting, meta learning or model selection, our framework embedded the auxiliary domain weights vector $\boldsymbol{\beta}$ into model Eq(2) as a parameter, then $\boldsymbol{\beta}$ can be learned by optimization method automatically. When $m = 1$, Eq(2) degenerate to a simple situation. $\boldsymbol{\beta}$ is fixed to be 1 which make Eq(2) much similar to Eq(1) from optimization aspect. Therefore, we only focus on the situation where $m > 1$ in Eq(2) in our paper.

2.2 Multiple Sources Regularization SVM (MSRSVM)

Loss function L in Eq(2) varies according to different models. It is easy to realize that our framework can be adapted to a wide variety of models including SVM, logistic regression, ridge regression and so on. SVM hinge loss is discussed in detail in the following. We choose SVM for discussion with the following reasons: (a) It is convenient to transform Eq(2) to its dual problem, extending linear model to kernel form.(b) SVM fits for problems with very little training samples which is consistent with the setting of domain adaptation.

Firstly, with hinge loss, Eq(2) can be reformulated as:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi},\beta} \quad & \sum_{i=1}^{l} \xi_i + \lambda \varPhi(\mathbf{w}) + \rho \|\mathbf{w} - W_s \boldsymbol{\beta}\|_2^2 \\ s.t. \quad & y_i(\boldsymbol{w}^T \boldsymbol{x} + b) \geq 1 - \xi_i, \xi_i \geq 0, i \in L \\ & \sum_{k=1}^{m} \beta_k = 1, \beta \geq 0 \end{aligned} \tag{3}$$

where $\boldsymbol{\xi}$ is the slack variable. From the viewpoint of optimization, Eq(3) is a QP problem. While Eq(3) is QP, it can be solved by numeric optimization package directly.

However, we transform Eq(3) to dual form instead of optimizing the prime problem directly for two reasons: a) variable dimension of Eq(3) is $d+n+m+1$ while dual problem shrinks to $n+m$. Optimizing the dual problem reduces the problem complexity. b) the dual problem can generalize the linear model to nonlinear case in RHKS (Reproduced Hilbert Kernel Space). It is worth to note that we do not transform all variables to dual problem. $\boldsymbol{\beta}$ remains fixed in prime form. This is because Eq(3) can be transform to an optimization whose structure is very closed to regular SVM dual problem without $\boldsymbol{\beta}$. Then the

Lagrange function of Eq(3) can be formulated as:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = C \sum_{i \in L} \xi_i &+ \frac{1}{2}\|\boldsymbol{w}\|^2 + \|\boldsymbol{w} - W_s \boldsymbol{\beta}\|^2 \\ &- \sum_{i \in L} \alpha_i \left(y_i \boldsymbol{w}^T \phi(\boldsymbol{x_i}) + b - 1 + \xi_i \right) \\ &+ \sum_{i \in L} \mu_i \xi_i \end{aligned}$$

Setting derivative of Lagrange function with (w, b, ξ) to zero and adding other constraints under KKT condition, we obtain the dual problem:

$$\begin{aligned} \max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} \quad & \alpha^T e - \frac{1}{2(1+\rho)} \left(\rho \beta^T \Sigma_s \beta - \alpha^T Y K Y \alpha \right) \\ & -\rho \beta^T \Omega \alpha \\ s.t. \quad & \alpha^T y = 0 \\ & 0 \leq \alpha \leq C \\ & \sum_{k=1}^{m} \beta_k = 1, \beta \geq 0 \end{aligned} \tag{4}$$

Where $\Sigma_s \in R^{m \times m}$ is a symmetric matrix representing correlation of feature weight among multiple source domains, analogous to covariance matrix of Gaussian distribution. Moreover, $\Omega\alpha$ is a vector related to the correlation of feature weight between target domain and source domain. Σ_s and Ω can be evaluated using definition of its element in kernel space:

$$\begin{aligned} \Sigma_s &= \begin{pmatrix} \alpha_{s_1}^T K_{s_1,s_1} \alpha_{s_1} & \alpha_{s_1}^T K_{s_1,s_2} \alpha_{s_2} & \cdots \\ \alpha_{s_2}^T K_{s_2,s_1} \alpha_{s_1} & \alpha_{s_2}^T K_{s_2,s_2} \alpha_{s_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \\ \Omega &= \begin{pmatrix} \alpha_{s_1}^T K_{s_1,t} \\ \vdots \\ \alpha_{s_m}^T K_{s_m,t} \end{pmatrix} Y \end{aligned} \tag{5}$$

Where $\boldsymbol{\alpha}$ and $Y = diag(y)$ denote dual variable and output diagonal matrix respectively. K are the kernel matrix constructed by input patterns from either source or target domains. We need to note that α_{sk} is optimized SVM dual variable training only on S_k.

Eq(4) appears to be more complicated than Eq(3). Actually, Eq(4) is a saddle-point minmax problem, which can be regard as a zero sum game between two players. In section 2.3, we develop a two stage iterative optimization algorithm to solve Eq(4) in a general framework. This problem is similar to the optimization problem referred in DIFFRC[13], SimpleNPKL[14] and SimpleMKL[15].

2.3 Iterative Optimization Algorithm for MSRSVM

The special structure of Eq (4) indicates that MSRSVM needs a customized optimization algorithm. Fortunately, many optimization algorithms have been proposed to solve similar min-max problems. The main idea of such algorithms is to separately optimize part of variables while keep others fixed. It turns out that Eq (4) can be decomposed into two subproblems.

The main steps of our optimization algorithm for MSRSVM is described as following. Denote $J(\alpha, \beta)$ as the objective function of Eq (4) and α^* and β^* as the optimal solution of model variables. The complicated min-max problem of Eq (4) can be decomposed into two simple optimization problems: minimizing $J(\alpha, \tilde{\beta})$ as well as maximizing $J(\tilde{\alpha}, \beta)$ where $\tilde{\alpha}$ and $\tilde{\beta}$ are fixed values. Therefore, after initialization of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the algorithm will loop over two steps until stop condition is met:

- Step 1: solve subproblem $\min_{0 \leq \alpha \leq C, \alpha^T y = 0} J(\alpha, \tilde{\beta})$ under fixed $\tilde{\beta}$.
- Step 2: solve subproblem $\max_{\sum_k \beta_k = 1, \beta \geq 0} J(\tilde{\alpha}, \beta)$ under fixed $\tilde{\alpha}$.

The optimization problem in Step 1 shares similar problem structure with common SVM. They differ only on the first order term of objective functions. The first order term of common SVM's objective function is all one vector $\boldsymbol{e}$ while Eq (4) has an extra negative term $\rho \beta^T \Omega$. Fast algorithms such as SMO or SVM^{light} could be adapted to solve this subproblem of MSRSVM without many modifications. Thus we optimize subproblem of Step 1 using a modified version of regular SVM algorithms. Without referring any specific implementation of SVM algorithm, we use SVMSolver (K, f, y) to define a general solver for SVM optimization where K, f, y denote kernel matrix, first order term and label vector respectively.

The subproblem of Step 2 is a classical QP problem. Since dimension of $\boldsymbol{\beta}$ is m and m is not large usually, Newton method is appropriate to optimize $\boldsymbol{\beta}$. However, as stated in section 2.1, Eq (4) is a saddle point minmax problem. If both optimization steps are taken to local optimum point, fluctuation happens and progress towards global optimum slows down. Therefore, as an alternately strategy we update $\boldsymbol{\beta}$ by taking one gradient step at each iteration. Regular gradient update formula can not be used here because the simplex constraint exists, and gradient method is for unconstrained optimization generally. In this paper, reduced gradient method is introduced to handle this simplex constraint optimization problem [15]. This method evaluates ascent gradient firstly, then projects the gradient into simplex using the formula stated below:

$$D = \begin{cases} 0 & \text{if } \beta_k = 0, \frac{\partial J_{\tilde{\alpha}}}{\partial \beta_k} - \frac{\partial J_{\tilde{\alpha}}}{\partial \beta_\mu} > 0 \\ -\frac{\partial J_{\tilde{\alpha}}}{\partial \beta_k} + \frac{\partial J_{\tilde{\alpha}}}{\partial \beta_\mu} & \text{if } \beta_k = 0, k \neq \mu \\ \sum_{\beta_\nu > 0} \left(\frac{\partial J_{\tilde{\alpha}}}{\partial \beta_\mu} - \frac{\partial J_{\tilde{\alpha}}}{\partial \beta_\nu} \right) & \text{for } k = \mu \end{cases} \tag{6}$$

Where $\frac{\partial J_{\tilde{\alpha}}}{\partial \beta_k}$ and $\frac{\partial J_{\tilde{\alpha}}}{\partial \beta_\mu}$ are the gradients of objective function with fixed $\tilde{\alpha}$, k and μ are vector indexes. Then we update $\boldsymbol{\beta}$ by using: $\beta_{t+1} = \beta_t + \eta D$. η

denotes the step length. Boyed [16] showed that when η is small enough at each iteration, global convergence could be guaranteed. η is choosen to be $O(\frac{1}{t})$. We use **objective gap** as convergence criterion. **Objective gap** represents absolute difference between the objective value after Step 1 and Step 2 within the same iteration. Algorithm 1 summarizes the whole iterative optimization algorithm.

Algorithm 1. Iterative optimization algorithm for MSRSVM

```
Input: target training data $(X_t, y_t)$, m source data sets $\{(X_{s_1}, y_{s_1}), (X_{s_2}, y_{s_2}), \dots, (X_{s_m}, y_{s_m})\}$;
Output: optimized variable $\alpha^*$ and $\beta^*$;
initialize $\alpha_0 = \mathbf{0}$ and $\beta_0 = \frac{1}{m}\boldsymbol{e}$;
for i=1 to m do
    construct kernel matrix $K_{s_i}$ using $X_{s_i}$, $K_{s_i,t}$ using $X_{s_i}$ and $X_t$;
    optimize corresponding dual variable $\alpha_{s_i}$ = SVMSolver $(K_{s_i}, e, y_{s_i})$;
    for j=1 to m do
        construct kernel matrix $K_{s_i,s_j}$ using $X_{s_i}$ and $X_{s_j}$;
    end for
end for
calculate $\Sigma_s$ and $\Omega$ by Eq (5) with $K_{s_i,s_j}$ and $K_{s_i,t}$ for $i, j \in \{1, \dots, m\}$;
while convergence criterion is not met do
    solve modified SVM subproblem, $\alpha_t$ = SVMslover $(K_t, (1+\rho)(\boldsymbol{e} - \rho\beta_t^T \Omega), y_t)$;
    calculate $D$ using Eq (6), then $\beta_t$ and update $\boldsymbol{\beta}$: $\beta_{t+1} = \beta_t + \eta D$, and $\eta = \frac{1}{t}$;
end while
```

3 Experiment

To demonstrate the effectiveness of our proposed framework MSRSVM, we perform experiments on multiple transfer learning data sets. They are real world data sets that frequently used in the context of transfer learning or multitask learning. Performance of MSRSVM are compared with some other state-of-art algorithms that can handle multiple source domains.

3.1 Data Sets and Experiment Setup

Three data collections are used in our experiment study, they are Reuters-21578[17], 20-Newsgroups[18] and Letters. Among them, Reuters-21578 and 20-Newsgroups are benchmark of text categorization for transfer learning. Letters is optical recognition dataset that is preprocessed for multitask learning.

Data Sets. All data sets that have been used in our experiment study are binary classification tasks. Reuters-21578 and 20-Newsgroups are both text categorization data collections with hierarchical class structure. For each dataset, we need to construct both target and source domain dataset. Target and source domain datasets are sampled from different subcategories of the same top categories. For example, for dataset ”comp vs rec”, its source task dataset is sampled from subcategories ”comp.windows.x” and ”rec.autos”, while target task dataset is sampled from subcategories ”comp.graphics” and ”rec.motocycles”. Therefore, source and target domain datasets share the same feature space but different words distribution. But in our multiple source adaptation setting, we need more than one source domain datasets for one target domain prediction task. To solve this problem, all the source domain datasets are grouped and shared as source domain datasets. For example, in 20-Newsgroups task, the source domain datasets of ”comp vs sci”, ”rec vs talk”, ”rec vs sci”, ”sci vs talk”, ”comp vs rec” and ”comp vs talk” constitute of the multiple source domains. While keeping the 6 source domains fixed, we can construct different multiple source adaptation problems with different target domain datasets.

For Letters dataset without hierarchical class structure, we build different learning tasks by randomly sampling from two different handwritten digit letters that are difficult to be distinguished. For example, ”c/e” denotes a prediction task that ”c” is the positive class while ”e” is the negative class. Each task is treated as target task and all the other tasks as source tasks. For example, if ”c/e” is target task, then task ”g/y”, ”m/n”, ”a/g”, ”i/j”, ”a/o”, ”f/t” and ”h/n” form the 7 source domain tasks.

Baseline. We compare the performance of MSRSVM with other SVM based learning algorithms which can cope with multiple sources adaptation problems. They are ASVM[9], FR[19], MCCSVM[8]and regular SVM without any transfer. ASVM can be obtained online, which is based on LibSVM and programmed in C++. Others including MSRSVM are implemented in matlab basing on SMO. ASVM combines source classifiers with weights by an independent meta learning algorithm. SVM classification parameter C is fixed to 10. Other related parameter are set to default values. Moreover, RBF kernel $k(x, y) = e^{-\sigma\|x-y\|^2}$ is chosen as kernel function, where σ is set to 0.0001 for text data and 0.01 for optical recognition. For MSRSVM, model parameter ρ is set to 1.

3.2 Performance Study

We adopt classification accuracy as evaluation metric to compare MSRSVM with other four state-of-art methods. All of the accuracy results in this paper are the average results of 10 experiments. The accuracy comparison results are summarized in Table 1 and Table 2 for text and Letters dataset respectively. Training ratio are fixed to 20% for text datasets, and 30 points are randomly selected as training set for Letters dataset. Note that the best results are highlighted in bold in the Table 1 and Table 2. On Reuters-21578 dataset, MSRSVM performs better than all of the baseline algorithms on all of the 3 tasks. For example, MSRSVM get the accuracy of 60.81% on **Pe vs PI** dataset, while ASVM get the accuracy of 59.27%, which is the best one of baseline methods. On 20-Newsgroup dataset, MSRSVM improves the accuracies significantly in most of time, comparing with the baseline methods. MSRSVM performs at least 3% better than regular SVM on 4 of 6 data sets. Meanwhile, MSRSVM outperforms other methods on all data sets except **Comp Vs Talk** where MCCSVM achieves highest accuracy, slightly

Table 1. Accuracy comparison on text data sets(%)

Method	Reuters			20-Newsgroup					
	O vs Pe	O vs Pl	Pe vs Pl	C vs S	T vs R	R vs S	S vs T	C vs R	C vs T
SVM	61.25	57.25	58.91	83.22	86.19	93.76	88.01	90.37	86.94
MCCSVM	60.73	55.63	58.46	84.37	87.80	89.27	89.66	88.06	**90.76**
FR	53.17	55.28	48.26	65.87	75.99	52.80	57.54	53.07	66.91
ASVM	57.28	56.68	59.27	83.00	64.80	76.43	50.23	51.12	65.26
MSRSVM	**62.50**	**59.40**	**60.81**	**85.92**	**91.40**	**94.76**	**93.02**	**93.38**	89.27

better than MSRSVM (less than 2%). Moreover, ASVM performs surprisingly poor on some tasks of 20-Newsgroup such as **Sci vs Talk** and **Comp vs Rec**, while MSRSVM behaves stable on all of the text datasets. Similar conclusions can be reached according to Letters dataset. MSRSVM achieves the highest accuracy on 5 of 8 datasets, and MCCSVM achieves on 3 of 8. And the performance of MSRSVM is still more stable than the others. Thus on the whole, MSRSVM significantly improves the accuracy most of the time.

Table 2. Accuracy comparison on Letters dataset(%)

Method	c vs e	g vs y	m vs n	a vs g	i vs j	a vs o	f vs t	h vs n
SVM	84.89	67.98	81.02	83.59	67.15	82.49	80.79	81.87
MCCSVM	84.07	68.99	87.58	**89.79**	**94.49**	**88.25**	78.14	90.14
FR	50.00	50.68	50.26	82.50	47.18	53.91	54.42	46.17
ASVM	78.11	50.00	50.38	60.64	37.60	50.32	52.21	62.71
MSRSVM	**89.48**	**71.52**	**87.59**	87.73	90.07	86.86	**83.20**	**91.10**

Performance of classifier may be dependent on the number of training data. When we refer training data here, it means training data of target domain. As mention before, samples of sources domains are fixed and all labeled. Figure 2 depicts the performance of MSRSVM, regular SVM and MCCSVM, with respect to different ratio or number of training data in target domain. Training data of target domain is assumed to be sparse in domain adaptation problem. Thus the ratio of training data varies from 0.05 to 0.3 for text datasets, and number of training data for Letters datasets varies from 18 to 38(1∼2% of the whole sample set) in the experiments. MSRSVM is compared with regular SVM and MCCSVM because they are more sensitive to the size of training data. Two important conclusions can be reached based on Figure 2. Firstly, performance of the three algorithms improve with the increase of the size of training data most of time. This is because the target classifier can get more information about target domain with more and more labeled data coming from target domain. Secondly, MSRSVM outperforms regular SVM and MCCSVM steadily most of time, especially when the size of training data are small. For example, MSRSVM wins on nearly all 20-Newsgroup data sets with only 5% of training data except for **Rec vs Sci**. Similar phenomena happens for Letters datasets. The accuracy gap between MSRSVM and MCCSVM is maximum when the number of training data is about 18-22. The curves also demonstrate MSRSVM can utilize the information of source domains more effectively than MCCSVM.

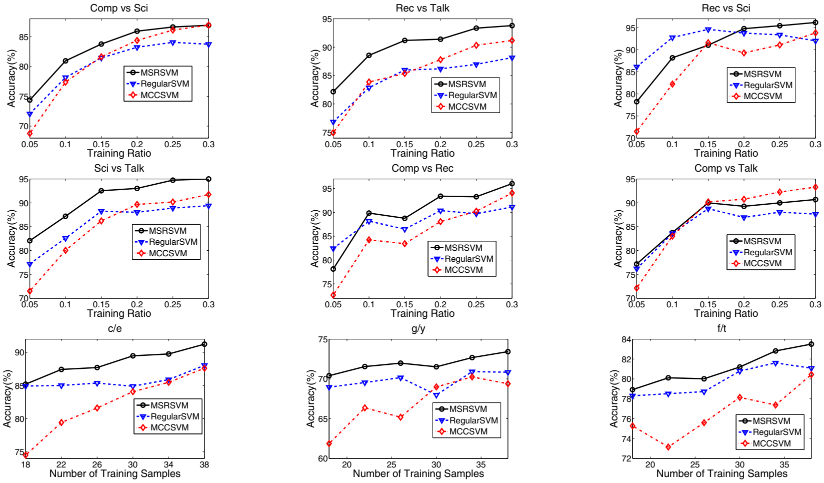


Fig. 2. Accuracy wrt ration or number of training data

4 Conclusion

We address multiple source domain adaptation problem in this paper. There exist more than one similar or related source domains whose distributions are not identical with the target domain. To adaptively utilize the information of sources domains and improve the performance of target classifier, we propose a simple framework named Multiple Source Regularization framework. This framework regularizes target classifier and make it approximate the convex combination of sources' classifier, while the combination weights will be learned adaptively. Our idea is that the sources information in regularization function acts as a prior to target domain. By substituting SVM's loss function into MSR framework, we propose a Multiple Source Regularization SVM (MSRSVM) model, and develop an optimization algorithm to solve this model in iterative manner. Experiments on both text and optical recognition datasets verify that MSRSVM outperforms many other state-of-art domain adaptation algorithms.

Acknowledgements. This work is supported by the Fundamental Research Funds for the Central Universities under grant 12lgpy40, Guangdong Natural Science Foundation under grant S2012010010390 and Beijing Municipal Education Commission Science and Technology Development Plan key project under grant KZ201210005007.

References

1. Crammer, K., Kearns, M., Wortman, J.: Learning from multiple sources. Journal of Machine Learning Research 9, 1757–1774 (2008)
2. Bruzzone, L., Marconcini, M.: Domain adaptation problems: A dasvm classification technique and a circular validation strategy. IEEE Trans. Pattern Anal. Mach. Intell. 32(5), 770–787 (2010)
3. Jiang, J., Zhai, C.: Instance weighting for domain adaptation in nlp. In: ACL. The Association for Computer Linguistics (2007)
4. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. In: IJCAI, pp. 1187–1192 (2009)
5. Duan, L., Tsang, I.W.-H., Xu, D., Maybank, S.J.: Domain transfer svm for video concept detection. In: CVPR, pp. 1375–1381. IEEE (2009)
6. Gao, J., Fan, W., Jiang, J., Han, J.: Knowledge transfer via multiple model local structure mapping. In: KDD, pp. 283–291 (2008)
7. Zhong, E., Fan, W., Peng, J., Zhang, K., Ren, J., Turaga, D.S., Verscheure, O.: Cross domain distribution adaptation via kernel mapping. In: KDD, pp. 1027–1036 (2009)
8. Schweikert, G., Widmer, C., Schölkopf, B., Rätsch, G.: An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In: NIPS, pp. 1433–1440 (2008)
9. Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. ACM Multimedia, 188–197 (2007)
10. Luo, P., Zhuang, F., Xiong, H., Xiong, Y., He, Q.: Transfer learning from multiple source domains via consensus regularization. In: CIKM, pp. 103–112 (2008)
11. Duan, L., Tsang, I.W., Xu, D., Chua, T.-S.: Domain adaptation from multiple sources via auxiliary classifiers. In: ICML, p. 37 (2009)
12. Vapnik, V.: Statistical Learning Theory. JohnWiley, NewYork (1998)
13. Bach, F., Harchaoui, Z.: Diffrac: a discriminative and flexible framework for clustering. In: NIPS (2007)
14. Zhuang, J., Tsang, I.W., Hoi, S.C.H.: Simplenpkl: simple non-parametric kernel learning. In: ICML, p. 160 (2009)
15. Szafranski, M., Grandvalet, Y., Rakotomamonjy, A.: Composite kernel learning. Machine Learning 79(1-2), 73–103 (2010)
16. Boyd, S., Xiao, L.: Least-squaures covariance matrix adjustment. SIAM Journal of Matrix Anal. Appl. 27, C532–C546 (2005)
17. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), `http://www.ics.uci.edu/mlearn/ML-Repository.html`
18. Davidov, D., Gabrilovich, E., Markovitch, S.: Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In: SIGIR, pp. 250–257 (2004)
19. Daumé III, H.: Frustratingly easy domain adaptation. In: Conference of the Association for Computational Linguistics (ACL), Prague, Czech Republic (2007)