Cross-domain Representation-learning Framework with Combination of Class-separate and Domain-merge Objectives

Wenting Tu and Shiliang Sun
Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, China
w.tingtu@gmail.com, slsun@cs.ecnu.edu.cn

ABSTRACT

Recently, cross-domain learning has become one of the most important research directions in data mining and machine learning. In multi-domain learning, one problem is that the classification patterns and data distributions are different among domains, which leads to that the knowledge (e.g. classification hyperplane) can not be directly transferred from one domain to another. This paper proposes a framework to combine class-separate objectives (maximize separability among classes) and domain-merge objectives (minimize separability among domains) to achieve cross-domain representation learning. Three special methods called DMCS_CSF, DMCS_FDA and DMCS_PCDML upon this framework are given and the experimental results valid their effectiveness.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – Data Mining

General Terms

Algorithm

Keywords

cross-domain learning, representation learning, discriminative model

1. INTRODUCTION

The representation learning aims to learn the meaningful and useful representations of the data that play important roles to many machine learning and data mining tasks (e.g. classification, regression and ranking). A very similar topic is called as dimensionality reduction. There are almost three subtopics on representation learning: feature selection, feature extraction and distance metric learning. The propose of feature selection methods is to find a subset of the original features (also called variables or attributes). Feature extraction tries to transform the data in the high-dimensional space to a new space. The data transformation may be linear while many nonlinear dimensionality reduction techniques also exist. Distance metric learning construct a metric or distance function which defines a distance between points of a dataset since the performance of many learning and data mining algorithms depend critically on their being given a good metric over the input space.

Most of traditional methods on representation learning perform well on the single domain scenario where distributions of source and target domains are identical. In other words, they are under the assumption that training and testing samples are independent and identically distributed. However, for lots of practical applications, this assumption does not hold since real-world problems may encounter multiple sources of data belongs to different feature distributions [5, 11]. The challenge of multiple domains is particularly relevant to hot real-world topics such as natural language processing, social networking, information retrieval. Therefore, it is meaningful to study cross-domain representation learning which can transfer common knowledge structures from source domains to the target domain to help the tasks on the target test datasets. For tasks on the cross-domain scenario, a good data representation should not only be helpful for later tasks but also can help task-related models be transferred among domains. Most of the traditional representation learning models only pursue the formal one, and the classification models constructed on this kind of data representation can not be transferred among domains.

The work in this paper focus on the cross-domain representation-learning framework which aims to learn a data representation model that can catch data characters which are not only helpful for later tasks , but also robust to the domain differences. The base idea of our work is to integrate task-related objectives and domain-merge objectives into a single one. For classification tasks, the task-related objectives always can be described as "maximize the differences between classes". For domain-merge objectives, a natural definition is "minimize the difference between domains". This paper proposes a framework to achieve both optimization of class-separate and domain-merge objectives. The advantages of our framework can be summarized as follows: First, our framework is easily understood and implemented since the domain-merge objectives are formulated by modifying the

KDD'12, August 12–16, 2012, Beijing, China.

definitions of traditional class-separate objectives. As a result, the companies or other non-research institutions can try our cross-domain representation model with engineering programming languages such as C, C++ or Java since there exists many toolkits that have implemented traditional classification models with those programming languages. Moreover, this paper proposes a framework rather than a special algorithm. Similar as many other frameworks that integrate a collection of methods that share same ideas in a united form [22, 10], our framework can motivate readers to design a greatly broad range of methods in cross-domain feature selection, cross-domain feature-construction, and cross-domain distance-metric-learning methods, which are flexible and interesting. I believe readers can easily develop their new methods upon our framework though only three special methods are proposed here.

Here we briefly introduce the concrete cross-domain representation learning methods that this paper proposes. In this paper, three concrete methods are proposed to provide concrete cross-domain representation learning methods and valid the extensibility of our framework. These methods are called as DMCS_CSF, DMCS_FDA and DMCS_PCDML. DMCS_CSF is proposed by changing the concept of class label to the domain label to revise the class-separate objective to the domain-merge objective. After that, the traditional correlation theory is modified to select the feature set that can reduce domain differences to merge domains. By combining class-separate and domain-merge evaluations, the final feature subset can pursue class-separate and domain-merge objectives. DMCS_FDA changes the mathematical items in traditional between-class and within-class scatter matrixes to obtain a new optimization objective that combines class-separate and domain-merge objectives. DMCS_CSF also integers two objectives into one objective function, and the optimization toolkits that used to solve traditional methods can be used here without modifies. To sum up, these methods aim to find a new data representation to pursue class-separate and domain-merge objectives simultaneity. Moreover, since there are one parameter to control the desired level of class-separate and domain-merge objective in these algorithms, "target-priority" cross-validation for cross-domain parameter selection is also presented.

In the next section we will describe our work: cross-domain representation learning framework. Firstly, the base ideas of our framework and domain-merge objectives are introduced. Subsequently, DMCS_CSF, DMCS_FDA (including its kernel version), DMCS_PCDML and "target-priority" cross validation are given in details. Section 3 will outline the experiments and analysis we performed. Finally, we will show our conclusion and recommendations for future work in Section 4.

2. CROSS-DOMAIN REPRESENTATION LEARNING FRAMEWORK

In this section, a cross-domain representation-learning framework that has a strong extended-ability and easy to implement is proposed. This is motivated by the fact that majorities of current domain-adaptation methods are not easily understood and implemented by engineering programming languages such as C, C++ [17] or Java [8]. We wish to encourage more companies try to employ cross-domain learning to real-world data. As a result, we propose methods that can be realized with small modifies with the traditional machine-learning toolkit.

Algorithm1 Cross-Domain Representation-Learning Framework (CDRLF)

Input:
Datasets $X_1, X_2, \cdots X_{n_s}$ from different domains, where $X_i \in \mathbb{R}^d$ $(i = 1, 2, \cdots, n)$ is on the raw data representation. Label information of labeled subsets. Domain information of $X_1, X_2, \cdots X_{n_d}$.

Output: The transformation operator $F_T(X_i) = \overline{X_i}$, where $\overline{X_i}$ is the new representation of X_i $(i = 1, 2, \cdots, n)$.

Objective function: $J(F_T)$ = combination of $Q_c(F_T)$ and $Q_d(F_T)$, where $Q_c(F_T)$ and $Q_d(F_T)$ indicate the class-separate related quality and domain-merge related quality, respectively.

Keys:
K_1**:** How to define the $Q_d(F_T)$ by modifying $Q_c(F_T)$.

K_2**:** How to define the combination of two terms $Q_c(F_T)$ and $Q_d(F_T)$.

As the Algorithm 1 shows, our framework aims to learn a data representation that achieves combination of class-separate objectives and domain-merge objectives. The domain-merge objectives are always achieved by modifying class-separate objectives, so that their implementations are relatively easy. The keys of developing a concert method with our framework are defining the formulation of domain-merge objectives and combination of class-separate and domain-merge objectives.

2.1 Domain-merge objectives

In this subsection, we discuss about the base idea to construct domain-merge objectives. To be easily understood and implemented, we wish to modify the class-separate objectives that can be implemented by current toolkits to form domain-merge objectives. Therefore, the concepts of class-separate and domain-merge tasks should be related. The components of class-separate tasks include training set, test set and objective functions. Here we list the differences of these three concepts in class-separate and domain-merge tasks in the Table 1.

In the following sections, we'll propose some methods obtained with our framework. This methods are obtained by changing classical representation-learning methods to the cross-domain version. As discussed in the Section 1, there are three categories of representation leaning: feature selection, feature reconstruction and distance metric learning.

Table 1: comparison of concepts in class–separate and domain–merge objectives

Concept	class–separate objective VS domain–merge objective
Training set	Set{x, class label} VS Set{x, domain label}
Test set	Set{x, predicted class label} VS − − −−
objective	maximize separability among classes VS minimize separability among domains

Special methods in these categories are proposed to provide concrete methods and show the extensibility of this framework.

2.2 Cross-domain Feature Selection

Feature selection [4] is a kind of explainable and visible statistics models to learn data representation. In this section, we wish to take use of the theory of one classical feature selection approach to gain a cross-domain feature selection method.

2.2.1 Correlation Feature Selection Theory

Here, we employ the theory of the Correlation Feature Selection (CFS) [7] to cross-domain feature selection. Note that there are many other feature-subset evaluations for traditional feature selection which evaluate the importance of a feature to the classification task can also be used.

CFS measure evaluates subsets of features on the basis of the following hypothesis: "Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other". The following equation gives the merit of a feature subset S consisting of k features:

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{1}$$

Here, $\overline{r_{cf}}$ is the average value of all feature-classification correlations, and $\overline{r_{ff}}$ is the average value of all feature-feature correlations. The CFS criterion is defined as follows:

$$CFS = \max_{S_k}\left[\frac{r_{cf_1} + r_{cf_2} + \cdots + r_{cf_k}}{\sqrt{k + 2(r_{f_1f_2} + \cdots + r_{f_if_j} + \cdots r_{f_kf_1})}}\right]. \tag{2}$$

The r_{cf_i} and $r_{f_if_j}$ variables are referred to as correlations such as Pearson's correlation:

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}. \tag{3}$$

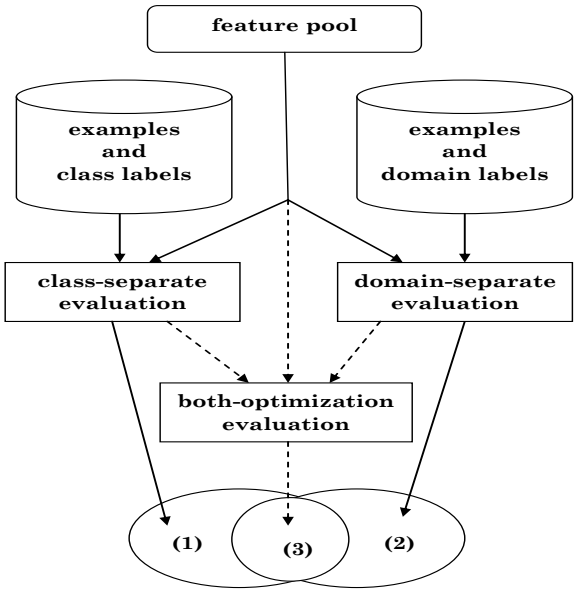

Figure 1: A general framework of cross-domain feature selection. Upon different evaluation criterions, different feature subsets will be obtained: (1) Feature subset that maximizes distances among classes, (2) Feature subset that maximizes distances among domains, (3) Feature subset that maximizes class-separate and domain-merge objective.

The concrete solution about (2) can be seen in [14].

2.2.2 Domain-Merge and Class-Separate CFS

Traditional feature selection models are mainly constituted by an evaluation and search components. The evaluation functions give scores to feature subsets while the search functions aim to search better subsets upon their scores. It is obvious that when different definitions of the evaluation functions are used, the final learned feature subsets will have different characters (as the Fig.1 shows) . As a result, we define a domain-merge related evaluation to estimate how much a feature subset reduces the domain differences by reversing the traditional feature-subset evaluation definition. Here, for the combination of two evaluations have two base ways. First (as the full line shows), we can obtain two subsets corresponding to the class-separate and domain-separate evaluations. Finally, the intersection of them can be regarded as the final feature subset. Second (as the dotted line shows), we can used a combination of two evaluations to be a new evaluation that integrates the class-separate and domain-separate objectives together.

Here, we modify the evaluation of CFS to obtain Domain-Merge and Class-Separate CFS (DMCS_CSF). Upon the CFS definition, the first term $\overline{r_{cf}}$ is used to estimate the feature importance to classification task. Therefore, if different definitions of labels to the classification task is used, the new evaluation can result different feature subset. Therefore, motivated by the discussion of Section 2.1, a domain-merged evaluation function is defined upon the domain labels. By changing the concept of the class label to the domain label,

the feature subset that has less correlation with domain differences (domain classification) will be selected out. Denote $\overline{r_{df}}$ is calculated by changing class labels to domain labels, two term r_{cf} and r_{df} that correspond to the class-separate and domain-merge objectives are usedčž

$$r_{cf/df} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i^{c/d} - \overline{Y^{c/d}})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i^{c/d} - \overline{Y^{c/d}})^2}}. \quad (4)$$

where Y^c and Y^d are class labels and domain labels. "/" means "or".

Therefore, similar as the fact that feature subset with high $\overline{r_{cf}}$ is assumed to be related to class differences, the feature subset corresponding to the low $\overline{r_{df}}$ is assumed to be less related to domain differences. Therefore, we wish to gain the feature subset that have high $\overline{r_{cf}}$ and low $\overline{r_{df}}$, as well as low feature-feature correlation $\overline{r_{ff}}$. The new evaluation is formulated as follows:

$$Merit_{S_k} = \frac{k\overline{r_{cf}} - \alpha * k\overline{r_{df}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (5)$$

Combining the class-separate related term $\overline{r_{cf}}$ and the domain-merge related term $-\overline{r_{cf}}$ with a parameter α the final feature subset is expected to be can maximize the class-separate and domain-merge objectives. As a result, we obtain $DMCS\_CSF$.

2.3 Cross-domain Feature Reconstruction

Compared with feature selection methods, feature reconstruction [20] methods are more flexible even the final reconstructed features are less explainable. Firstly, let's review a common framework of feature reconstruction methods that aims to learn a low-dimensional representation by learning a transformation. Here, we firstly focus on linear transformation framework, then the non-linear version is also proposed here.

Suppose raw data are d-dimensional samples vectors $x_i \in \mathbb{R}^d$ $(i = 1, 2, \cdots, n)$ and we wish to learn a low-dimensional representation of them. Assume r is the dimensionality we wish to reduce to. We need to learn a transformation matrix $\mathbf{\Phi} = [\phi_1, \phi_2, \cdots, \phi_r] \in \mathbb{R}^{d\times r}$ to get the representation $z_i \in \mathbb{R}^r$ for the raw samples x_i:

$$z_i = \mathbf{\Phi}^\top x_i. \quad (6)$$

To learn the transformation matrix $\mathbf{\Phi}$, many algorithms find an objective function related to some quality measure of the low-dimensional space. By optimizing the objective function, $\mathbf{\Phi}$ can be learned. Observing that the objective functions of many dimensionality reduction techniques developed so far are often related, we offer a dimensionality reduction framework with the object function formulated as:

$$J(\phi) = \phi^\top Q_{max}\phi, \quad (7)$$

Roughly speaking, Q_{max} encodes the quantity that we want to increase. Commonly, Q_{max} is related with class-separate degree. By reversing the definition of Q_{max}, we wish to obtain a item to estimate domain-merge degree and combine it into the final objective.

2.3.1 Fisher Discriminant Analysis Theory

Upon the above discussion, there are many methods can be extended to be adaptive to cross-domain learning. Here, let's take classical fisher discriminant analysis (FDA) [12] to be an extend object owing to the fact that though it is used widely in real-world applications, FDA tends to give undesired results if samples in training and test sets belong to different distributions. The basis objective of FDA is "maximize the class-separate degree". Therefore, the Q_{max} is a combination of two items that estimate the between-class and within-class scatter degree, respectively. Their concrete definitions are:

$$\begin{aligned} S_b &= \sum_{i=1,2} n_i(\mu_i - \mu)(\mu_i - \mu)^\top, \\ S_w &= \sum_{i=1,2}(\sum_{j=1}^{n_i}(x_j^i - \mu_i)(x_j^i - \mu_i)^\top), \end{aligned} \quad (8)$$

where μ is the total sample mean, n_i is the sample number in the ith class, μ_i is the mean vector of ith class, and x_j^i is the jth sample in the ith class.

2.3.2 Domain-Merge and Class-Separate FDA

Upon the description of FDA, we can see the modification object is S_w. Here, we can define an item to estimate the domain-separate degree as (some related work can also see [18]):

$$S_d^U = (\mu^{D_1} - \mu^{D_2})(\mu^{D_1} - \mu^{D_2})^\top, \quad (9)$$

where μ^{D_1} and μ^{D_2} are data means of a domain pair $\{D_1, D_2\}$. Moreover, considering that there may be some labeled sample in the domains, here a more precision item:

$$S_d^L = (\mu_1^{D_1} - \mu_1^{D_2})(\mu_1^{D_1} - \mu_1^{D_2})^\top + (\mu_2^{D_1} - \mu_2^{D_2})(\mu_2^{D_1} - \mu_2^{D_2})^\top, \quad (10)$$

where $\mu_i^{D_1}(i = 1, 2)$ and $\mu_i^{D_2}(i = 1, 2)$ are class means of a domain pair $\{D_1, D_2\}$.

Finally, here a between-domain scatter matrix can be defined as:

$$S_d = S_d^U + (1 + \min(n_{tr}^{D_1}/n_{te}^{D_1} + n_{tr}^{D_2}/n_{te}^{D_2}))S_d^L, \quad (11)$$

where $n_{tr}^{D_i}$ and $n_{te}^{D_i}$ are sample numbers of labeled and unlabeled set in Domain D_i (i = 1,2). The weight $\min(n_{tr}^{D_1}/n_{te}^{D_1} + n_{tr}^{D_2}/n_{te}^{D_2})$ gives more power to the S_d^L since it is more precision while its precision is proportional to the ratio of the labeled example number. Note that the equation (9) is defined with domain pairs. For multiple domains, we can use the sum of S_d of each domain-pair. Finally, the combination of class-separate and domain-merge objective can be achieved by the combination of S_d, S_b and S_w:

$$J(\phi) = \frac{\phi^\top S_b \phi}{\phi^\top (S_w + \alpha S_d)\phi}, \quad (12)$$

where α is a parameter to control the balance between the desired levels of class-separate and domain-merge degree objective. The final model is called as Domain-Merge and

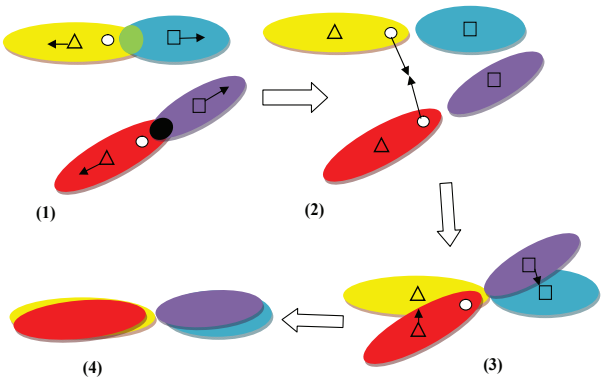

Figure 2: The roles of the items in (10). Yellow and blue regions respectively denote data belongs to class 1 and class 2 in the domain D_1, and the red and purple ones draw the data belongs to class 1 and 2 in the domain D_2. The two classes in both domains are overlap to some extent, as the green and black regions reveal. (1) indicates the traditional class-separate objectives with S_b and S_w. (2) shows the effect of the unsupervised domain-merge objectives with S_d^U in S_d. (3) gives the effect of the supervised domain-merge objectives with S_d^L in S_d. (4) shows the final status.

Class-Separate FDA (DMCS_FDA). Fig.2 gives some visualization explanation of DMCS_FDA.

2.3.3 *Kernel Domain-Merge and Class-Separate FDA*

The algorithm described above is a linear method. However, many real datasets are only of approximate linear structures. Therefore it is important to consider generalizing the DMCS_FDA criterion to cope with the case of nonlinear feature extraction [13]. In this section, we discuss how to perform DMCS_FDA in the reproducing kernel hilbert space (RKHS) by means of the kernel trick, which gives rise to kernel DMCS_FDA.

Here we consider the problem in a feature space induced by some nonlinear mapping Ψ and thus an inner product $(\cdot)$ can be defined on the feature space which makes for an RKHS. Then, we generalize the terms $\phi^\top S^\Psi \phi$ ($S = S_B^{ST}, S_W^{ST}, S_U^{ST}$ and S_L^{ST}) in DMCS_FDA in the RKHS.

From the theory of reproducing kernels, we know that any solution ϕ must lie in the span of samples in $\mathcal{F}$:

$$\phi = \sum_{i=1}^{l} \beta_i \Psi(x_i), \tag{13}$$

where β_i $(i = 1, \cdots, l)$ are scalars.

First, we summarize $\phi^\top S^\Psi \phi$ ($S = S_B^{ST}, S_U^{ST}$ and S_L^{ST}) in a unified form. Denote a mathematical term called between datasets scatter matrix as $S_{12} = (m_1 - m_2)(m_1 - m_2)^\top$, where m_1 and m_2 are data means of two datasets $Set_1 = \{x_1^1, x_2^1, \cdots, x_{l_1}^1\}$ and $Set_2 = \{x_1^2, x_2^2, \cdots, x_{l_2}^2\}$. We show that the term $\phi^\top S_{12} \phi$ can be generalized into the RKHS: $\phi^\top S_{12}^\Psi \phi = \phi^\top (m_1^\Psi - m_2^\Psi)\phi$. Using the expansion (13) and the definition of m_i^Ψ we can see:

$$\begin{aligned} \phi^\top m_i^\Psi &= \frac{1}{l_i} \sum_{j=1}^{l} \sum_{k=1}^{l_i} \beta_j k(x_j, x_k^i) \\ &= \beta M_i, \end{aligned} \tag{14}$$

where M_i is a vector with $(M_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$ and the dot products is replaced by the kernel function.

Then, we get:

$$\phi^\top S_{12}^\Psi \phi = \beta^\top M \beta, \tag{15}$$

where $M = (M_1 - M_2)(M_1 - M_2)^\top$.

Now, we point that $\phi^\top S^\Psi \phi$ ($S = S_b, S_d^U$ and S_d^L can be computed by expansion (15) with different definitions of Set_1 and Set_2: For $\phi^\top S_b^\Psi \phi$, Set_1 and Set_2 should be the samples subsets of training set belong to class 1 and 2, respectively. For $\phi^\top S_d^{U\Psi}$ ϕ, Set_1 and Set_2 should be the unlabeled set from domain D_1 and D_2. For

$$\phi^\top S_d^{L\Psi} \ \phi$$

, it can be computed as $\phi^\top (S_{12}^{1\Psi} + S_{12}^{2\Psi})\phi$, where $S_{12}^{i\Psi}$ is based on Set_1 and Set_2 as the subsets of labeled set from domain D_1 and D_2 belong to class i $(i = 1, 2)$.

After defining $\phi^\top S_b^\Psi \phi$, $\phi^\top S_d^{U\Psi} \phi$, and $\phi^\top S_d^{L\Psi} \phi$, There is still only one term $\phi^\top S_w^\Psi \phi$ need be formulated. Using the expansion (13), the definition of m_i^Ψ and similar analysis mentioned above, we find:

$$\phi^\top S_w^\Psi \phi = \beta^\top N \beta, \tag{16}$$

where $N = \sum_{j=1,2} K_j (I - \mathbf{1}_{l_j}) K_j^\top$ and K_j is a $l \times l_j$ matrix with $(K_j)_{nm} = k(x_n, x_m^j)$ which is the kernel matrix for class j, $\mathbf{I}$ is t he identity and $\mathbf{1}_{l_j}$ is the matrix with all entries $1/l_j$.

Finally, by combining the terms, we can formulate the objective function of DMCS_FDA in RKHS as maximizing:

$$\frac{\phi^\top S_b^\Psi \phi}{\phi^\top S_w^\Psi \phi + \alpha(\phi^\top S_d^{U\Psi} \phi + \phi^\top S_d^{L\Psi} \ \phi)}. \tag{17}$$

2.4 Cross-domain Distance Metric Learning

Distance metric learning [23] aims to learn between-point distance representation. The performance of many machine learning and data mining algorithms depend critically on their being given a good metric over the input space. Distance metric learning aims to learn data-distance representation which can help later tasks. However, for cross-domain learning, the distance function fitted to one domain may be unfitted to another owing to the domain differences. As a result, the distance function is expected to not only maximize the pair distance between points belong to different classes but also make sure that the pairs of points belong to different domains are not far away to each other.

2.5 Pairwise constraints based Distance Metric Learning

Here is the introduction of distance metric learning with pairwise constraints (denoted as PCDML). It is based on two pairwise constraints on the data: equivalence and inequivalence constraints. The equivalence constraints are defined on the samples which should be close together in the

learned metric, while the inequivalence constraints indicate the points that should not be near in the learned metric. In [21] and [1], the distance metric is explicitly learned to minimize the distance between data points within the equivalence constraints and maximize the distance between data points in the inequivalence constraints.

Let $\mathcal{C} = \{x_1, x_2, \ldots, x_n\}$ be a collection of data points, where n is the number of samples in the collection. Each $x_i \in \mathbb{R}^m$ is a data vector where m is the number of features. Let the set of equivalence constraints denoted by

$$\mathcal{S} = \{(x_i, x_j | x_i \; and \; x_j \; belong \; to \; the \; different \; class)\} \tag{18}$$

and the set of inequivalence constraints denoted by

$$\mathcal{D} = \{(x_i, x_j | x_i \; and \; x_j \; belong \; to \; the \; same \; classes)\} \tag{19}$$

Let the distance metric denoted by matrix $\mathbf{A} \in \mathbf{R}^{m \times n}$, and the distance between any two data points x and y expressed by

$$d^2_{\mathbf{A}} = ||x - y||^2_{\mathbf{A}} = (x - y)^T \mathbf{A}(x - y) \tag{20}$$

Given the equivalence constraints in $\mathcal{S}$ and the inequivalence constraints in $\mathcal{D}$, [11] formulated the problem of metric learning into the following convex programming problem [14]:

$$\begin{aligned} &\min_{\mathbf{A} \in \mathbf{R}^{m \times n}} \sum_{(x_i, x_j) \in \mathcal{S}} ||x - y||^2_{\mathbf{A}} \\ &s.t. \quad \mathbf{A} \succeq 0, \sum_{(x_i, x_j) \in \mathcal{D}} ||x - y||^2_{\mathbf{A}} \geq 1 \end{aligned} \tag{21}$$

Note that the positive semi-definitive constraint $\mathbf{A} \succeq 0$ is needed to ensure the negative distance between any two data points and the triangle inequality. There are many optimization tools to solve this problem.

2.6 Domain-Merge and Class-Separate PCDML

In traditional PCDML method, equivalence and inequivalence constraints only aim to maximize the class-separate degree by reducing the distance between point pairs that belongs to the same class and enlarging the distance between ones that have different class labels. For multi-domain learning, the objective function should consider not only the distance between classes but also the differences between domains. Upon the above discussion, the set $\mathcal{S}$ is used to control the pair distance between points belong to the same class. Here, we modify its definition to control the pair distance between points belong to different domains to obtain Domain-Merge and Class-Separate PCDML (DMCS_PCDML). After the term to control the data presentation between points belong to different domains, there are three items in DMCS_PCDML:

$$\begin{aligned} &\mathcal{S}_{cs} = \{(x_i, x_j | x_i \; and \; x_j \; belong \; to \; the \; same \; domain \; and \\ &same \; class)\} \\ &\mathcal{S}_{dm} = \{(x_i, x_j | x_i \; and \; x_j \; belong \; to \; the \; different \; domains)\} \\ &\mathcal{D}_{cs} = \{(x_i, x_j | x_i \; and \; x_j \; belong \; to \; the \; same \; domain \; and \\ &different \; classes)\} \end{aligned} \tag{22}$$

The $\mathcal{S}_{cs}$ and $\mathcal{D}_{cs}$ are items for class-separate objective and $\mathcal{S}_{dm}$ for domain-merge objective. By combing them, we could obtain a cross-domain distance metric learning that can control the balance between the desired levels of two objectives with a parameter α:3

$$\begin{aligned} &\min_{\mathbf{A} \in \mathbf{R}^{m \times n}} \sum_{(x_i, x_j) \in \mathcal{S}_{cs}} ||x - y||^2_{\mathbf{A}} + \; \alpha \times \sum_{(x_i, x_j) \in \mathcal{S}_{dm}} ||x - y||^2_{\mathbf{A}} \\ &s.t. \quad \mathbf{A} \succeq 0, \sum_{(x_i, x_j) \in \mathcal{D}_{cs}} ||x - y||^2_{\mathbf{A}} \geq 1 \end{aligned} \tag{23}$$

With the new optimization function, we obtain Domain-Merge and Class-Separate PCDML (DMCS_PCDML).

2.7 Target-Priority Cross-validation for Cross-domain Parameter Selection

Representation learning models (as well as a majority of machine learning methods) always have one or more parameters to be determined, which is just like the parameter α in our methods. K-fold cross-validation technology [9] is a very popular method used to determined the parameters. However, in cross-domain learning, there are often one target domain and several source domains. The target domain often only has very few labeled data. Source domains has relatively more labeled examples but they are underlying different distributions with the target domain. In this setting, the number of target labeled examples in the traditional validation set in traditional K-fold cross-validation technology will be greatly larger than the number of source labeled examples. Therefore, the score of a candidate parameter will nearly just rely the performance on source datasets.

Here, we propose a "target-priority" strategy to modify the parameter selection step in the previous cross-validation technology when it is used to perform parameter selection in transfer learning scenarios. Here suppose we wish to determine parameter α. We firstly choose a set of α-values corresponding to the best performance on target samples (owning to the small number of target samples, there are always a lot of α-values corresponding to the best performance on target samples). Then, in that set, we choose the α-value corresponding to the best performance on source samples as the final α to be used in later tasks. This strategy gives a prior consideration to the performance on target samples because these samples are drawn from the same distribution as test samples (as Algorithm3 indicates). As a result, this parameter selection method adapts to transfer learning problems and is used to determine α in our later experiments with K is 10 and the candidate α-value set is $[0.1, 0.15, 0.2, 0.25, \cdots, 1]$.

3. EXPERIMENTS

We evaluate our algorithms together with corresponding traditional algorithms on the "Amazon reviews" benchmark data sets [2]. In the original dataset, each review is associated with a rating of 1-5 stars. For simplicity, we are only concerned about whether or not a review is positive (higher than 3 stars) or negative (3 stars or lower). That is, $y_i = +1, -1$, where $y_i = 1$ indicates that it is a positive review, and -1 otherwise. To simulate the cross-domain

Algorithm3 Target-Priority Cross-validation

Input:
labeled dataset X^S from source domains
labeled dataset X^T from the target domain
Parameter candidates $\alpha_{set} = [\alpha_1, \alpha_2, \cdots, \alpha_n]$
Number of folds: K

Output:
a special $\alpha \in \alpha_{set}$

Step 1:
Divide X^S into K folds: $X_1^S, \cdots, X_K^S$
Divide X^T into K folds: $X_1^T, \cdots, X_K^T$

Step 2:
For $i = 1, 2, \cdots, n$
  For $k = 1, 2, \cdots, K$
    Set X_k^S, X_k^T as source and target validation sets, respectively.
    Set $T_k = ..., X_j^S, X_j^T, ...$, where $j = 1, k-1, \cdots, k+1, K$ as the training set.
    Calculate the performance of α_i corresponding to X_k^T and T_k as $Perf_k^T(\alpha_i)$
    Calculate the performance of α_i corresponding to X_k^S and T_k as $Perf_k^S(\alpha_i)$
  End
End

Step 3:
For $i = 1, 2, \cdots, n$
  Set target performance of α_i as
  $Perf^T(\alpha_i) = \sum_{k=1}^{K} \frac{1}{K} Perf_k^T(\alpha_i)$.
  Set source performance of α_i as
  $Perf^S(\alpha_i) = \sum_{k=1}^{K} \frac{1}{K} Perf_k^S(\alpha_i)$.
End

Step 4:
Step 4.1:
Select the subset α_{set}^T as the set of α values corresponding to the highest $Perf^T(\alpha_i)$ (for all α_i values in α_{set}).
Step 4.2:
Select the α value in α_{set}^T corresponding to the highest $Perf^S(\alpha_j)$ (for all α_j values in α_{set}^T) as the α value we want.

Table 2: The classification accuracies (%) of our methods and traditional approaches

Target	Accuracy: Means ± Std	
	DMCS_CSF	CSF
books	**76.11±2.1**	72.49±2.5
dvd	**74.93±1.8**	71.22±1.4
electronics	**74.02±2.4**	71.14±3.2
kitchen and housewares	**82.93±3.4**	78.78±3.5
Target	Accuracy: Means ± Std	
	DMCS_FDA	FDA
books	**76.62±2.5**	73.23±1.9
dvd	73.22±1.9	**74.17±1.3**
electronics	**72.13±1.4**	73.28±2.5
kitchen and housewares	**81.34±2.9**	78.05±3.0
Target	Accuracy: Means ± Std	
	DMCS_PCDML	PCDML
books	**74.56±2.3**	73.21±1.8
dvd	73.12±1.9	**73.19±1.6**
electronics	73.58±1.9	**74.48±2.5**
kitchen and housewares	**78.12±2.9**	75.28±2.3

learning, the data from four results corresponding to different target domains (e.g. target: books, sources:). In each experiment, the number of labeled examples of the target domain is 50, and the number of labeled examples of each source domain is 500, the unlabeled example number of the target domain is 450 and the test is 500.

The original feature space of unigrams and bigrams is on average approximately 120, 000 dimensions across different domains. To reduce the dimensionality, we only use features that appear at least 10 times in a particular domain adaptation task (with approximately 40; 000 features remaining). Further, we pre-process the data set with standard tf-idf [15] feature re-weighting. Moreover, the dimensionality of the new data-representation is 4,000. Two classifiers (kNN [16] with $k = 5$, SVM [3] with C = 1 and polynomial kernel) are employed to perform classification tasks in new data spaces and the average results are presented. Moreover, since training and test are both selected randomly, the experiment is repeated 10 times and uses "means±std" as final results.

Table 2 compares the performances of our methods and corresponding traditional methods to valid the effectiveness of our framework. We can see that, in most cases, our methods gain better results. Moreover, it is found that the DMCS_CSF and DMCS_FDA have larger improvement than DMCS_PCDML.

4. CONCLUSIONS

This paper proposes propose a framework to combine class-separate and domain-merge objective to achieve cross-domain representation learning. The final data representation is expected to maximize the discriminant among classes and minimize the differences among domains. Moreover, in this paper, the domain-merge objective is defined upon the modification with traditional class-separate objective, so that the generated methods can be implemented with current toolkits. Three special methods are proposed with this framework

and they belong to feature selection, feature reconstruction and distance metric learning categories.

There are still some important issues worth researching. For example, the theoretical analysis and validation of lower error rate of combination of class-separate and domain-merge objectives in cross-domain learning is a big challenge. Moreover, improving exist incremental algorithms with our framework is promising to be contributed to on-line cross-learning [6]. Last, cross-domain research can help many other real-world applications such as bioinformatics [19] and computer vision [24], and employing our framework on improving research on them is interesting.

5. ACKNOWLEDGMENTS

his work is supported in part by the National Natural Science Foundation of China under Project 61075005, and the Fundamental Research Funds for the Central Universities.

6. ADDITIONAL AUTHORS

7. REFERENCES

[1] Bar-Hillel, A., Hertz, T., Shental, N., and Weinshall, D. Learning distance functions using equivalence relations. In *machine learning international workshop* (2003), vol. 20, p. 11.

[2] Blitzer, J., McDonald, R., and Pereira, F. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (2006), Association for Computational Linguistics, pp. 120–128.

[3] Cortes, C. Support-vector networks. *Machine Learning 20*, 3 (1995), 273–297.

[4] Dash, M., and Liu, H. Feature selection for classification. *Intelligent data analysis 1*, 1-4 (1997), 131–156.

[5] Daumé III, H., and Marcu, D. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research 26*, 1 (2006), 101–126.

[6] Dredze, M., and Crammer, K. Online methods for multi-domain learning and adaptation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), Association for Computational Linguistics, pp. 689–697.

[7] Hall, M. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[8] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter 11*, 1 (2009), 10–18.

[9] Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence* (1995), vol. 14, LAWRENCE ERLBAUM ASSOCIATES LTD, pp. 1137–1145.

[10] Lafon, S., and Lee, A. 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 (2006), 1393–1403.

[11] Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. *Advances in neural information processing systems 21* (2009), 1041–1048.

[12] Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop* (1999), IEEE, pp. 41–48.

[13] Muller, K., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. An introduction to kernel-based learning algorithms. *Neural Networks, IEEE Transactions on 12*, 2 (2001), 181–201.

[14] Nguyen, H., Franke, K., and Petrovic, S. Optimizing a class of feature selection measures. In *NIPS 2009 Workshop on Discrete Optimization in Machine Learning: Submodularity, Sparsity & Polyhedra (DISCML), Vancouver, Canada* (2009).

[15] Salton, G., and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information processing & management 24*, 5 (1988), 513–523.

[16] Shakhna-rovich, G., Darrell, T., and Indyk, P. Nearest-neighbor methods in learning and vision. *IEEE Transactions on Neural Networks 19*, 2 (2008), 377.

[17] Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., Bona, F., Binder, A., Gehl, C., and Franc, V. The shogun machine learning toolbox. *The Journal of Machine Learning Research 11* (2010), 1799–1802.

[18] Tu, W., and Sun, S. Transferable discriminative dimensionality reduction. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on* (2011), IEEE, pp. 865–868.

[19] Tu, W., and Sun, S. Subject transfer framework for eeg classification. *Neurocomputing 82* (2012), 109–116.

[20] Van Der Maaten, L., Postma, E., and Van Den Herik, H. Dimensionality reduction: A comparative review. *Published online 10*, February (2007), 1–35.

[21] Xing, E., Ng, A., Jordan, M., and Russell, S. Distance metric learning, with application to clustering with side-information. *Advances in neural information processing systems 15* (2002), 505–512.

[22] Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., and Lin, S. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 1 (2007), 40–51.

[23] Yang, L., and Jin, R. Distance metric learning: A comprehensive survey. *Michigan State Universiy* (2006), 1–51.

[24] Yuan, X., and Yan, S. Visual classification with multi-task joint sparse representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (2010), IEEE, pp. 3493–3500.