Domain Transfer Multi-Instance Dictionary Learning

Ke Wang \cdot Jiayong Liu \cdot Daniel González

Received: date / Accepted: date

Abstract In this paper, we invest the domain transfer learning problem with multi-instance data. We assume we already have a well-trained multi-instance dictionary and its corresponding classifier from the source domain, which can be used to represent and classify the bags. But it cannot be directly used to the target domain. Thus we propose to adapt them to the target domain by adding an adaptive term to the source domain classifier. The adaptive function is a linear function based a domain transfer multi-instance dictionary. Given a target domain bag, we first map it to a bag-level feature space using the domain transfer dictionary, and then apply a the linear adaptive function to its baglevel feature vector. To learn the domain-transfer dictionary and the adaptive function parameter, we simultaneously minimize the average classification error of the target domain classifier over the target domain training set, and the complexities of both the adaptive function parameter and the domain transfer dictionary. The minimization problem is solved by an iterative algorithm which update the dictionary and the function parameter alternately. Experiments over several benchmark data sets show the advantage of the proposed method over existing state-of-the-art domain transfer multi-instance learning methods.

Ke Wang College of Mathematics, Sichuan University, Chengdu 610064, China

Jiayong Liu* College of Electronics and Information Engineering, Sichuan University, Chengdu 610064, China E-mail: jiayongliuscu@gmail.com * Corresponding author

Daniel González Computer Science Department, Catholic University of Murcia, Murcia 30107, Spain E-mail: danielgonzalezmurcia@gmail.com Keywords Multi-instance Learning \cdot Domain Transfer learning \cdot Classifier Adaptation \cdot Gradient Descent

1 Introduction

Machine learning usually assumes that the training data are from the same domain [29,9,10,8,7,11,30,32,21,31,15]. Recently, transfer learning has been a well-studied topic in machine learning community. It refers to the problem of learning a machine learning modeling for a target domain with help from a source domain [34,24,25,39,41]. For a machine learning problem, a source domain is a domain with sufficient training data, which are well labeled, and thus it is easy to train models in this domain. A target domain is usually in lack of training data, or in lack of labels. Because of the lack of data and/or label, the training in the target domain is difficult, and has the problem of over-fitting. The source domain and target domains share the same feature space and label space, thus it would be very helpful to use the source domain data to help the learning problem in the target domain. However, the data distributions of the source and target domains are usually very different, if we simply use the models trained by using the source domain training set, the prediction performance over the target domain can be very inferior. The reason for this is that the difference between the two different domain distributions can even be more significant than the difference between different classes. To solve this problem, we need to transfer the source domain model to the target domain by adapting the source domain predictor to the target domain data set. This is called domain transfer learning. Given a source domain predictor, its train a target domain predictor by adapting it to the target domain over the target domain training set, even when the target domain training set itself is insufficient to train a new target domain predictor. One example of domain transfer learning is spoken Arabic digit recognition [13, 14]. In a spoken Arabic digit data set, there are two domains of data, which are male voice signal, and female voice signal. For each domain, there are voice data of ten classes, and each class is a one digit, varying from one to ten. Although the voice signal of both domains shares the same feature space and class label space, however, the classifier trained with male domain data cannot be used directly in the female voice signal domain, due to the significant difference between male and female voice signal domains. Thus an adaptation is necessary to transfer the classifier for the male domain classifier to the female domain. Another example of transfer learning is spam detection problem [5]. For one email user, we can train a spam detector from the well labeled spam and normal emails. However, it cannot be used to detect the spams of another user because the emails of these two users may are significantly different from each other. Thus to use the first user's detector to the second user's email, we need to perform an adaptation to the detector.

Up to now, most of the domain transfer learning methods are focused on single instance data [6,5,16,38,27]. That is, one data point is only presented

as one single instance. However, in many machine learning applications, one data point can contains multiple instances, and is presented as a bag of instances. This type of data is called multi-instance data. In domain transfer learning problem, many applications require to learn domain transfer classifier for multi-instance data. However, only a few works are done to this direction, while most existing works ignore the nature of multi-instance data and only treat them as one single feature vector. To overcome this problem, two methods have been proposed for domain transfer learning with multi-instance data.

- Zhang and Si [40] proposed the problem of domain transfer learning for multi-instance data, and a novel method to solve it within the framework of multi-task learning. More than one source domains are considered and each domain is considered as a task. The target domain classifier is obtained by a weighted linear combination of the classifiers of the multiple tasks.
- Wang et al. [35] proposed an adaptive knowledge transfer learning framework for multi-instance data. This method adapts the source-target domain cross class knowledge to the target domain for the multi-instance data to boost the learning. It also build a data-dependent mixture model to combine the knowledge from the source domain to enhance the target domain weak classifier. The objective of this model is optimized by an iterative coordinate descent method as a constraint concave-convex programming problem.

It has been shown that the most effective way for multi-instance learning method is using multi-instance dictionary to map a bag of instances to a bag-level feature vector [3,12]. This representation method is based on bag-instance similarity, and the critical component of this method is the learning of the dictionary. In this paper, we invest the problem of domain transfer multi-instance learning problem and propose a novel solution based on the dictionary. According to our knowledge, this is the first work toward the direction of learning multi-instance dictionary in the scene of domain transfer learning.

We assume that in the source domain, we have a well-trained multi-instance dictionary and a corresponding classifier. Using this dictionary, we can map a bag of instances to a bag-level feature by using the bag-instance similarity, and then in the bag-level feature space, we can apply the classifier to classify the bag. However, the source domain dictionary and classifier cannot be directly applied to the target domain, we propose to add an adaptive term to the source domain classifier to construct the target domain classifier to transfer the knowledge of the source domain to the target domain. To construct the domain adaptive term, we propose to learn a domain transfer dictionary to represent a target domain bag to the bag-level feature space, and the adaptive term is designed to be a linear function in this space. The problem is transferred to the problem of learning the domain transfer dictionary and adaptive function parameter. To this end, we propose to optimize both the mentioned parameters by learn from a limited target domain training set. We propose to minimize the hinge loss of the target domain training bags, the complexity of the adaptive function, and the complexity of the domain transfer dictionary. To solve the minimization problem, we develop an iterative algorithm. The contributions of this paper are listed as follows.

- Contribution #1. We propose a novel problem of transfer domain dictionary learning. This problem is beyond simple domain transfer learning or dictionary learning. It is proposed to explore the cross-domain knowledge in the aspect of dictionary learning. Its significance is to learn the critical dictionary which can bridge the source and target domains. It only learns a good predictor in the target domain with help of source domain, but also reveals the nature of the multi-instance data which connects the to domains.
- Contribution #2. We build a novel learning model to solve this problem. This model based on a joint optimization problem of both the domain transfer dictionary and the domain adaptation function. The objective is composed of three terms. The first term is the average classification error over the target domain training bags, measured by the hinge loss, the second term is the complexity term of the adaptation function, and the last term is the complexity term of the domain transfer dictionary. In this way, our problem is modeled as a minimization problem.
- Contribution #3. We also develop an effective algorithm to solve the problem proposed in Contribution #2. We use the Lagrange multiplier method and an alternate optimization strategy. The optimization of the adaptation function parameter is transferred to the optimization of some Lagrange multipliers, and the optimization of domain transfer dictionary and the Lagrange multipliers are conducted alternately. The optimization of Lagrange multipliers is performed as a quadratic programming problem, while the domain transfer dictionary is updated by gradient descent algorithm.

This paper is organized in the following way. In section 2, we introduce the proposed domain transfer dictionary learning method, in section 3, we give the experimental results of the proposed method over several benchmark data sets, and finally in section 4, we conclude the paper with some future works.

2 Learning Domain Transfer Dictionary

In this section, we introduce the newly proposed multi-instance dictionary learning method for domain transfer learning problem. In section 2.1, we model the learning problem as a minimization problem, and the problem is solved in section 2.2. Furthermore, in section 2.3, we propose an iterative algorithm to implement the solution of the problem.

2.1 Problem modeling

In the problem of domain transfer multi-instance learning, we suppose we have two domains, which are a source domain and a target domain. For the source domain, we have a training set, and we have learned a well-trained multiinstance dictionary, $\Phi = \{\phi_k\}_{k=1}^t$, where $\phi_k \in \mathbb{R}^d$ is the vector of the k-th word of the dictionary, and ι is the number of the words in this dictionary. With this dictionary, we can represent a bag of instances as a feature vector of ι dimensions. Suppose we have a bag denoted as $B = \{\mathbf{x}_j\}_{j=1}^m$, where \mathbf{x}_j is *d*-dimensional feature vector the *j*-th instance of the bag, and *m* is the number of the instances of this bag, we can represent as

$$B \to \mathbf{z}_B^{\boldsymbol{\Phi}} = \left[\max_{j=1}^m \boldsymbol{\phi}_1^\top \mathbf{x}_j, \cdots, \max_{j=1}^m \boldsymbol{\phi}_{\iota}^\top \mathbf{x}_j \right]^\top \in \mathbb{R}^{\iota}$$
(1)

where the k-th dimension is the maximum dot-produce between the k-th codeword and the instances of B. The motive to use the maximum dot-product to measure the similarly between an instance and a bag is it is simple and parameter-free. Some other similarity measures such as Gaussian kernel requires additional parameters such as the band-width parameter. The tuning of these parameters is time-consuming and has the problem of over-fitting. The chosen similarity maximum dot-product does not have such problems because it is parameter-free. With this bag-level feature vector, we can classify the bags in the bag-level space. To this end, we also have a well-trained classifier, f(B), which is learned from the source domain, to map the bag B to its true binary class label, $y \in \{+1, -1\}$, from its bag-level features,

$$y \leftarrow f(B; \Phi, \mathbf{v}) = \mathbf{v}^\top \mathbf{z}_B^{\Phi},\tag{2}$$

where $\mathbf{v} \in \mathbb{R}^{\iota}$ is the parameter of the source domain classifier. The source domain dictionary, Φ , and its corresponding bag-level classifier, \mathbf{v} are trained over the source domain, and are supposed work well in the source domain. Moreover, we have a target domain, and we also want to present and classify the bags of the target domain. One direct method is to apply the source domain dictionary and classifier to the target domain data. However, this is not suitable due to the significant difference between the distributions of the source domain and target domain. We propose to design the target domain multi-instance classifier, g, by adapting the source domain dictionary classifier, f, to the target domain. More specifically, the target domain classifier is the combination of the source domain classifier and an adaptation term, Δ ,

$$g(B;\Psi,\mathbf{w}) = f(B;\Phi,\mathbf{v}) + \Delta(B;\Psi,\mathbf{w})$$
(3)

where Ψ and \mathbf{w} are parameters of the adaptation term which will be specified as follows. Ψ is the domain transfer multi-instance dictionary, and it contains κ codewords, $\Psi = \{\psi_k\}_{k=1}^{\kappa}$, where $\psi_k \in \mathbb{R}^d$ is its k-th codeword. We use it to represent a bag B to a κ -dimensional bag-level feature vector,

$$B \to \mathbf{z}_B^{\Psi} = \left[\max_{j=1}^m \boldsymbol{\psi}_1^\top \mathbf{x}_j, \cdots, \max_{j=1}^m \boldsymbol{\psi}_\kappa^\top \mathbf{x}_j \right]^\top \in \mathbb{R}^{\kappa}.$$
(4)

The adaptation term is a linear function in this bag-level feature space constructed by the domain transfer dictionary, Ψ ,

$$\Delta(B; \Psi, \mathbf{w}) = \mathbf{w}^{\top} \mathbf{z}_{B}^{\Psi} \tag{5}$$

 $\mathbf{w} = [w_1, \cdots, w_{\kappa}] \in \mathbb{R}^{\kappa}$ is the κ -dimensional adaptation function parameter vector, and w_k is its k-th element. Substituting the definition of the adaptation term Δ to (8), we can rewrite the target domain classifier g as follows,

$$g(B; \Psi, \mathbf{w}) = f(B; \Phi, \mathbf{v}) + \mathbf{w}^{\top} \mathbf{z}_{B}^{\Psi}, \tag{6}$$

and the problem of domain transfer multi-instance dictionary learning is to learn both Ψ and \mathbf{w} by using a target domain training set. The target domain training set is denoted as $X = \{(B_i, y_i)\}_{i=1}^n$, where B_i is the bag of the *i*th target domain training data point, and $y_i \in \{+1, -1\}$ is the binary class label of the *i*-th target domain training data point. B_i contains m_i instances, $B_i = \{\mathbf{x}_j^i\}_{j=1}^{m_i}$, where $\mathbf{x}_j^i \in \mathbb{R}^d$ is the *j*-th instance of the *i*-th bag. Given a dictionary Ψ , without confusion, the bag-level vector of B_i , $\mathbf{z}_{B_i}^{\Psi}$ is simply denoted as \mathbf{z}_i^{Ψ} . Moreover, the classification response of the source domain classifier over this bag is denoted as

$$f_i = f(B_i; \Phi, \mathbf{v}) = \mathbf{v}^\top \mathbf{z}_i^{\Phi}.$$
 (7)

In this way, the response of the target domain classifier over B_i is given as

$$g(B_i; \Psi, \mathbf{w}) = f_i + \mathbf{w}^\top \mathbf{z}_i^{\Psi}, \tag{8}$$

To learn both the domain transfer dictionary Ψ and the classifier parameter **w** over the target domain training set, we consider the following three minimization problems.

 Minimization of the classification errors. The classification error of the *i*-th training bag is measure by the hinge loss,

1

$$E(g(B_i; \Psi, \mathbf{w}), y_i) = \max\left(0, 1 - y_i g(B_i; \Psi, \mathbf{w})\right)$$

= max $\left(0, 1 - y_i \left(f_i + \mathbf{w}^\top \mathbf{z}_i^{\Psi}\right)\right)$ (9)

To seek the optimal classifier and dictionary, this classification error term should be minimized. We propose to minimize the average hinge loss over the target domain training set,

$$\min_{\Psi, \mathbf{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} E(g(B_i; \Psi, \mathbf{w}), y_i) = \frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i \left(f_i + \mathbf{w}^\top \mathbf{z}_i^{\Psi}\right)\right) \right\}.$$
(10)

The minimization of this problem is difficult because it is couple with a maximization problem within the hinge loss function. To solve this problem, we introduce a slack variable ξ_i to present the maximum variable between 0 and $1 - y_i \left(f_i + \mathbf{w}^\top \mathbf{z}_i^{\Psi} \right)$,

$$\xi_i \ge 0, \xi_i \ge 1 - y_i \left(f_i + \mathbf{w}^\top \mathbf{z}_i^{\varPsi} \right). \tag{11}$$

With this slack variable, we can rewrite the problem in (10) as a constrained minimization problem as follows,

$$\min_{\Psi, \mathbf{w}, \xi_i \mid_{i=1}^n} \frac{1}{n} \sum_{i=1}^n \xi_i,$$

$$s.t. \ \xi_i \ge 0, \xi_i \ge 1 - y_i \left(f_i + \mathbf{w}^\top \mathbf{z}_i^{\Psi} \right), \forall \ i = 1, \cdots, n.$$
(12)

- Reducing the complexity of the adaptation function. To prevent the over-fitting problem, we want to keep the adaptation function as simply as possible, and reduce the complexity of the adaptation function. The complexity of the adaptation function is measured by the squared ℓ_2 norm of the adaptation function parameter,

$$R(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_2^2 = \frac{1}{2} \sum_{k=1}^{\kappa} w_k^2.$$
 (13)

To this end, we propose to minimize this regularization term as follows,

$$\min_{\mathbf{w}} \left\{ R(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_2^2 \right\}$$
(14)

The solution of this single problem is an all-zero vector $\mathbf{w} = [0, \dots, 0]$. This solution is not optimal for the overall problem, but when this minimization problem is combined with the other problems, this term can bring a tradeoff between the classification error and the adaptation function simplicity.

- Reducing the complexity of the dictionary. We also hope the dictionary can be as simple as possible. The complexity is also measure by the squared ℓ_2 norms of the codewords,

$$\min_{\Psi} \left\{ Q(\Psi) = \frac{1}{2} \sum_{k=1}^{\kappa} \|\psi_k\|_2^2 \right\}$$
(15)

Similarly, the solution for this single problem is κ all-zero codewords, $\psi_k = [0, \dots, 0], k = 1, \dots, \kappa$. So we also use it as a regularization term to tradeoff the learning of the dictionary.

The overall minimization problem is the weighted linear combination of the problems in (16), (14), and (15).

$$\min_{\Psi, \mathbf{w}, \xi_{i=1}^{n}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \xi_{i} + C_{1} R(\mathbf{w}) + C_{2} Q(\Psi) \\
= \frac{1}{n} \sum_{i=1}^{n} \xi_{i} + \frac{C_{1}}{2} \|\mathbf{w}\|_{2}^{2} + \frac{C_{2}}{2} \sum_{k=1}^{\kappa} \|\psi_{k}\|_{2}^{2} \right\}$$

$$s.t. \ \xi_{i} \ge 0, \xi_{i} \ge 1 - y_{i} \left(f_{i} + \mathbf{w}^{\top} \mathbf{z}_{i}^{\Psi} \right), \forall \ i = 1, \cdots, n,$$

$$(16)$$

where C_1 is the weight of the regularization term of \mathbf{w} , and C_2 is the weight of the regularization term of Ψ . Please note that in the objective function, the first term is the classification error term of the target domain, which is critical for the learning of optimal domain transfer multi-instance dictionary. This term grantees that the learned dictionary and its corresponding classifier can lead to a good classification accuracy over the training set of the target domain. The second and third terms are regularization terms which try to generalize the learned dictionary and classifier to the data beyond the training set.

2.2 Problem solving

In this section, we discuss how to solve the overall problem in (16). We use the Lagrange multiplier method to solve this problem. We first define a Lagrange multiplier variable $\alpha_i \geq 0$ for each constraint $\xi_i \geq 0$, and a Lagrange multiplier variable $\beta_i \geq 0$ for each constraint $\xi_i \geq 1 - y_i \left(f_i + \mathbf{w}^\top \mathbf{z}_i^{\Psi}\right)$. The Lagrange function of this constrained problem is,

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \xi_{i} + \frac{C_{1}}{2} \|\mathbf{w}\|_{2}^{2} + \frac{C_{2}}{2} \sum_{k=1}^{\kappa} \|\psi_{k}\|_{2}^{2} - \sum_{i=1}^{n} \alpha_{i}\xi_{i} - \sum_{i=1}^{n} \beta_{i} \left(\xi_{i} - 1 + y_{i} \left(f_{i} + \mathbf{w}^{\top} \mathbf{z}_{i}^{\Psi}\right)\right).$$
(17)

Thus the dual form of the problem is given as

$$\max_{\substack{\alpha_{i}|_{i=1}^{n},\beta_{i}|_{i=1}^{n} \quad \Psi, \mathbf{w}, \xi_{i=1}^{n}}} \min_{\substack{\Psi, \mathbf{w}, \xi_{i=1}^{n}}} \left\{ \mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \xi_{i} + \frac{C_{1}}{2} \|\mathbf{w}\|_{2}^{2} + \frac{C_{2}}{2} \sum_{k=1}^{\kappa} \|\psi_{k}\|_{2}^{2} - \sum_{i=1}^{n} \alpha_{i}\xi_{i} - \sum_{i=1}^{n} \beta_{i} \left(\xi_{i} - 1 + y_{i} \left(f_{i} + \mathbf{w}^{\top} \mathbf{z}_{i}^{\Psi}\right)\right)\right\}$$

$$s.t. \ \alpha_{i} \ge 0, \beta_{i} \ge 0, \forall \ i = 1, \cdots, n.$$
(18)

To solve this problem, we set the partial derivatives of \mathcal{L} with regard to \mathbf{w}, ξ_i to zeros,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = C_1 \mathbf{w} - \sum_{i=1}^n \beta_i y_i \mathbf{z}_i^{\Psi} = 0,$$

$$\Rightarrow \mathbf{w} = \frac{1}{C_1} \sum_{i=1}^n \beta_i y_i \mathbf{z}_i^{\Psi},$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = \frac{1}{n} - \alpha_i - \beta_i = 0,$$

$$\Rightarrow \frac{1}{n} - \beta_i = \alpha_i \ge 0,$$

$$\Rightarrow \frac{1}{n} \ge \beta_i.$$
(19)

We substitute the results of (19) to (18), and obtain the following problem,

$$\max_{\beta_{i}|_{i=1}^{n}} \min_{\Psi} \left\{ \mathcal{L} = \frac{C_{1}}{2} \|\mathbf{w}\|_{2}^{2} + \frac{C_{2}}{2} \sum_{k=1}^{\kappa} \|\psi_{k}\|_{2}^{2} + \sum_{i=1}^{n} \beta_{i} \left(1 - y_{i} f_{i}\right) - \sum_{i=1}^{n} \beta_{i} y_{i} \mathbf{w}^{\top} \mathbf{z}_{i}^{\Psi} \right. \\ \left. \left. = \frac{C_{2}}{2} \sum_{k=1}^{\kappa} \|\psi_{k}\|_{2}^{2} + \sum_{i=1}^{n} \beta_{i} \left(1 - y_{i} f_{i}\right) - \frac{1}{2C_{1}} \sum_{i,j=1}^{n} \beta_{i} \beta_{j} y_{i} y_{j} \mathbf{z}_{i}^{\Psi^{\top}} \mathbf{z}_{j}^{\Psi} \right\} \\ s.t. \frac{1}{n} \ge \beta_{i} \ge 0, \forall i = 1, \cdots, n.$$

$$(20)$$

To solve this problem, we use an iterative algorithm with alternate optimization strategy. We only consider two variables in (20), which are $\beta_i|_{i=1}^n$ and Ψ , and all other variables have vanished. To optimize them, in an iteration, we first fix Ψ and update $\beta_i|_{i=1}^n$, then with the updated $\beta_i|_{i=1}^n$, we update Ψ .

2.2.1 Updating $\beta_i|_{i=1}^n$

When Ψ is fixed, the problem in (20) is reduced to

$$\max_{\substack{\beta_{i}|_{i=1}^{n}\\ s.t. }} \left\{ \frac{C_{2}}{2} \sum_{k=1}^{\kappa} \|\psi_{k}\|_{2}^{2} + \sum_{i=1}^{n} \beta_{i} \left(1 - y_{i} f_{i}\right) - \frac{1}{2C_{1}} \sum_{i,j=1}^{n} \beta_{i} \beta_{j} y_{i} y_{j} \mathbf{z}_{i}^{\boldsymbol{\psi}^{\top}} \mathbf{z}_{j}^{\boldsymbol{\psi}} \right\}$$
(21)
s.t. $\frac{1}{n} \ge \beta_{i} \ge 0, \forall i = 1, \cdots, n.$

This is a maximization problem with regard to $\beta_i|_{i=1}^n$. The objective is a quadratic function and the constraints are linear functions. Thus it is a linear

constrained quadratic programming problem. We can solve it by active set algorithm.

2.2.2 Updating Ψ

When $\beta_i|_{i=1}^n$ is fixed, the problem in (20) is reduced to

$$\min_{\Psi} \left\{ \frac{C_2}{2} \sum_{k=1}^{\kappa} \|\psi_k\|_2^2 + \sum_{i=1}^{n} \beta_i \left(1 - y_i f_i\right) - \frac{1}{2C_1} \sum_{i,j=1}^{n} \beta_i \beta_j y_i y_j \mathbf{z}_i^{\Psi^{\top}} \mathbf{z}_j^{\Psi} \right\}.$$
(22)

We further rewrite the bag-level feature vector as

$$\mathbf{z}_{i}^{\Psi} = \begin{bmatrix} m_{i} \\ \max_{j=1}^{m_{i}} \boldsymbol{\psi}_{1}^{\top} \mathbf{x}_{j}^{i}, \cdots, \max_{j=1}^{m_{i}} \boldsymbol{\psi}_{\kappa}^{\top} \mathbf{x}_{j}^{i} \end{bmatrix}^{\top} \\ = \begin{bmatrix} \boldsymbol{\psi}_{1}^{\top} \mathbf{x}_{\pi_{1}^{i}}^{i}, \cdots, \boldsymbol{\psi}_{\kappa}^{\top} \mathbf{x}_{\pi_{\kappa}^{i}}^{i} \end{bmatrix}^{\top},$$

$$(23)$$

where

$$\pi_k^i = \arg\max_{j=1}^{m_i} \boldsymbol{\psi}_k^\top \mathbf{x}_j^i \tag{24}$$

is the index of the instance which gives the maximum product between ψ_k and the instance. Thus we can rewrite $\mathbf{z}_i^{\Psi^{\top}} \mathbf{z}_j^{\Psi}$ in (20) as follows,

$$\mathbf{z}_{i}^{\boldsymbol{\psi}^{\top}} \mathbf{z}_{j}^{\boldsymbol{\psi}} = \sum_{k=1}^{\kappa} \left(\boldsymbol{\psi}_{k}^{\top} \mathbf{x}_{\pi_{k}^{i}}^{i} \right) \left(\boldsymbol{\psi}_{k}^{\top} \mathbf{x}_{\pi_{k}^{j}}^{j} \right)$$
$$= \sum_{k=1}^{\kappa} \boldsymbol{\psi}_{k}^{\top} \left(\mathbf{x}_{\pi_{k}^{i}}^{i} \mathbf{x}_{\pi_{k}^{j}}^{j}^{\top} \right) \boldsymbol{\psi}_{k}.$$
(25)

Substituting (25) to (22), we can rewrite it as follows,

$$\min_{\Psi} \left\{ \frac{C_2}{2} \sum_{k=1}^{\kappa} \|\psi_k\|_2^2 + \sum_{i=1}^{n} \beta_i (1 - y_i f_i) - \frac{1}{2C_1} \sum_{i,j=1}^{n} \beta_i \beta_j y_i y_j \left[\sum_{k=1}^{\kappa} \psi_k^\top \left(\mathbf{x}_{\pi_k^i}^i \mathbf{x}_{\pi_k^j}^{j^\top} \right) \psi_k \right] - \sum_{k=1}^{\kappa} \left[\frac{C_2}{2} \|\psi_k\|_2^2 - \frac{1}{2C_1} \psi_k^\top \left(\sum_{i,j=1}^{n} \beta_i \beta_j y_i y_j \mathbf{x}_{\pi_k^i}^i \mathbf{x}_{\pi_k^j}^{j^\top} \right) \psi_k \right] + \sum_{i=1}^{n} \beta_i (1 - y_i f_i) \right\}.$$
(26)

It is apparently that the objective function is composed of a submission of subfunctions over individual codewords and a term irrelevant to the codewords. Thus we can optimize each codewords one by one independently. We define the objective for each ψ_k as follows,

$$h(\boldsymbol{\psi}_{k}) = \frac{C_{2}}{2} \|\boldsymbol{\psi}_{k}\|_{2}^{2} - \frac{1}{2C_{1}} \boldsymbol{\psi}_{k}^{\top} \left(\sum_{i,j=1}^{n} \beta_{i} \beta_{j} y_{i} y_{j} \mathbf{x}_{\pi_{k}^{i}}^{i} \mathbf{x}_{\pi_{k}^{j}}^{j}^{\top} \right) \boldsymbol{\psi}_{k}, \qquad (27)$$

which is a quadratic function of ψ_k . The problem of (26) can be rewritten as follows,

$$\min_{\Psi} \left\{ \sum_{k=1}^{\kappa} h(\psi_k) + \sum_{i=1}^{n} \beta_i \left(1 - y_i f_i \right) \right\}.$$
 (28)

Ignoring the last term irrelevant to the dictionary learning, this problem can be decomposed to κ independent sub-problems,

$$\min_{\boldsymbol{\psi}_k} h(\boldsymbol{\psi}_k), k = 1, \cdots, \kappa.$$
(29)

To solve each of these problems, we use the sub-gradient deselect algorithm. Firstly, we update the instance indexes according to (24) by using previous updated codewords, and then fix them to update the codewords themselves. To update them, we fist calculate the sub-gradient function of $h(\psi_k)$ with regard to ψ_k ,

$$\nabla h(\boldsymbol{\psi}_k) = C_2 \boldsymbol{\psi}_k - \frac{1}{C_1} \left(\sum_{i,j=1}^n \beta_i \beta_j y_i y_j \mathbf{x}_{\pi_k^i}^i \mathbf{x}_{\pi_k^j}^j ^\top \right) \boldsymbol{\psi}_k.$$
(30)

Then we descent each codeword to the sub-gradient direction,

$$\boldsymbol{\psi}_{k} \leftarrow \boldsymbol{\psi}_{k} - \eta \nabla h(\boldsymbol{\psi}_{k}) = \boldsymbol{\psi}_{k} - \eta \left(C_{2} \boldsymbol{\psi}_{k} - \frac{1}{C_{1}} \left(\sum_{i,j=1}^{n} \beta_{i} \beta_{j} y_{i} y_{j} \mathbf{x}_{\pi_{k}^{i}}^{i} \mathbf{x}_{\pi_{k}^{j}}^{j}^{\top} \right) \boldsymbol{\psi}_{k} \right).$$
(31)

where η is the step size of the descent. The algorithm of updating the k-th codeword is summarized in Algorithm 1.

- Algorithm 1: Updating the *k*-th codeword.

1.
$$\boldsymbol{\psi}_{k} \leftarrow [0, \cdots, 0]^{\top};$$

2. For $t = 1, \cdots, T$
3. $\pi_{k}^{i} = \arg \max_{j=1}^{m_{i}} \boldsymbol{\psi}_{k}^{\top} \mathbf{x}_{j}^{i}$ for $i = 1, \cdots, n;$
4. $\boldsymbol{\psi}_{k} \leftarrow \boldsymbol{\psi}_{k} - \eta \left(C_{2} \boldsymbol{\psi}_{k} - \frac{1}{C_{1}} \left(\sum_{i,j=1}^{n} \beta_{i} \beta_{j} y_{i} y_{j} \mathbf{x}_{\pi_{k}^{i}}^{i} \mathbf{x}_{\pi_{k}^{j}}^{j}^{\top} \right) \boldsymbol{\psi}_{k} \right);$
5. End of For

6. Output ψ_k .

2.3 Iterative algorithm

The overall algorithm for the learning of domain transfer dictionary and its corresponding adaptation function is summarized in Algorithm 2. In this algorithm, the domain transfer dictionary and the Lagrange multiplier variables are updated alternately in a while loop. After the while loop is completed, the adaptation function parameter is recovered from the Lagrange multiplier variables.

- Algorithm 2: Learning domain transfer dictionary and adaptation function parameter (DTC).
 - 1. Input: target training set, $\{(B_i, y_i)\}_{i=1}^n$;
 - 2. Input: source training dictionary and its corresponding classifier parameter, Φ and \mathbf{v} ;
 - 3. Input: size of the domain transfer dictionary, κ ;
 - 4. Input: weights of regularization terms, C_1 and C_2 .
 - 5. Initialization: $f_i = \mathbf{v}^\top \mathbf{z}_i^{\Phi}$ for $i = 1, \dots, n$. 6. Initialization: t = 1;

 - 7. Initialization: Initialize the domain transfer dictionary $\Psi^1 = \{\psi_k^1\}_{k=1}^{\kappa}$.
 - 8. Repeat
 - (a) **For** $i = 1, \dots, n$
 - Update $\mathbf{z}_{i}^{\Psi^{t}} = \left[\max_{j=1}^{m_{i}} \psi_{1}^{t^{\top}} \mathbf{x}_{j}^{i}, \cdots, \max_{j=1}^{m_{i}} \psi_{\kappa}^{t^{\top}} \mathbf{x}_{j}^{i}\right]^{\top};$ (b)
 - (c) End of For
 - (d) Update $\beta_i^t|_{i=1}^n$ by solve the quadratic programming problem by fixing $\mathbf{z}_i^{\Psi^t} |_{i=1}^n$;

$$\beta_{i}^{t}|_{i=1}^{n} = \arg \max_{\beta_{i}|_{i=1}^{n}} \left\{ \sum_{i=1}^{n} \beta_{i} \left(1 - y_{i} f_{i}\right) - \frac{1}{2C_{1}} \sum_{i,j=1}^{n} \beta_{i} \beta_{j} y_{i} y_{j} \mathbf{z}_{i}^{\boldsymbol{\psi}^{t} \top} \mathbf{z}_{j}^{\boldsymbol{\psi}^{t}} \right\}, \quad (32)$$

s.t. $\frac{1}{n} \geq \beta_{i} \geq 0, \forall i = 1, \cdots, n.$

- $\begin{array}{ll} \text{(e)} & \mathbf{For} \ k=1,\cdots,\kappa \\ \text{(f)} & \text{Update} \ \pmb{\psi}_k^{t+1} \ \text{by fixing} \ \beta_i^t|_{i=1}^n \ \text{and using Algorithm 1;} \end{array}$
- (g) End of For
- (h) t = t + 1.
- 9. Until convergency 10. Update $\mathbf{w} = \frac{1}{C_1} \sum_{i=1}^n \beta_i^{t+1} y_i \mathbf{z}_i^{\Psi^{t+2}}$,; 11. Output: Ψ^{t+2} , w.

3 Experimental results

In this section, we use three benchmark data sets to evaluate the performance of the proposed algorithm. The method is compared to state-of-the-arts domain adaptation algorithms, and also some single domain dictionary learning algorithms. The running time of the algorithm is reported, and the convergency of the iterative algorithm is also plot.

3.1 Benchmark data sets

We use two benchmark domain transfer multi-instance data sets in our experiments, which are introduced as follows.

3.1.1 TRECVID data set

The first data set used is the TRECVID data set [26,1]. This data set is a set of key frames of video programs. The total number of the key frames is 61,901, and they belongs to 36 classes of concepts. This data set is composed of two sub-sets, which are TRECVID 2005 data set and TRECVID 2007 data set. The differences of program structure and production values between these two sub-sets are significantly, thus we treat them as two different domains. Moreover, we choose the key frames of the Chinese channel CCTV4 from TRECVID 2005 as source domain data, and the entire TRECVID 2007 data set as the target domain [5]. To present each key frame, we extract the SIFT local features, and treat each key frame as a bag of the local features. Each local feature is treated as an instance. Thus this is a multi-instance learning problem.

3.1.2 MRSC+VOC data set

The seconde data is the combination of the MRSC data set and VOC. The MRSC data is an image data set containing 4,323 images of 18 classes, and the VOC data set is also an image data set containing 5,011 images of 20 classes. The MRSC data set is publicly accessible at http://research.microsoft.com/enu s/projects/objectclassrecognition, and the VOC data set is publicly accessible at http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007. Both the two data sets share 6 common classes, which are listed as follows: aeroplane, bicycle, bird, car, cow, and sheep. However, the distractions of the data of these two data sets are significantly different, and we tree them as two different domains. Thus we can combine the images of these 6 classes of both the data sets to one cross-domain data set. The cross-domain data set is composed of a source domain set of MSRC containing 1,269 images, and a target domain of VOC containing 1,530 images [23]. Thus the total number of the images of this cross-domain data set is 2,799. To present each image, we also extract a set of SIFT local features, and treat each local feature as an instance. Each image is treated as a bag of instances.

3.1.3 20 Newsgroups data set

The third data set is a subset of the 20 Newsgroups data set. This data set is a set of 18,774 documents. The documents belong to 6 main classes, and 20 sub-classes. To construct the setting of domain learning, we set the main class "comp" to be the positive class, and the main class "rec" to be the negative class. In the positive class, we further select two sub-classes to be source domain and target domain respectively, i.e., "comp.windows.x" and "comp.sys.ibm.pc". For the negative class, we also select two sub-classes to be source domain and target domain, i.e., "rec.sport.hockey" and "rec.motorcycles". Thus the source domain contains the data of "comp.windows.x" and "rec.sport.hockey", and the target domain contains the data of "comp.sys.ibm.pc" and "rec.motorcycles". To represent each document, we treat each paragraph as a instance, and use the word-frequency feature as the feature of the instance. Thus each document is a bag of instances.

3.2 Experimental protocol

To conduct the experiment, we use the 10-fold cross-validation. The target domain data set is split to 10 sub-sets. Each sub-set is used as a training set, and the other 9 sub-sets are used as testing set. We fist train a source domain dictionary and classifier over the entire source domain data set, and then use the target training set to train the domain-transfer dictionary and its corresponding adaptive function parameter. Then the target domain predictor is obtained as the combination of the source domain predictor and the adaptive function. The target domain predictor is finally tested over the target domain test data set. Please note that only one fold is used as target domain training set. This is insufficient for training a good target domain predictor without the help of the source domain data. This setting makes it necessary to perform domain transfer learning. The average classification accuracy over the ten folds of training set is evaluated as performance measure.

3.3 Experimental results

In this section, we report the experimental results of the proposed algorithm.

3.3.1 Comparison to state-of-the-arts

We first compare the proposed algorithm, DTC, against both some common domain transfer learning methods and two domain transfer multi-instance learning methods. The compared domain transfer learning methods are listed as follows,

- Domain transfer SVM (DT-SVM) [6],
- Domain transfer multiple kernel learning (DT-MKL) [5],

Method	TRECVID	MRSC+VOC	20 Newsgroups
DT-SVM	0.208	0.362	0.856
DT-MKL	0.268	0.397	0.918
DLRC	0.196	0.384	0.862
CDSC	0.209	0.367	0.901
DTC	0.301	0.435	0.932

 ${\bf Table \ 1} \ {\rm Average \ classification \ accuracy \ of \ comparison \ among \ domain \ transfer \ learning \ methods }$

- Deep low-rank coding for transfer learning (DLRC) [4], and

- Cross-domain sparse coding (CDSC) [33].

For these methods, the dictionary used to represent the bags are not learned cross domains. The dictionary is learned before the transfer learning is conducted and it is fixed during the transfer process. We also compare the proposed domain transfer multi-instance learning to two other such algorithms, which are

- the original multiple instance transfer learning (MITL) [40], and
- knowledge transfer learning for multiple instance learning (AKTL-MIL)
 [35].

Please note that for the MITL algorithm, it assumes that there are several source domains, and the algorithm learns the predictors of different source domains and the combination coefficients of the predictors. However, in our experimental setting, we have only one source domain. Thus we only learn one source domain predictor and set its coefficient to one.

The comparisional results of different domain transfer learning algorithms are given in Table 1. From Table 1 we observe that the proposed method outperforms all the other four compared domain transfer learning methods, including linear classifier [6], multi-kernel classifier [5], deep coding [4], and sparse coding [33]. This indicates that for multi-instance domain transfer learning, using a dictionary is the most effective method to represent the multi-instance data. Moreover, the outperforming of the proposed method over the compared method is even more significant over the data sets of TRECVID and MRSC+VOC. For example, over TRECVID data set, DTC is the only algorithm that achieves an average accuracy higher than 0.300, and over MRSC+VOC, DTC is also the only algorithm that has a higher average accuracy higher than 0.400. Both these two data sets are image sets with local features. This implies that the proposed domain-transfer dictionary learning is especially suitable for computer vision tasks which represent images as collections of visual local features.

The results of different domain transfer multi-instance learning methods are given in Table 2. From Table 2, we observe that the proposed DTC algorithm also outperforms the two existing multi-instance transfer learning algorithms, especially when compared to MITL. This is not surprising, because

 ${\bf Table \ 2} \ {\rm Average \ classification \ accuracy \ of \ comparison \ among \ domain \ transfer \ multi-instance \ learning \ methods }$

Method	TRECVID	MRSC+VOC	20 Newsgroups
MITL	0.260	0.343	0.837
AKTL-MIL	0.293	0.399	0.901
DTC	0.301	0.435	0.932

Table 3 Running time (second) of comparison among domain transfer learning methods

Method	TRECVID	MRSC+VOC	20 Newsgroups
DT-SVM	1682	654	482
DT-MKL	1812	722	587
DLRC	947	365	148
CDSC	861	247	130
DTC	2043	510	254

MITL conduct the adaptation to the target domain by weighting the predictors of different source domains. However, in our experimental setting, there is only one domain, thus the domain adaptation is not accessible. AKTL-MIL also use bag-instance similarity to represent the bags, however, the dictionary is not learned. Our algorithm learns the adaptive function and the dictionary by transferring from the source domain to the target domain, and thus it is natural to obtain a good result.

3.3.2 Running time

We are also interested in the running time of the compared algorithms. The running time of the compared domain transfer learning methods are reported in Table 3. From Table 3, we observe that the proposed DTC method is the most time-consuming method, but it is still comparable to the other domain transfer learning methods. The reason is its additional process of learning the dictionary beside the domain transfer classifier. Actually, the other methods take the bag-level features as input, and thus the time of constructing the dictionary is ignored. Thus it is not a fair comparison. Moreover, considering the significant improvement of classification accuracy archived by DTC, it is still the best option over the domain transfer multi-instance learning problem. The least time-consuming method is a sparse coding method, CDSC. It also learns a dictionary, but this dictionary is used to represent the bag-level features, and it takes the bag-level features as inputs. Thus the multi-instance learning process is not involved in the learning process. Moreover, over the three data sets, the running time for the TRECVID data set is the longest, due to its large size.

We also compare the proposed method against the two domain transfer multi-instance learning method regarding the running time performance. The compared methods include MITL and AKTL-MIL. The results are reported Method

MITL	1923	521	198
AKTL-MIL	2684	721	364
DTC	2043	510	254
	TR	ECVID data set	
1			C_1
රි 0.75 - සා වා			
classification a			
aberade A 0.52		7 ~~	
0	F 0.25 1	4 16	1024 0.0525

 Table 4 Running of comparison among domain transfer multi-instance learning methods

TRECVID MRSC+VOC 20 Newsgroups

Fig. 1 Curves of changes of accuracy to changes of C_1 and C_2 .

in Table 4. It seems that MITL is comparable with DTC in the respect of running time, but AKTL-MIL is more time-consuming than DTC. This is not surprising because AKTL-MIL is a more complex algorithm than DTC.

3.4 Analysis of sensitivity of the algorithm to C_1 and C_2

We perform the analysis of the sensitivity of the proposed algorithm to the two tradeoff parameters, C_1 , and C_2 . We plot the average classification accuracy of the proposed algorithm with different values of C_1 and C_2 over the TRECVID data set in Fig. 1. According to the figure, the proposed algorithm are stable to the changes of both the two parameters.

4 Conclusions

In this paper, we study the problem of domain transfer learning with multiinstance learning. We proposed to use the multi-instance dictionary to represent the bag of instances. However, due to the significant difference between the distributions of source and target domains, the dictionary learned from the source domain is not suitable to the target domain. We propose to learn a domain transfer dictionary to solve this problem. A target domain bag is represented by both the source domain and the domain transfer dictionaries, and then classified by the source domain classifier and an adaptive function simultaneously. The target domain classification response is the combination of the source domain response and the result of the domain adaptive function. To learn the domain transfer dictionary and its corresponding adaptive function parameter, we model a minimization problem, which minimizes the complexities of the adaptive function parameters, the domain transfer dictionary, and the average classification error jointly. The optimization problem is solved by an iterative algorithm. Our proposed method is shown to be effective by experiments over three benchmark data sets. In the future, we will also use the proposed algorithm to applications of bioinformatics [28,42,37,36,22] and data analysis of nanotechnology [17,18,20,19,2].

References

- 1. Awad, G., Over, P., Kraaij, W.: Content-based video copy detection benchmarking at treevid. ACM Transactions on Information Systems **32**(3), 14 (2014)
- Chen, A., Eberle, M., Lunt, E., Liu, S., Leake, K., Rudenko, M., Hawkins, A., Schmidt, H.: Dual-color fluorescence cross-correlation spectroscopy on a planar optofluidic chip. Lab on a Chip 11(8), 1502–1506 (2011)
- Chen, Y., Bi, J., Wang, J.Z.: Miles: Multiple-instance learning via embedded instance selection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28(12), 1931–1947 (2006)
- Ding, Z., Shao, M., Fu, Y.: Deep low-rank coding for transfer learning. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 3453–3459. AAAI Press (2015)
- Duan, L., Tsang, I.W., Xu, D.: Domain transfer multiple kernel learning. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(3), 465–479 (2012)
- Duan, L., Tsang, I.W., Xu, D., Maybank, S.J.: Domain transfer svm for video concept detection. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 1375–1381. IEEE (2009)
- Fan, X., Malone, B., Yuan, C.: Finding optimal bayesian network structures with constraints learned from data. In: Proceedings of the 30th Annual Conference on Uncertainty in Artificial Intelligence (UAI-14), pp. 200–209 (2014)
- Fan, X., Tang, K.: Enhanced maximum auc linear classifier. In: Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on, vol. 4, pp. 1540–1544. IEEE (2010)
- Fan, X., Tang, K., Weise, T.: Margin-based over-sampling method for learning from imbalanced datasets. In: Advances in Knowledge Discovery and Data Mining, pp. 309– 320. Springer (2011)
- Fan, X., Yuan, C.: An improved lower bound for bayesian network structure learning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2439 – 2445 (2015)
- Fan, X., Yuan, C., Malone, B.: Tightening bounds for bayesian network structure learning. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 2439 – 2445 (2014)
- Fu, Z., Robles-Kelly, A., Zhou, J.: Milis: Multiple instance learning with instance selection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(5), 958–977 (2011)
- Hammami, N., Bedda, M.: Improved tree model for arabic speech recognition. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol. 5, pp. 521–526. IEEE (2010)
- Hammami, N., Sellam, M.: Tree distribution classifier for automatic spoken arabic digit recognition. In: Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for, pp. 1–4. IEEE (2009)

- Lin, F., Wang, J., Zhang, N., Xiahou, J., McDonald, N.: Multi-kernel learning for multivariate performance measures optimization. Neural Computing and Applications pp. 1–13 (2016)
- Ling, X., Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Spectral domain-transfer learning. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 488–496. ACM (2008)
- 17. Liu, S., Hawkins, A.R., Schmidt, H.: Optofluidic devices with integrated solid-state nanopores. Microchimica Acta 183(4), 1275–1287 (2016)
- Liu, S., Wall, T.A., Ozcelik, D., Parks, J.W., Hawkins, A.R., Schmidt, H.: Electro-optical detection of single λ-dna. Chemical Communications 51(11), 2084–2087 (2015)
- Liu, S., Yuzvinsky, T.D., Schmidt, H.: Effect of fabrication-dependent shape and composition of solid-state nanopores on single nanoparticle detection. ACS nano 7(6), 5621–5627 (2013)
- Liu, S., Zhao, Y., Parks, J.W., Deamer, D.W., Hawkins, A.R., Schmidt, H.: Correlated electrical and optical analysis of single nanoparticles and biomolecules on a nanoporegated optofluidic chip. Nano letters 14(8), 4816–4820 (2014)
- Liu, X., Wang, J., Yin, M., Edwards, B., Xu, P.: Supervised learning of sparse context reconstruction coefficients for data representation and classification. Neural Computing and Applications pp. 1–9 (2015)
- Liu, Y., Yang, J., Zhou, Y., Hu, J.: Structure design of vascular stents. Multiscale simulations and mechanics of biological materials pp. 301–317 (2013)
- Long, M., Ding, G., Wang, J., Sun, J., Guo, Y., Yu, P.: Transfer sparse coding for robust image representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 407–414 (2013)
- 24. Lu, G., Yan, Y., Ren, L., Saponaro, P., Sebe, N., Kambhamettu, C.: Where am i in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging. Neurocomputing **173**, 83–92 (2016)
- Mei, S., Zhu, H.: Multi-label multi-instance transfer learning for simultaneous reconstruction and cross-talk modeling of multiple human signaling pathways. BMC Bioinformatics 16(1), 417 (2015)
- Oomen, J., Over, P., Kraaij, W., Smeaton, A.: Symbiosis between the trecvid benchmark and video libraries at the netherlands institute for sound and vision. International Journal on Digital Libraries 13(2), 91–104 (2013)
- Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th international conference on World wide web, pp. 751–760. ACM (2010)
- Peng, B., Liu, Y., Zhou, Y., Yang, L., Zhang, G., Liu, Y.: Modeling nanoparticle targeting to a vascular surface in shear flow through diffusive particle dynamics. Nanoscale research letters 10(1), 1–9 (2015)
- Redko, I., Bennani, Y.: Non-negative embedding for fully unsupervised domain adaptation. Pattern Recognition Letters 77, 35–41 (2016)
- Wang, H., Wang, J.: An effective image representation method using kernel classification. In: 2014 IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI 2014), pp. 853–858 (2014)
- Wang, J., Wang, H., Zhou, Y., McDonald, N.: Multiple kernel multivariate performance learning using cutting plane algorithm. In: Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, pp. 1870–1875 (2015)
- Wang, J., Zhou, Y., Duan, K., Wang, J.J.Y., Bensmail, H.: Supervised cross-modal factor analysis for multiple modal data classification. In: Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, pp. 1882–1888 (2015)
- Wang, J.J.Y., Bensmail, H.: Cross-domain sparse coding. In: CIKM 2013, Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pp. 1461–1464. ACM (2013)
- Wang, J.J.Y., Sun, Y., Bensmail, H.: Domain transfer nonnegative matrix factorization. In: Neural Networks (IJCNN), 2014 International Joint Conference on, pp. 3605–3612. IEEE (2014)
- Wang, Q., Ruan, L., Si, L.: Adaptive knowledge transfer for multiple instance learning in image classification. In: AAAI Conference on Artificial Intelligence (2014)

- Wang, S., Zhou, Y., Tan, J., Xu, J., Yang, J., Liu, Y.: Computational modeling of magnetic nanoparticle targeting to stent surface under high gradient field. Computational mechanics 53(3), 403–412 (2014)
- Xu, J., Yang, J., Huang, N., Uhl, C., Zhou, Y., Liu, Y.: Mechanical response of cardiovascular stents under vascular dynamic bending. Biomedical engineering online 15(1), 1 (2016)
- Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. In: Proceedings of the 15th international conference on Multimedia, pp. 188–197. ACM (2007)
- Yang, L., Jing, L., Ng, M.: Robust and non-negative collective matrix factorization for text-to-image transfer learning. IEEE Transactions on Image Processing 24(12), 4701–4714 (2015)
- Zhang, D., Si, L.: Multiple instance transfer learning. In: Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on, pp. 406–411. IEEE (2009)
- Zhao, S., Cao, Q., Chen, J., Zhang, Y., Tang, J., Duan, Z.: A multi-atl method for transfer learning across multiple domains with arbitrarily different distribution. Knowledge-Based Systems 94, 60–69 (2016)
- 42. Zhou, Y., Hu, W., Peng, B., Liu, Y.: Biomarker binding on an antibody-functionalized biosensor surface: the influence of surface properties, electric field, and coating density. The Journal of Physical Chemistry C 118(26), 14,586–14,594 (2014)