

# Multi-Transfer: Transfer Learning with Multiple Views and Multiple Sources

Ben Tan\*

Erheng Zhong\*

Evan Wei Xiang\*

Qiang Yang\*

## Abstract

Transfer learning, which aims to help the learning task in a target domain by leveraging knowledge from auxiliary domains, has been demonstrated to be effective in different applications, e.g., text mining, sentiment analysis, etc. In addition, in many real-world applications, auxiliary data are described from multiple perspectives and usually carried by multiple sources. For example, to help classify videos on Youtube, which include three views/perspectives: image, voice and subtitles, one may borrow data from Flickr, Last.FM and Google News. Although any single instance in these domains can only cover a part of the views available on Youtube, actually the piece of information carried by them may compensate with each other. In this paper, we define this transfer learning problem as *Transfer Learning with Multiple Views and Multiple Sources*. As different sources may have different probability distributions and different views may be compensate or inconsistent with each other, merging all data in a simplistic manner will not give optimal result. Thus, we propose a novel algorithm to leverage knowledge from different views and sources collaboratively, by letting different views from different sources complement each other through a co-training style framework, while revise the distribution differences in different domains. We conduct empirical studies on several real-world datasets to show that the proposed approach can improve the classification accuracy by up to 8% against different state-of-the-art baselines.

**Keywords:** Transfer Learning, Multi-View Learning, Multiple Data Sources

## 1 Introduction

In real-world applications, the lack of labeled data makes many supervised learning algorithms fail to build accurate models. To solve the limited supervision problem, transfer learning aims to borrow knowledge from auxiliary domains to improve the target-domain model performance. Many applications have been reported, ranging from text classification [5], sentiment analysis [4], event recognition [6], to multimedia analysis [16]. Although traditional transfer works for single-source and single-view scenario, in fact, many real-world appli-

cations are complex where the auxiliary examples are often described from different perspectives and come from a variety of potential sources. For example, in video analysis, a video can be described by different issues, such as images, voice, and subtitles, where data from different views can be borrowed from different domains, such as news from Google News<sup>1</sup>, voice from Last.FM<sup>2</sup> and images from Flickr<sup>3</sup>. Another example is text classification on Google News, where 20Newsgroups<sup>4</sup> and Reuters<sup>5</sup> can be considered as source domains and cover different vocabularies of google news.

In the recent years, several approaches have been proposed to place transfer learning under the multi-view (MVTL) setting [18] or the multi-source setting (MSTL) [10]. Existing algorithms in MVTL solve the transfer learning problem where source and target domains share the same views while existing MSTL approaches consider that there are multiple sources but with one view for source and target data. In fact, in many real-world applications, multi-view information is distributed on multiple source domains and each source domain can cover only parts of the target views. For example, in video analysis, a video can be described from three different views, including images in each frame, voices and texts in subtitles. Then, different image, text and voice sources can be exploited while these sources can only cover parts of the target views. We define this problem as *Transfer Learning with Multiple Views and Multiple Sources*, and TL-MVMS for short. An intuitive way to use these rich data is to simply merge all sources or all views together, and directly employ MVTL or MSTL respectively. Unfortunately, different sources have different feature spaces and may follow different distributions, while different views from different sources may be even inconsistent with each other. Such intuitive solution may make us fail to make full use of rich source information. For example, songs on Last.FM and images from Flickr may have different probability densities and may not agree with each other on categorizing videos. On one hand, if we apply MSTL, different views will be considered equally and the inconsistency cannot

\*Hong Kong University of Science and Technology.  
{btan, ezhong, wxiang, qyang}@cse.ust.hk.

<sup>1</sup><http://news.google.com/>

<sup>2</sup><http://www.last.fm>

<sup>3</sup><http://www.flickr.com/>

<sup>4</sup><http://qwone.com/~jason/20Newsgroups/>

<sup>5</sup><http://www.daviddlewis.com/resources/testcollections/>

be removed; on the other hand, if MVTL is applied, different source distributions may make algorithms fail to build a consistent model.

Recently, co-training [2] has been demonstrated to be effective to utilize multi-view data, where a classifier built from one view will provide pseudo labeled data with high confidences to enhance the performance of another classifier from another view. Thus, even the knowledge in each view is incomplete, they can compensate each other by exchanging information. However, applying co-training simply may cause two problems: 1. due to the distribution shift in both the marginal distribution and conditional probability between source and target domains, the decision boundaries of source and target domains can be very different and hence the confidence measure is not an accurate indicator anymore; 2. on account of the joint-distribution differences, the predictions across domains are no longer consistent.

To cope with the nature of multiple views and multiple sources in TL-MVMS, we extend co-training accordingly and develop a novel solution called **multi-transfer**. Multi-transfer overcomes the above challenges from two aspects. It first introduces a harmonic-function based criterion to select the appropriate target instances. Such criterion is insensitive to the conditional probability shift. Secondly, it applies a density ratio weighting scheme to account for the marginal-distribution shift and exploits a non-parametric method to measure the joint-distribution ratio between data from two domains. This strategy re-weights the instances in source domains, in order to revise the distribution shift and build a consistent model for the target domain. We will show that, on one hand, the co-training style procedure can exploit knowledge from different views to help each other; on the other hand, the distribution revision can guarantee the robustness of knowledge transfer across different source domains. We show extensive experimental studies that our proposed method can outperform state-of-the-art transfer learning techniques on real datasets.

## 2 Problem Formulation

We define the problem, Transfer Learning with Multiple Views and Multiple Sources (TL-MVMS) as follows. The notations are summarized in Table 1. Let  $\mathcal{S} = \{S^k\}_{k=1}^N$  denote the source domains, where  $N$  is the number of sources. For each  $S^k$ , we have  $S^k = \{X_s^k, Y_s^k\} = \{(\mathbf{x}_i^k, y_i^k)\}_{i=1}^{n_k}$ , where  $n_k$  denotes the number of instances in the  $k$ -th source domain. Let  $T = \{X_t\} = \{\mathbf{x}\}_{j=1}^m$  denote the target domain, where  $m$  is the number of instances. We define the view sets as  $\mathcal{V} = \{v_\ell\}_{\ell=1}^F$ , where  $F$  is the number of views. For each source domain  $S^c$ , its view set is defined as

Table 1: Definition of Notations

Notation	Notation Description
$\mathcal{S}$	Source domains, $\mathcal{S} = \{S^k\}_{k=1}^N$
$S^k$	The $k$ -th source domain, $S^k = \{X_s^k, Y_s^k\}$
$T$	The target domain, $T = \{X_t\}$
$V_s^k$	The view set of the $k$ -th source domain
$V_t$	The view set of the target domain
$n_k$	Number of instances in $S^k$
$m$	Number of instances in $T$
$N$	Number of source domains
$F$	Number of views
$p(\mathbf{x})$	Marginal distribution of $\mathbf{x}$
$p(y \mathbf{x})$	Conditional distribution of $(\mathbf{x}, y)$
$p(y, \mathbf{x})$	Joint distribution of $(\mathbf{x}, y)$

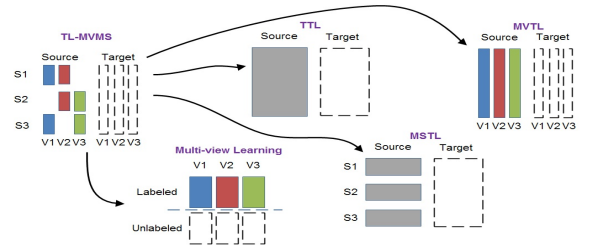


Figure 1: TL-MVMS and Other Learning Problems

$V_s^c = \{v_\ell^c\}_{\ell=1}^{f_s^c} \in \mathcal{V}$  and for the target domain, its view set is  $V_t = \{v_\ell^t\}_{\ell=1}^{f_t} \in \mathcal{V}$ . Let  $p_s^k(\mathbf{x})$ ,  $p_s^k(y|\mathbf{x})$  and  $p_s^k(\mathbf{x}, y)$  denote the marginal, conditional and joint distributions of the  $k$ -th source domain respectively, and  $p_t(\mathbf{x})$ ,  $p_t(y|\mathbf{x})$  and  $p_t(\mathbf{x}, y)$  be the parallel definitions for the target domain. The goal of TL-MVMS is to build models for  $T$  with the help of  $\mathcal{S}$ . We emphasize that this is a general framework. The difference between TL-MVMS and the previous learning problems, i.e., traditional transfer learning (TTL), multi-view learning, multi-view transfer learning (MVTL) and multi-source transfer learning (MSTL) are illustrated in Figure 1. These approaches can be considered as special cases of TL-MVMS.

- Multi-view learning:  $N = 0$  and  $f_t > 1$
- TTL:  $N = 1$  and  $f_s^k = f_t = 1$
- MVTL:  $N = 1$ ,  $f_s^k = f_t > 1$  and  $V_t = V_s^k$
- MSTL:  $N > 1$  and  $f_s^k = f_t = 1$ :

Clearly, due to the distribution shift between source and target, existing multi-view learning algorithms may fail to build consistent models for the target domain based on source data. In addition, TTL and MSTL do not consider the multi-view setting, and hence cannot take the full advantage of the source data. Finally, on account of the distribution shift among sources, MVTL may not build consistent models if simply merging all source domains together.

## 3 The Multi-Transfer Algorithm

The intuition of the multi-transfer algorithm is to regulate the model built from some views in one source

domain from the knowledge of other views in another source domain while avoiding any negative impacts of domain differences. We implement this idea by embedding transfer learning in a co-training framework. For simplicity, we assume that there are two source domains and one target domain, where the target domain contains two views and each source domain covers each of them respectively.

**3.1 Multi-transfer** Based on the co-training framework, in each iteration, we build two models  $f_1$  and  $f_2$  from two different view sets  $V_s^1$  and  $V_s^2$ , from two different source domains  $S^1$  and  $S^2$  respectively. Then, these two models are used to predict the pseudo labels of the remaining unlabeled target instances. Analog to co-training, those target instances with high prediction confidence and their predicted labels will be used to build two new models with the existing source domain data for the next iteration. However, due to the domain differences, multi-transfer needs to address two challenges (1). how to solve the distribution shift across domains for selecting appropriate unlabeled target data and (2). how to revise the distribution gap between source and target domains, for building consistent models. To select the target instances, we propose a harmonic function based instance-selection criterion that considers the joint distribution shift between two domains. Then, we introduce a non-parametric method to estimate the joint distribution ratio of each source instance, which can be treated as clues for setting their weights and used to reduce the negative effects of distribution shift. We discuss these two processes as follows.

**3.2 Target Instance Selection** In each iteration, multi-transfer selects several target instances with predicted labels to enlarge the training set of each source domain and exchange knowledge across different views. To avoid the negative impact of distribution shift, i.e.,  $p_t(\mathbf{x}, y) \neq p_s(\mathbf{x}, y)$ , we propose an unbiased criterion. Since the marginal distributions of two domains are different, i.e.  $p_t(\mathbf{x}) \neq p_s(\mathbf{x})$ , the models learned in source domains are inconsistent to the target domain [12]. Formally, let  $f_t^*$  denote the ideal hypothesis of the target domain and  $f_s$  denote the hypothesis constructed from  $n$  source instances. Then, if  $p_t(\mathbf{x}) \neq p_s(\mathbf{x})$ , we obtain  $\lim_{n \rightarrow \infty} (f_s) \neq f_t^*$ . Instead, if the density ratio of one instance  $\mathbf{x}$  between two domains is close to 1, its prediction is consistent and would be correct with high probability. Thus, we define a marginal distance-measure  $|p_s^c(\mathbf{x}) - p_t^c(\mathbf{x})|$  for each target instance  $\mathbf{x} \in X_t$  under view  $v_c$ . As we assume each source domain covers one target view, the index  $c$  also indicates the  $c$ -th source domain, i.e.,  $S^c$ . However, it may be hard to estimate

each probability density. We rewrite it as

$$(3.1) \quad \left| \log p_s^c(\mathbf{x}) - \log p_t^c(\mathbf{x}) \right| = \left| \log \frac{p_s^c(\mathbf{x})}{p_t^c(\mathbf{x})} \right|$$

We estimate the ratio  $g^c(\mathbf{x}) = \frac{p_s^c(\mathbf{x})}{p_t^c(\mathbf{x})}$  via Gaussian Process (GP) [14], which generates a function from a Gaussian distribution. Specifically, under the view  $v_c$ , we consider that the labels of all source instances from  $S^c$  to be positive and the labels of all target instances to be negative. Subsequently, after building the GP model, we can obtain the estimation  $\frac{p_s^c(\mathbf{x}|GP)}{p_t^c(\mathbf{x}|GP)} \approx \frac{p_s^c(\mathbf{x})}{p_t^c(\mathbf{x})}$ . Following the analysis in [12], the selection of unlabeled target data is unbiased after weighting with Eq.(3.1). As the conditional distributions of two domains are different, i.e.,  $p_t(y|\mathbf{x}) \neq p_s(y|\mathbf{x})$ , the decision boundaries of two domains are different too. The confidence measure  $|p(y = 1|\mathbf{x}, f) - p(y = -1|\mathbf{x}, f)|$  in co-training is no longer an indicator to the prediction risk. Thus, we utilize the local data structure in the target domain to generate a more robust criterion. We propose a harmonic-function based method, which is similar to the one in [20]. Let  $w_{ij}^c = \exp(-\sigma(\mathbf{x}_i^c - \mathbf{x}_j^c)^2)$  denote the similarity between two target instances under view  $v_c$ . The harmonic measure of  $\mathbf{x}_i^c$  is

$$(3.2) \quad \sum_{j \in N(i)} w_{ij}^c (f^{-c}(\mathbf{x}_i) - f^{-c}(\mathbf{x}_j))^2$$

where  $N(i)$  is the nearest neighbors of  $\mathbf{x}_i$  and  $f^{-c}$  is the model trained from another source. Its physical meaning is that the predictions of two instances should be similar if they are neighbors. Under the cluster-manifold assumption [1], of which meaning is that two instances are close geometrically, they tend to have the same label. We obtain that, if the conditional distributions of two domains are not very different, e.g.,  $p_t(y = 1|\mathbf{x}) \approx p_s(y = -1|\mathbf{x})$  does not hold, the predictions of  $f^{-c}$  and the true labels would be identical on some target instances. Consequently, if the instances in the target domain follow the manifold assumption, the value of Eq.(3.2) can identify these instances. Combining Eq.(3.1) and Eq.(3.2) together, we obtain a final selection criterion as

$$(3.3) \quad v^c(\mathbf{x}_i) = \left| \log \left( \frac{p_s^c(\mathbf{x}_i)}{p_t^c(\mathbf{x}_i)} + \lambda \right) \right| \sum_{j \in N(i)} w_{ij}^c (f^{-c}(\mathbf{x}_i) - f^{-c}(\mathbf{x}_j))^2$$

where  $\lambda > 0$  is to avoid the zero value. Then, in each iteration, we select  $n_a$  instances with smallest  $v^c$  values to help build new models for the next iteration.

**3.3 Source Instance Weighting** Due to the joint-distribution shift, i.e.,  $p_s(\mathbf{x}, y) \neq p_t(\mathbf{x}, y)$ , the models built on source domains may produce biased predictions

on the target domain. In other words, for a given instance, its existing probabilities are different in the source and target domains. We solve this problem by weighting the source instances with the ratio between the target and source joint-distributions. We first show that, after weighting, the models would be unbiased. In model building, we aim to build a model  $f_c$  in the source domain  $S^c$  by minimizing the objective

$$(3.4) \quad \min_{f_c} \frac{1}{n_c} \sum_{(\mathbf{x}, y) \in S^c} \frac{p_t^c(\mathbf{x}, y)}{p_s^c(\mathbf{x}, y)} (f_c(\mathbf{x}) - y)^2 + \beta \mathcal{R}(f_c)$$

Clearly, the losses of different instances are different and hence those source instances which are close to the target instances can obtain higher impacts. Due to the irrelevance to  $\mathbf{x}$  and  $y$ , we can ignore the regularization term  $\beta \mathcal{R}(f_c)$  and obtain the empirical loss of  $f_c$  as follows when  $n_c \rightarrow \infty$ .

$$\begin{aligned} \varepsilon(f_c) &= \frac{1}{n_c} \sum_{(\mathbf{x}, y) \in S^c} \frac{p_t^c(\mathbf{x}, y)}{p_s^c(\mathbf{x}, y)} (f_c(\mathbf{x}) - y)^2 \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim p_s^c(\mathbf{x}, y)} \int_{\mathbf{x}} \int_y \frac{p_t^c(\mathbf{x}, y)}{p_s^c(\mathbf{x}, y)} (f_c(\mathbf{x}) - y)^2 p_s^c(\mathbf{x}, y) d\mathbf{x} dy \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim p_s^c(\mathbf{x}, y)} \int_{\mathbf{x}} \int_y (f_c(\mathbf{x}) - y)^2 p_t^c(\mathbf{x}, y) d\mathbf{x} dy \\ &= \frac{1}{n_c} \sum_{(\mathbf{x}, y) \sim p_t^c(\mathbf{x}, y)} (f_c(\mathbf{x}) - y)^2 \end{aligned}$$

To estimate the ratio, we extend the method in [8]. Let  $g(\mathbf{x}, y)_c$  denote the ratio under view  $c$  and  $\hat{g}(\mathbf{x}, y)_c$  denote the estimated one. Let  $T_c^p$  represent the target data under the view  $v_c$  with pseudo labels from another view in each iteration. For each labeled instance  $(\mathbf{x}, y)$  in source  $S^c$ , we model  $\hat{g}(\mathbf{x}, y)_c$  as a kernel function:

$$(3.5) \quad \hat{g}(\mathbf{x}, y)_c = \sum_{i=1}^{|T_c^p|} \alpha_i k_x(\mathbf{x}, \mathbf{x}_i) k_y(y, y_i) \quad (\mathbf{x}_i, y_i) \in T_c^p$$

where  $k_x$  and  $k_y$  denote the kernel functions on instances and labels. We define  $k_x(a, b) = k_y(a, b) = \exp(-\sigma(a - b)^2)$ , where  $\sigma$  is the kernel parameter. The estimated joint distribution of the target domain is represented as  $\hat{p}_t^c(\mathbf{x}, y) = \hat{g}(\mathbf{x}, y)_c p_s^c(\mathbf{x}, y)$ . The objective is to learn the parameters  $\alpha_i$  so that the Kullback-Leibler divergence from  $\hat{p}_t^c(\mathbf{x}, y)$  to  $p_t^c(\mathbf{x}, y)$  can be minimized:

$$\begin{aligned} KL[p_t^c(\mathbf{x}, y) \parallel \hat{p}_t^c(\mathbf{x}, y)] &= \int_D p_t^c(\mathbf{x}, y) \log \frac{p_t^c(\mathbf{x}, y)}{\hat{p}_t^c(\mathbf{x}, y)} d\mathbf{x} dy \\ &= \int_D p_t^c(\mathbf{x}, y) \frac{p_t^c(\mathbf{x}, y)}{p_s^c(\mathbf{x}, y)} d\mathbf{x} dy - \int_D p_t^c(\mathbf{x}, y) \log \hat{g}(\mathbf{x}, y)_c d\mathbf{x} dy \end{aligned}$$

We can ignore the first term, which is independent of the parameters. In addition, we add a normalized term for the parameters  $\alpha = \{\alpha_i\}_{i=1}^{|T_c^p|}$ , since  $p_t^c(\mathbf{x}, y)$  is a probability density function. The objective becomes

$$(3.6) \quad \begin{aligned} \max_{\alpha} & \left[ \sum_{i=1}^{|T_c^p|} \log(\hat{g}(\mathbf{x}_i, y_i)_c) \right] \\ \text{s.t.} & \sum_{i=1}^{n_c} \hat{g}(\mathbf{x}_i, y_i)_c = n_c \quad \forall \alpha_j \geq 0 \end{aligned}$$

---

### Algorithm 1 Multi-transfer

---

- 1: **Input:** source domains:  $S = \{S^1, S^2\}$ , unlabeled target data:  $X_u, I, n_a$
  - 2: **Output:** Built model  $f$
  - 3: Set instances' weights of  $S^1$  and  $S^2$  as  $1/n_1$  and  $1/n_2$
  - 4: Build two models  $f_1^0$  and  $f_2^0$  using  $S^1$  and  $S^2$  respectively
  - 5: Let  $X_1 = X_u, X_2 = X_u, T_1^p = \emptyset, T_2^p = \emptyset$
  - 6: **for**  $i = 1$  to  $I$  **do**
  - 7:   Perform prediction on  $X_1$  using  $f_1^{i-1}$ :  $\{X_1, \hat{Y}_1\}$
  - 8:   Perform prediction on  $X_2$  using  $f_2^{i-1}$ :  $\{X_2, \hat{Y}_2\}$
  - 9:   Select instances from  $\{X_1, \hat{Y}_1\}$  and  $\{X_2, \hat{Y}_2\}$  as  $P_1$  and  $P_2$  using Eq.(3.3)
  - 10:    $X_1 = X_1 \setminus P_1, X_2 = X_2 \setminus P_2, T_1^p = T_1^p \cup P_1, T_2^p = T_2^p \cup P_2$
  - 11:   Re-estimate the weights of instances of using Eq.(3.6)
  - 12:   Build two models  $f_1^i$  and  $f_2^i$  using  $S^1 \cup T_1^p$  and  $S^2 \cup T_2^p$
  - 13: **end for**
  - 14: **Return**  $f = (f_1^I + f_2^I)/2$
- 

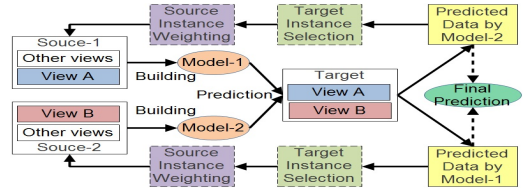


Figure 2: Main Flow of Multi-transfer

where  $n_c$  denotes the number of instances in source  $S^c$  and  $\mathbf{x}^c$  is an instance under view  $v_c$ . This is a convex optimization problem and the globally optimal solution can be obtained. After learning the  $\alpha$ , we can obtain the weight of each source instance under different views and then utilize weighted source instances to build a unbiased model for the target domain.

**3.4 The Proposed Framework** The framework of Multi-Transfer is illustrated in Figure 2 and Algorithm 1. Initially, instances in two source domains are weighted as  $1/n_1$  and  $1/n_2$ , respectively. In each iteration, two models are built from two sources under two different view sets. After that, they are utilized to predict the labels of target instances and obtain two labeled sets. To avoid the negative impact from the wrongly labeled target data, we only select those instances in these two sets with high correct prediction probability, using the adaptive criterion in Eq.(3.3). Second, these two sets can be utilized to estimate the weights of each instance in source domains. This process is repeated until exceeds the maximal iterations. Finally, the predictions of the classifiers in the last iteration will be averaged as the final predictions. We notice that, if we merge two sources into one and exploit multi-view transfer learning approaches, the different distributions and feature spaces between two source domains will misguide the model building. In addition, if we ignore the multi-view nature and exploit multi-source transfer learning approaches, we cannot utilize the cross-view compensations and may suffer from the negative impacts of inconsistencies across different views.

We analyze the time complexity as follows. Suppose the number of iterations is  $I$  and the time complexity of the base model is  $O(Q)$ . In each iteration, multi-transfer needs  $O(n_t^2)$  to compute the selection criterion values for every target instances and  $O(n_t \log n_t)$  to select the appropriate target instances. Then, it needs  $O(n_1 n_t + n_2 n_t)$  to compute the weight of each source instance and  $O(n_1 + n_2)$  to update. In summary, the whole time complexity is  $O(I(Q + (n_t + n_1 + n_2)n_t))$ .

## 4 Experiment

**4.1 A Synthetic Example** We begin by analyzing multi-transfer on a synthetic dataset, which has two source domains and one target domain with different views, as shown in Figure 3. Clearly, source domains have their own domain specific views, i.e., audio feature, and also have shared views, image feature for source domain 1 (Figure 3(a)) and text feature for source domain 2 (Figure 3(b)), with the target domain (Figure 3(c)) respectively. In addition, instances in different domains follow different distributions. Specifically, data from different source domains follow two Gaussian distributions under different views, while target data is constructed along parabola curves under another feature space.

First of all, suppose we have already learned the correct decision boundaries of source domains. As different source domains only cover one view of the target domain, when we apply the constructed boundaries on the target data, the boundaries reduce to a vertical or horizontal line due to absence of source specific view in target domain. That is to say, target data’s audio feature is equal to zero. The boundaries are shown as dash dot lines in Figure 3(c). We notice that, using single boundary from source domain will misclassify many instances. However, if we combine these two boundaries together, the boundary (the dash line) can discriminate the target data much better. Thus, we should exploit multiple views together to improve the performance.

However, due to the distribution shift, we cannot directly apply multi-view learning algorithms, e.g., co-training, whose selection criterion is purely based on confidence. Inappropriate target examples could be selected to mislead classifier updating. For example, in Figure 3(c), classifier 2 (i.e., Boundary 2) will assign negative labels to points around  $[-0.2, -0.3]$  with high confidence. Likewise, it will confidently assign positive labels to points around  $[-1.3, 0.9]$ , although, which actually are belonging to negative class. Followed by passing these selected points to source domain 1, we get Figure 3(d), where data points along line  $L$  ( $L$ : audio feature = 0) are newly added. These data will push the boundary down along image feature axis, and deteriorate the final performance. Similar incorrect results of

source domain 2 can be found in Figure 3(e). On the contrary, by revising the distribution shift, the selection process of multi-transfer is an unbiased model. It selects data points with high confidence as well as large distribution similarity. For example, data in source domain 2 are mainly distributed around  $[-0.5, -0.5]$  and  $[0.5, 0.5]$ . Taken this data distribution and classifier 2’s confidences into account, classifier 2 will select points around  $[-0.3, -0.35]$  and  $[0.4, 0.5]$  in target domain for source domain 1 (Figure 3(f)). Likewise, classifier 1 will select proper data for source domain 2 (Figure 3(g)). These selected data will push the boundaries up along image and text feature respectively and improve the performance. After re-weighting new training instances in this iteration, the decision boundaries constructed by co-training and multi-transfer are shown in Figure 3(h), from which we can see multi-transfer’s boundary is moving towards the ground-truth boundary (the solid line) and correctly classifies most data points, while co-training misclassifies lots of target instances.

**4.2 Experimental Setting** We evaluate the performance of multi-transfer algorithm on two real-world text data collections, 20 Newsgroups and spam detection, and compare to three state-of-the-art methods, i.e., *LatentMap* [15], co-adaptation [13] and co-training [2]. These baseline methods stand different learning paradigms. *LatentMap* is a traditional transfer learning approach that considers only one source domain under single-view, co-adaptation is a multi-view transfer learning which does not consider the domain differences across multiple sources, and co-training is a representative multi-view learning algorithm. The performance is measured with classification accuracy on unlabeled target data. For co-adaptation, *LatentMap* is introduced as a base classifier. For co-training and multi-transfer, SVM and C4.5 are adopted.

**4.3 Data Description** The data processing procedure is as follows. First, each document is converted to a term-frequency vector. Secondly, to reduce the number of features, we remove the vocabularies whose frequency counts are less than 1% of the document count. Finally, term frequency is used as the feature value in the experiments. **The 20 Newsgroups data set** contains the top categories, such as ‘comp’, ‘sci’, ‘rec’ and ‘talk’. Each category has some sub-categories, such as ‘sci.crypt’ and ‘sci.med’. We use 4 main categories to generate 5 datasets, in each of which two top categories are chosen for generating binary classification tasks. With a hierarchical structure, for each category, all of the subcategories are then organized into three parts, where each part is of different distribution.

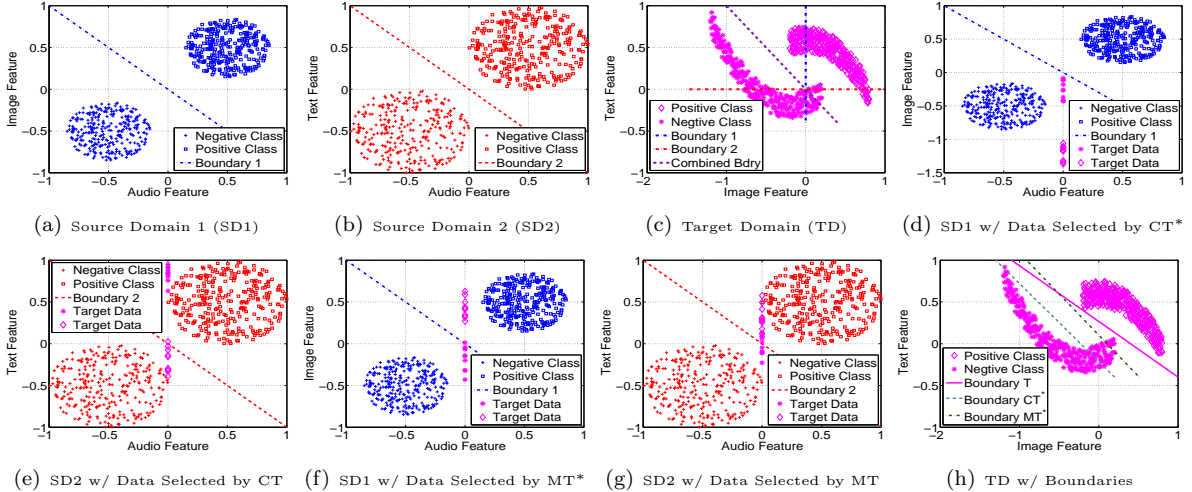


Figure 3: A Synthetic Example to Illustrate the Problem of Transfer Learning with Multiple Views and Multiple Sources. (\*CT: for co-training; MT: multi-transfer; Boundary CT: Boundary constructed by co-training; Boundary MT: boundary constructed by multi-transfer.)

Table 2: Dataset Description

Dataset	Source domain-1	Source domain-2	Target domain	#S-1	#S-2	#T
20-Newsgroup						
rec-vs-comp	autos : misc	baseball : mac	hockey : windows	1164	1169	1190
rec-vs-tech	autos : guns	motorcycles : mideast	hockey : misc	1137	1160	1062
sci-vs-comp	electronics : graphics	med : misc	space : windows	1172	1166	1185
sci-vs-tech	crypt : guns	electronics : mideast	med : misc	1139	1155	970
comp-vs-tech	graphics : guns	misc : mideast	windows : politics	1126	1136	1065
Spam Detection						
Filter 1	User 0	User 1	User 2	2500	2500	2500
Filter 2	User 0	User 2	User 1	2500	2500	2500
Filter 3	User 1	User 2	User 0	2500	2500	2500

Therefore, one part can be treated as the target domain data and the other two are used for the source domain purpose. To generate the multi-view in multi-source, we further let the vocabularies of these two source domains be overlapping the ones in the target domain but not identical. To this end, we can see that each dataset has two source domains and one target domain, each of which discusses different sub-category topics. We can also notice that the dataset has four views: (1) each source domain has one specific view (i.e., domain specific vocabularies) and one shared view with the target domain (i.e., vocabularies shared with target domain) and (2) target domain has two views, which are respectively shared with two source domains. Besides, the distribution of shared views in the target domain is different from that in the source domains due to the distinct sub-category topics. **The spam detection data set** is from Task A of ECML/PKDD Discovery Challenge 2006. The task aims to construct spam filters for 3 users, each of which has 2500 emails. The emails of a user consist of 50% spams and 50% non-spams. In addition, the data distribution between users is different. That is to say, users have their specific vocabularies as well as common ones. In our experiment, we use two

users as two source domains and another user as the target domain. The details of datasets are reported in Table 2. Their dimensions range from 2405 to 5984. In all datasets, the source instances are fully labeled, while the target domain contains only unlabeled data.

**4.4 Performance** Figure 4 presents the performance on each data set given by LatentMap, co-adaptation, co-training and multi-transfer. In each subfigure, the method with higher histogram has better performance than that with lower one. Multi-transfer always achieves the best classification accuracy. LatentMap and co-adaptation seems fail to transfer knowledge from source domain to target domain in this multi-view setting. Specifically, LatentMap does not consider the multi-view nature in the problem and co-adaptation does not consider the differences between source domains. Besides, without taking distribution shift into account, co-training obtains lower accuracy than multi-transfer. Within multi-transfer, we can see that SVM as base learner is more accurate than C4.5, since SVM is known as a first choice classifier for traditional classification problem. However, when using C4.5 as base learner, multi-transfer obtains much higher accuracy than co-

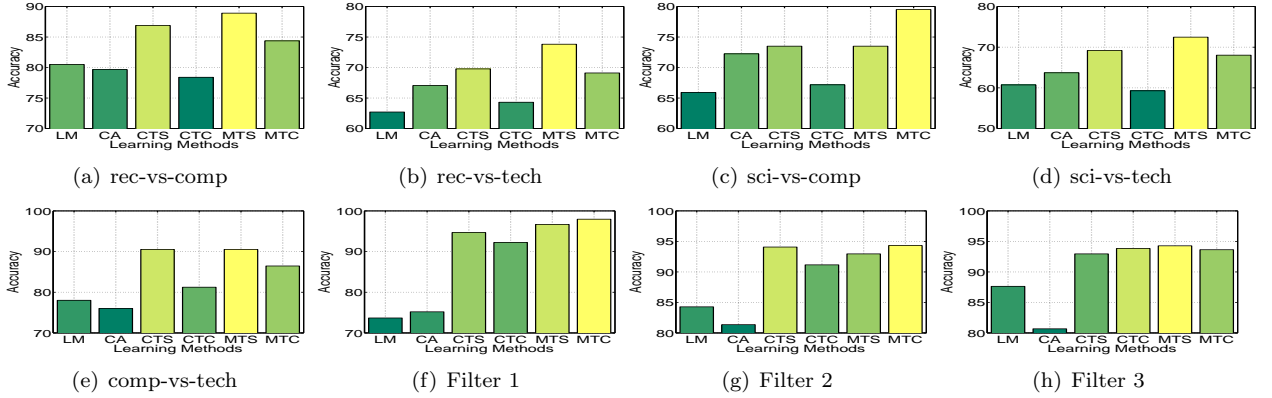


Figure 4: Performance on Different Datasets. (LM: LatentMap; CA: Co-Adaptation; CTS: Co-Training with SVM; CTC: Co-Training with C4.5; MTS: Multi-Transfer with SVM; MTC: Multi-Transfer with C4.5.)

training. Especially, on 20NG sci-vs-comp dataset, the accuracy of multi-transfer is over 10 percent higher.

**4.5 Effect of Model Parameter** As mentioned before, our algorithm has three parameters, which directly impact the final performance. The first one is the number of nearest neighbors in KNN. The selection process is easily influenced by noise when small  $K$  is used, while large  $K$  leads to a very smooth confidence distribution on unlabeled data. From Figure 5(a), we can see the trend where the performance improves when  $K$  grows and decreases when  $K$  grows too large. So  $K = \{5, 7\}$  are suitable choices. From the result, we can also see that our method always outperforms co-training.

Secondly,  $n_a$ , the number of selected unlabeled data with most confident labels, is also important for classifier updating. It is easy to understand that the classifier is updated in a smooth way when small value  $n_a$  is used. On the contrary, large value  $n_a$  brings unstable updating process. This phenomenon can be seen in Figure 5(b). However, adding a large set augments labeled data quickly and accelerates the algorithm. Considering the pros and cons, we add 10 percent unlabeled data in each iteration. Therefore, the algorithm evolves fast at the beginning and goes stable afterwards, as shown in Figure 5(c).

The last aspect we care about is the convergence. From Figure 5(c) we can see that the classifier becomes more accurate on target domain when target data are added into the learning process successively and then converges. In the experiments, we find that setting maximum iteration as 25 works well in most cases.

**4.6 Model Analysis** Besides the parameters, base learner selection and the relationship between source and target domains also have important impacts upon the classification accuracy. We conduct extensive experiments to test these impacts. Firstly, the strate-

gy of base learner selection in one iteration affects the performance in the next iteration. For example, an alternative way of using a same base learner is running different classifiers on two domains. In this method, classifiers can learn from each other in each iteration. We use this strategy on 20NG’s rec-vs-comp data set and get its classification accuracy of 87.82%, which lies between performances of multi-transfer with SVM or C4.5 as base learner. Also, from the Figure 6(a), we can see that the algorithm converges.

Secondly, we test how the relationship between source and target domains influences the performance of multi-transfer. For instance, in both source domains, we remove part of common features shared by source and target domains, then train classifiers on these incomplete views. By changing the removal-ratio, which is the percentage of removed features, we obtain the performance curve given by the solid line in Figure 6(b). In this figure, we can see that even 60% common features in each view are removed, the performance of multi-transfer is still better than that of classifier trained on single complete domain, whose features are never deleted. We also remove the common features in one view while keeping the other unchanged. This allows us to study how source views affect the final performance. By removing with increasing proportions, we obtain the classification accuracy results in Figure 6(c). Clearly, in this setting, the performance is better than a complete single source. In addition, the effect of each source is quite different, since the gap between curves is large. That is why we exploit multi-view knowledge collaboratively rather than combining views.

Finally, we analyze the situation where only common features are used as a single view in source and target domains followed by randomly splitting them into two views. This setting is used to test the performance of multi-transfer and co-adaptation with the condition where the distribution of two source views is the same.

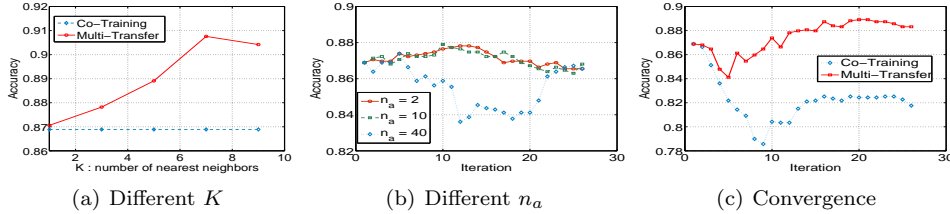


Figure 5: Parameter Analysis on 20NG’s rec-vs-comp Dataset

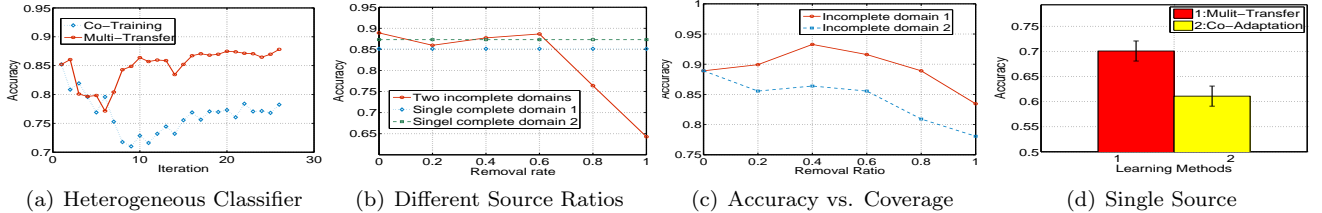


Figure 6: Model Analysis on 20NG’s rec-vs-comp Dataset

By repeating the splitting five times, we plot the means and variances of classification accuracies in Figure 6(d), where multi-transfer is still better than co-adaptation.

It is worth noting that the experiments are conducted on text data sets, but the algorithm could be directly applied on data set with multiple heterogeneous views. The reason is that, all the views are using completely different words in data sets; any view can be replaced by other type of feature, such as image.

## 5 Related Works

We summarize the related works on multi-view learning, transfer learning and their variants. Generally, the studied problem in this paper can be considered as a general framework which unifies all these learning tasks.

**Multi-view Learning** In many real-world applications, examples are represented by multiple views. Co-training [2] is a representing multi-view learning method which first learns a separate classifier for each view using any labeled examples and then pick the most confident predictions of each classifier on the unlabeled data to iteratively construct additional labeled training data. In [9], the authors incorporate the consistency Laplacian term into multi-view semi-supervised learning problems. However, most existing multi-view learning methods are for the single-domain settings instead of cross-domain.

**Traditional Transfer Learning (TLL)** Transfer learning (TL) addresses the problem of insufficient labeled data in a target domain by using auxiliary data in related but different source domains [11]. Two representative techniques for transfer learning are instance weighting [5], which extends Adaboost to filter those useless source domain data, and feature mapping [15, 19] which transfers knowledge across domains through kernel based dimension reduction. However, these traditional transfer learning approaches focus

on addressing the distribution shift across domains but work with a single source domain under a single-view.

**Multi-view Transfer Learning (MVTL)** Several approaches have been proposed to handle the situation where data from source and target domains are composed by multiple views. For example, the co-adaptation algorithm proposed in [13] uses the labeled source domain examples to construct classifiers and then applies the co-training algorithm to construct the classifier for the target domain. Co-training has been extended to cross-domain context by adding a feature selection process [4]. Recently, a maximal margin based method [18] is introduced to integrate the multi-view and transfer learning nature in a principled way. However, these works assume that there is only one source domain and source and target views are identical.

**Multi-source Transfer Learning (MSTL)** There are a few research works on multi-source transfer learning. For example, the work in [17] extends TrAdaboost [5] by adding a wrapper boosting framework on weighting each source domain. [3] presents a linear combination over multiple sources to reach a consensus. However, these approaches work under single-view setting. Recently, a close related work is proposed in [7], which addresses a multi-task learning problem under multiple views. It proposes a graph-based algorithm to capture the relations among different views in different tasks. However, it requires that all tasks contain labeled data and does not consider the distribution shift among domains.

## 6 Conclusion

In this paper, we studied a novel and general transfer learning problem: *Transfer Learning with Multiple Views and Multiple Sources*, where the source and target domains are under multiple views and the knowledge of target views are distributed on different source do-



mains. We have introduced a multi-transfer algorithm, which works in an iterative manner to predict the labels of the unlabeled target data. Comparing with previous works, multi-transfer considers the domain differences and multi-view nature together to perform cross-domain knowledge transfer. Following the co-training process, in each iteration, the target data with pseudo labels from one domain can be exploited to enhance the model building in another domain. Besides, we proposed two novel heuristics in each iteration to overcome the distribution shift. We found that by applying density-weighted harmonic function, the proposed criterion is unbiased to select high-confidence target data. In addition, it is better to estimate the importance of each source instance, which helps build a consistent model for the target domain. We conducted empirical studies on two real text collections, 20-newsgroup and spam detection, where the proposed method can boost several state-of-the-art algorithms as high as 8% on accuracy.

We carried out experiments under a setting that contains two source domains and one target domain while each source domain covers one view of the target. In our future work, we plan to extend the experiments over multiple sources and multiple views, and on a more general setting that the source and target views may be inconsistent and the source views may be overlapping. In addition, we would consider extending the algorithm under heterogeneous contexts, where the feature or label spaces are different in source and target domains.

**Acknowledgement** We thank the support of Hong Kong CERG Projects 621010, 621211 and Hong Kong ITF Project GHX/007/11.

## References

- [1] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
- [2] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.
- [3] Rita Chattopadhyay, Jieping Ye, Sethuraman Panchanathan, Wei Fan, and Ian Davidson. Multi-source domain adaptation and its application to early detection of fatigue. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–725, 2011.
- [4] M. Chen, K. Weinberger, and J. Blitzer. Co-Training for Domain Adaptation. In *Advances in Neural Information Processing Systems 24*, 2011.
- [5] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200, 2007.
- [6] Lixin Duan, Dong Xu, and Shih-Fu Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [7] Jingrui He and Rick Lawrence. A graphbased framework for multi-task multi-view learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 25–32, 2011.
- [8] T. Kanamori, T. Suzuki, and M. Sugiyama. Theoretical analysis of density ratio estimation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E93-A(4):787–798, 2010.
- [9] Guangxia Li, Steven C. H. Hoi, and Kuiyu Chang. Two-view transductive support vector machines. In *Proceedings of the SIAM International Conference on Data Mining*, pages 235–244, 2010.
- [10] Ping Luo, Fuzhen Zhuang, Hui Xiong, Yuhong Xiong, and Qing He. Transfer learning from multiple source domains via consensus regularization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 103–112, 2008.
- [11] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.
- [12] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.
- [13] Gokhan Tur. Co-adaptation: Adaptive co-training for semi-supervised learning. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3721–3724, 2009.
- [14] Christopher K.I. Williams and David Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1342–1351, 1998.
- [15] Sihong Xie, Wei Fan, Jing Peng, Olivier Verscheure, and Jiangtao Ren. Latent space domain transfer between high dimensional overlapping distributions. In *Proceedings of the 18th international conference on World wide web*, pages 91–100, 2009.
- [16] Jun Yang, Rong Yan, and Alexander G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th international conference on Multimedia*, pages 188–197, 2007.
- [17] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *The 23rd IEEE Conference on Computer Vision and Pattern Recognition*, pages 1855–1862, 2010.
- [18] Dan Zhang, Jingrui He, Yan Liu, Luo Si, and Richard Lawrence. Multi-view transfer learning with a large margin approach. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1208–1216, 2011.
- [19] Erheng Zhong, Wei Fan, Jing Peng, Kun Zhang, Jiangtao Ren, Deepak Turaga, and Olivier Verscheure. Cross domain distribution adaptation via kernel mapping. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1027–1036, 2009.
- [20] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003.