A survey of multi-source domain adaptation

Shiliang Sun*, Honglei Shi, Yuanbin Wu

Shanghai Key Laboratory of Multidimensional Information Processing,
Department of Computer Science and Technology, East China Normal University,
500 Dongchuan Road, Shanghai 200241, P.R. China

Abstract

In many machine learning algorithms, a major assumption is that the training and the test samples are in the same feature space and have the same distribution. However, for many real applications this assumption does not hold. In this paper, we survey the problem where the training samples and the test samples are from different distributions. This problem can be referred as *domain adaptation*. The training samples, always with labels, are obtained from what is called source domains, while the test samples, which usually have no labels or only a few labels, are obtained from what is called target domains. The source domains and the target domains are different but related to some extent; the learners can learn some information from the source domains for the learning of the target domains. We focus on the multi-source domain adaptation problem where there is more than one source domain available together with only one target domain. A key issue is how to select good sources and samples for the adaptation. In this survey, we review some theoretical results and well developed algorithms for the multi-source domain adaptation problem. We also discuss some open problems which can be explored in future work.

Keywords: machine learning, multi-source learning, domain adaptation, transfer learning

*Corresponding author. Tel.: +86-21-54345183; fax: +86-21-54345119.
Email address: shiliangsun@gmail.com, slsun@cs.ecnu.edu.cn (Shiliang Sun)

1. Introduction

In machine learning, most models such as Gaussian process (GP), linear discriminative analysis (LDA), support vector machine (SVM) [1, 2] and principal component analysis (PCA), assume that training samples are drawn according to the same distribution as the unseen test samples. Uniform convergence theory guarantees that a model's empirical training error is close to its true error with high probability. However, there are many cases in practice where the training and the test distributions differ. We wish to train a model in one or more domains (called source domains) and then apply it to another different but related domain (called target domain). Such learning task is known as *domain adaptation* [3, 4, 5, 6, 7, 8], which is confronted in many applications, like computer vision [9, 10, 11, 12], sentimental analysis [13, 14, 15, 16], natural language processing [17], video concept detection [18, 19], and wifi localization detection [20]. In these problems, users are generally reluctant to annotate abundant samples (like consumer videos, or the reviews for certain products) to train an effective model for later classification. What they have are a set of limited labeled samples and a large number of unlabeled data. The task is to combine the labeled source data and unlabeled target data to classify the target data as correctly as possible. The difficulty lies in the mismatch between the source distribution and the target distribution. Domain adaptation approaches explicitly or implicitly handle the mismatch between data distributions of the source and target domains.

Domain adaptation is one of the branches of transfer learning. According to Pan et. al [8], transductive transfer learning can be categorized into two cases. The first case is that the feature spaces between the source and target domains are different, i.e. $\mathcal{X}_S \neq \mathcal{X}_T$. The second case is that the feature spaces between the source and target domains are the same, but the marginal probability distributions of the input data are different, i.e. $\mathcal{X}_S = \mathcal{X}_T$, but $P(\mathcal{X}_S) \neq P(\mathcal{X}_T)$. The latter case can be referred as *domain adaptation*. Domain adaptation is different from semi-supervised learning and data set shift [21]. It assumes that the labeled and unlabeled data come from different but related domains, while semi-supervised learning methods employ both labeled and unlabeled data from the same domain. On the other hand, data set shift assumes that the joint distribution $P(\mathcal{X}, \mathcal{Y})$ of input $\mathcal{X}$ and output $\mathcal{Y}$ changes across the source and target domains, i.e. $P(\mathcal{X}, \mathcal{Y})_S \neq P(\mathcal{X}, \mathcal{Y})_T$. However, the focus of domain adaptation is that the marginal probability distributions of the input data are different.

For the single-source domain setting, much work has been developed. Several theoretical analyses have considered the single-source domain adaptation problem. Ben-David et al. [22] defined two sources of adaptation errors. Firstly, feature distributions differ between the source and the target domains, which means that the test examples are different from the training examples in the sense of data distributions. Since many applications usually use the lexical items as features, this problem can be especially difficult. In general, this problem can be addressed by using the unlabeled target data since feature distributions can be measured and aligned without annotated examples. Secondly, the decision functions differ between domains. The instance may be labeled differently depending on the domains. To correct this error, one needs the knowledge of the labeling function, which can only be gained from labeled target samples. Dredze et al. [23] showed how domain adaptation for parsing is difficult when annotation guidelines differ for different domains.

In addition to the theoretical analyses, there is also much empirical work on algorithms for single-source domain adaptation. Chelba and Acero [24] trained a classifier on the source domain, and then used the maximum a posteriori (MAP) estimation of the weights of a maximum entropy target domain classifier. The prior is a Gaussian distribution whose mean is equal to the weights of the source domain classifier. Daume and Marcu [25] used an empirical Bayes model to estimate a latent variable model which groups the instances into two categories domain-specific or common across both domains. Blitzer et al. [26] introduced structural correspondence learning to automatically induce correspondences among features from two domains, without using the labeled target data. Unlike the work of Daume and Marcu [25], they found a common representation for features from different domains, rather than instances.

Often in practice, one may be offered more than one source domain for training. It is wasteful if we only use one source for training. The most common way is to add up all the sources as one source. However, this approach ignores the difference among the sources. A second way is to train a classifier per source and combine these multiple base classifiers. Based on the principles of risk minimization, one can derive a solution which assigns weights for each base model, and combines multiple base models to maximize their combined accuracy on the new domain. The combined model can get a reasonable high accuracy for the target task. The second model for multi-source domain adaptation is displayed in Figure 1.

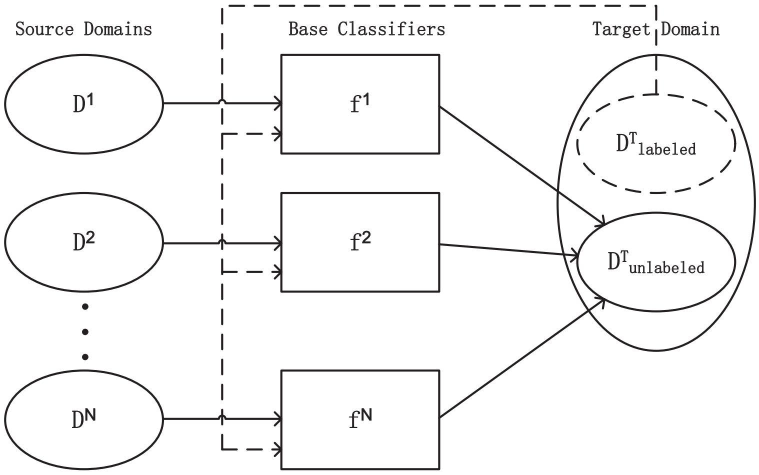

Figure 1: The model for multi-source domain adaptation.

One popular domain adaptation problem arises in text classification tasks where one can retrieve information from several source domains and make predictions about another target domain. In natural language processing, sentiment classification is a task of classifying documents according to the sentiments. Given a piece of text (usually a review or essay), what is of interest is whether the opinion expressed by the text is positive or negative. Sentiment analysis is useful on a number of text domains, ranging from stock message boards to congressional floor debates. In some domains (e.g. movie reviews and book reviews), one can have plenty of labeled data for machine learning algorithms to train a model for classification, while there are also many domains (e.g. piano reviews) that can not have enough data available for training. Domain adaptation algorithms can solve such problems by using the domains which have plenty of labeled as sources, and domains lack of labeled data as target domains. Usually, the source and target domains are assumed to be different but related.

Table 1: Summarization of important methods introduced in this paper.

Categories	References	Characteristics
Theoretical methods for multi-source domain adaptation	Crammer et al. [27]	The authors gave a general theorem which establishes a general bound on the expected loss of the model by minimizing the empirical loss on the nearest k sources. These nearest k sources are a recommended set of sources to be used for training the model.
	Mansour et al. [28]	The authors studied two types of combinations for multi-source domain adaptation problems, i.e., a linear combining rule and a distribution weighted combining rule. They proved that the first combination rule may perform poorly in real-world applications while the second one can guarantee a loss of at most ϵ for any target mixture of the source distributions.
	Ben-David et al. [3]	By introducing an $\mathcal{H}\Delta\mathcal{H}$-distance between the target and the source domains, the authors gave two learning bounds for empirical risk minimization. Two different types of $\mathcal{H}\Delta\mathcal{H}$-distance are used: a pair-wise $\mathcal{H}\Delta\mathcal{H}$-distance and a $\mathcal{H}\Delta\mathcal{H}$-distance between the target domain and the weighted combination of source domains.

continued on next page

Well-developed algorithms	Feature representation approaches	Chattopadhyay et al. [29]	The authors proposed a weighting scheme which measures the conditional probability distribution differences across multiple source domains.
		Sun et al. [30]	The authors proposed a two-stage multi-source domain adaptation methodology. The data from multiple sources are re-weighted based on marginal probability differences in the first stage. In the second stage the source domains are re-weighted based on conditional probability differences.
		Duan et al. [31, 10, 32]	The authors introduced a data-dependent regularizer into the objective of support vector regression using the ϵ-insensitive loss. In [31, 32] all the sources are used for the domain adaptation problem while in [10] a data-dependent regularizer for domain selection is proposed.
	Combination of prelearned classifiers	Schweikert et al. [33]	Given some labeled target data, the authors proposed a multiple convex combinations of prelearned source classifiers and target classifiers for multi-source domain adaptation.
		Sun and Shi [34]	The authors proposed a dynamic Bayesian learning framework for multi-source domain adaptation. The domain priors for the source domains are constructed with the Laplacian matrix on the unlabeled target data. The point-wise likelihood is calculated according to the distance of the k-nearest neighbors.

continued on next page

Well-developed algorithms	Combination of prelearned classifiers	Yang et al. [12]	The authors proposed adaptive support vector machines by adding a delta function into the standard support vector machine objective. The delta function is learned among source classifiers and target classifier using an objective function similar to SVMs.
		Tu and Sun [35]	The authors proposed a cross-domain representation learning framework that combines class-separate objectives and domain-merge objectives simultaneously to learn a data representation model. The framework not only maximizes the differences between classes but also minimizes the differences between domains.
		Xu and Sun [36, 37]	The authors proposed a multi-view adaboost transfer learning method. The target classifier is learned by combining the weighted prelearned classifiers using a new parameter.
		Xu and Sun [38, 39]	The authors proposed a part-based transfer learning method. By dividing the feature space into N parts, the dataset can be divided into N parts containing different information. Then N part-based source classifiers for each source can be trained to form the ensemble target classifier.

Another domain adaptation application is the computational advertising system. The system may rank advertisements for queries originating from many different countries, in many different languages, and covering a variety of product domains. A system trained on all queries together, agnostic with respect to such properties, may benefit from having a large quantity of training data. However, it is also possible that data sources have conflicting properties that reduce the performance of a single model trained in this manner. In this case, it would be preferable to train separate systems. In fact, both approaches are inadequate. Data sources typically share some common characteristics and behaviors, though differ from one another. A single system obscures differences, while separate systems ignore similarities.

Besides the above applications, some efforts have also been made on domain adaptation for event recognition in consumer videos [18, 10]. For example, Duan et al. [10] learned a classifier which uses both the SIFT features of web images from source domains and the space-time (ST) features as well as SIFT features from the target domain to make decisions for the target video.

In this paper, we investigate both theoretical analyses and existing algorithms for multi-source domain adaptation. We hope to provide a useful resource for the research of multi-source domain adaptation. The rest of the survey is organized as follows. In Section 2, some theoretical analyses are provided. Section 3 covers some well-developed algorithms. We summarize Section 2 and Section 3 in Table 1 for a quick access to the methods introduced in this paper. In Section 4, some performance evaluation measurements as well as publicly available datasets about multi-source domain adaptation are listed. Conclusions and some worth-working lines for multi-source domain adaptation are summarized in Section 5.

2. Theoretical analyses for multi-source domain adaptation

We formalize the multi-source domain adaptation problem as follows. Let $\mathcal{X}$ be the input space, D be a distribution on $\mathcal{X}$, and $f : \mathcal{X} \to \mathbb{R}$ be the target function to learn. A domain is defined as a pair $\langle D, f \rangle$. Let $\mathcal{L}(f(x), y) \in \mathbb{R}$ be a loss function with respect to f. Suppose we have N distinct sources, with each source S_j associated with an unknown distribution D_j over the input points, and an unknown labeling function f_j. Each source S_j has $m_j = \eta_j m$ labeled samples where m is the total sample number from all the sources, and $\eta_j \in [0, 1]$, $\sum \eta_j = 1$. The objective is to use these samples to train a model to perform well on a target domain $\langle D_T, f_T \rangle$. The multi-source domain adaptation problem is to combine each source S_j to derive a hypothesis h with a small loss $\mathcal{L}(f_T(x), h(x))$ on the target domain.

Blitzer et al. [40] gave a bound on the error rate of a hypothesis derived from a weighted combination of the source data sets for the specific case of empirical risk minimization.

Crammer et al. [27] addressed a problem where multiple sources are present. But the nature of the problem differs from adaptation since the distribution of the input points is the same for all these sources, and only the labels change due to the varying amounts of noise. They gave a general bound on the expected loss of the model by minimizing the empirical loss on the nearest k sources. These nearest k sources form a recommended set of sources. Two key ingredients needed to apply this bound were introduced: an approximate triangle inequality and a uniform convergence bound.

Definition 1. *For $\alpha \geq 1$, the α-***triangle inequality** *holds for a hypothesis space $\mathcal{H}$ if for all $g_1, g_2, g_3 \in \mathcal{F}$ the following inequality holds:*

$$e(g_1, g_2) \leq \alpha(e(g_1, g_3) + e(g_1, g_2)). \tag{1}$$

where $\mathcal{F}$ is a class of candidate models and $e(g_1, g_2) = E_{x \sim D}\mathcal{L}(g_1(x), g_2(x))$ is the expected loss function. The parameter $\alpha \geq 1$ is a constant that depends on $\mathcal{H}$ and $\mathcal{L}$.

Definition 2. *A* **uniform convergence bound** *for a hypothesis space $\mathcal{H}$ and loss function $\mathcal{L}$ is a bound that states that for any $0 < \delta < 1$, with probability at least $1 - \delta$ for any $h \in \mathcal{H}$*

$$|\hat{e}(h) - e(h)| \leq \beta(n, \delta), \tag{2}$$

where $\hat{e}(h) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(h(x_i), y_i)$ for n observations $(x_1, y_1), \ldots, (x_n, y_n)$ generated independently according to distributions $P_1, \ldots, P_n$ and $e(h) = E[\hat{e}(h)]$ where the expectation is taken with respect to $(x_1, y_1), \ldots, (x_n, y_n)$. Here β is function of the number of observations n and the confidence δ, and depends on $\mathcal{H}$ and $\mathcal{L}$.

Definition 2 asserts that for every model in $\mathcal{H}$, its empirical loss on a sample of size n and the expectation of this loss will be "close" when $\beta(n, \delta)$ is small.

In general the function β will incorporate standard measures of the complexity of $\mathcal{H}$, and will be a decreasing function of the sample size n, as in the classical $O(\sqrt{d/n})$ bounds of the VC theory.

Given K source domains $S_1 = \langle D, f_1 \rangle, S_2 = \langle D, f_2 \rangle, \ldots, S_K = \langle D, f_K \rangle$, and for S_j, there are n_j samples (x_i^j, y_i^j). Denote $n_{1:k} = \sum_{j=1}^{k} n_j$. To analyze the multi-source domain adaptation problem, the performance of hypothesis $\hat{h}_k$ can be measured as:

$$\hat{h}_k = \arg\min_{h \in \mathcal{H}} \frac{1}{n_{1:k}} \sum_{j=1}^{k} \sum_{i=1}^{n_j} \mathcal{L}(h(x_i^j, y_i^j)). \tag{3}$$

Theorem 1. *Let e be the expected loss function for loss $\mathcal{L}$, and $\mathcal{F}$ be a class of models for which the α-triangle inequality holds with respect to e. Let $\mathcal{H} \subseteq \mathcal{F}$ be a class of hypothesis models for which there is a uniform convergence bound β for $\mathcal{L}$, $f = f_1, f_2, \ldots, f_K$ are the unknown source models in $\mathcal{F}$, $\epsilon_i = \max(e(f, f_i), e(f_i, f))$, and w.o.l.g., let $\epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_K$. For any δ such that $0 \leq \delta \leq 1$, with probability at least $1 - \delta$, for any $k \in \{1, \ldots, K\}$,*

$$\begin{aligned} e(f, \hat{h}_k) \leq & \alpha^2 \min_{h \in \mathcal{H}} \{e(f, h)\} + (\alpha + \alpha^2) \sum_{i=1}^{k} (\frac{n_i}{n_{1:k}}) \epsilon_i \\ & + 2\alpha\beta(n_{1:k}, \delta/2K). \end{aligned} \tag{4}$$

The bound is on the expected loss incurred by using all data sources within a given disparity of the target source. According to the theorem, optimizing this bound can get a recommended subset of the data to be used in learning a model for each source. Thus one can avoid the negative adaptation. The optimized number of sources k^* to be used for estimating the target f is given as

$$k^* = argmin_k((\alpha + \alpha^2) \sum_{i=1}^{k} (\frac{n_i}{n_{1:k}}) \epsilon_i + 2\alpha\beta(n_{1:k}, \delta/2K)). \tag{5}$$

To demonstrate the applicability of the general theory given by Theorem 1, Crammer et al. [27] also gave a bound for (noise-free) binary classification and (noise-free) regression.

In (noise-free) binary classification, the loss function $\mathcal{L}(h(x), y)$ is defined as 0 if $y = h(x)$ and 1 otherwise, and the expected loss is given by $e(g_1, g_2) = E_{x\sim D}[\mathcal{L}(g_1(x), g_2(x))] = \mathbf{Pr}_{x\sim D}[g_1(x) \neq g_2(x)]$. For 0/1 loss it is easy to see that the 1-triangle inequality holds. Classical VC theory provides us with uniform convergence as follows.

Lemma 1. *Let $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ be a class of functions with VC dimension d, and let $\mathcal{L}(h(x), y) = |y - h(x)|$ be the 0/1 loss. The following function β is a uniform convergence bound for $\mathcal{H}$ and $\mathcal{L}$ when $n \geq d/2$:*

$$\beta(n, \delta) = \sqrt{\frac{8(d \ln(2en/d) + \ln(4/\delta))}{n}}\ . \tag{6}$$

With Lemma 1 and Theorem 1, the bound for binary classification is given as follows.

Theorem 2. *Let $\mathcal{F}$ be the set of all functions from an input set $\mathcal{X}$ into $\{0, 1\}$ and let d be the VC dimension of $\mathcal{H} \in \mathcal{F}$. Let e be the expected 0/1 loss. Let $K, f = f_1, f_2, \ldots, f_K \in \mathcal{F}$, ϵ_i, n_i, and $\hat{h}_k$ be defined as above, and assume that $n_1 \geq d/2$. For any δ such that $0 < \delta < 1$,, with probability at least $1 - \delta$, for any $k \in \{1, \ldots, K\}$*

$$\begin{aligned} e(f, \hat{h}_k) \leq & \min_{h\in\mathcal{H}}\{e(f, h)\} + 2\sum_{i=1}^{k}(\frac{n_i}{n_{1:k}})\epsilon_i \\ & + \sqrt{\frac{32(d \ln(2en_{1:k}/d) + \ln(8K/\delta))}{n_{1:k}}}. \end{aligned} \tag{7}$$

In (noise-free) regression with squared loss, assume that the target model f is any function from an input class $\mathcal{X}$ into some bounded subset of $\mathbb{R}$. The loss function is $\mathcal{L}(h(x), y) = (y - h(x))^2$. The expected loss is $e(g_1, g_2) = E_{x\sim D}[\mathcal{L}(g_1(x), g_2(x))] = E_{x\sim D}[(g_1(x) - g_2(x))^2]$. The following lemma states that the 2-triangle inequality holds for regression.

Lemma 2. *Given any three functions $g_1, g_2, g_3 : \mathcal{X} \to \mathbb{R}$, a fixed and unknown distribution P on the inputs $\mathcal{X}$, and the expected loss $e(g_1, g_2) = E_{x\sim D}[(g_1(x) -$*

$g_2(x))^2]$,

$$e(g_1, g_2) \leq 2(e(g_1, g_3) + e(g_3, g_1)). \tag{8}$$

With Lemma 2 and Theorem 1, the bound for regression with squared loss is given as follows.

Theorem 3. *Let $\mathcal{F}$ be the set of functions from $\mathcal{X}$ into $[-B, B]$ and $\mathcal{H} \in \mathcal{F}$. Let e be the expected square loss. Let $K, f = f_1, f_2, \ldots, f_K \in \mathcal{F}$, ϵ_i, n_i, and $\hat{h}_k$ be defined as above. For any δ such that $0 < \delta < 1$, with probability at least $1 - \delta$, for any $k \in \{1, \ldots, K\}$*

$$\begin{aligned} e(f, \hat{h}_k) \leq & 4 \min_{h \in \mathcal{H}} \{e(f, h)\} + 6 \sum_{i=1}^{k} (\frac{n_i}{n_{1:k}}) \epsilon_i \\ & + 32 B R_{n_{1:k}}(\mathcal{H}) + 16 B^2 \sqrt{\frac{2 \ln(4K/\delta))}{n_{1:k}}}, \end{aligned} \tag{9}$$

where $R_n(\mathcal{H})$ is the Rademacher complexity for n observations.

Mansour et al. [28] introduced two types of combinations for multi-source domain adaptation: a linear combining rule and a distribution weighted combining rule. The first rule is based on a parameter $z \in \Delta$ and the target hypothesis H_T is set to $H_T = \sum_{j=1}^{N} z_j H_j$. The second rule is also based on a parameter $z \in \Delta$ but sets the target hypothesis H_T to be $H_T = \sum_{j=1}^{N} \frac{z_j D_j}{\sum_{i=1}^{N} z_i D_i} H_j$ where $\sum_{i=1}^{N} z_i D_i > 0$. When the target mixture distribution is known, they showed that the natural and widely used convex combining rule can perform poorly. Any such convex combination would expect a classification error of $\frac{1}{2}$, even when the source hypotheses make no error on their respective domains. A detailed example was given in [28]. On the other hand, suppose target distribution $D_T = \sum_{j=1}^{N} \lambda_j D_j$. There exists a distribution weighted combination with parameter λ whose loss $\mathcal{L}(D_\lambda, H_\lambda, f)$ is at most ϵ with respect to any mixture adaptation problem, where H_λ represents the distribution weighted combining rule with parameter λ. To show this,

$$H_\lambda = \sum_{j=1}^{N} \frac{\lambda_j D_j}{\sum_{i=1}^{N} \lambda_i D_i} H_j = \sum_{j=1}^{N} \frac{\lambda_j D_j}{D_T} H_j. \tag{10}$$

Suppose the loss function $\mathcal{L}$ is convex with respect to the first argument, the loss of H_λ with respect to D_T and a target $f \in \mathcal{F}$ can be bounded as

$$\begin{aligned}
\mathcal{L}(D_T, H_\lambda, f) &= \sum_{x \in X} \mathcal{L}(H_\lambda(x), f(x)) D_T(x) \\
&\leq \sum_{x \in X} \sum_{j=1}^{N} \lambda_j D_j(x) \mathcal{L}(H_j(x), f(x)) \\
&= \sum_{j=1}^{N} \lambda_j \epsilon_j \leq \epsilon,
\end{aligned}$$

where $\epsilon_j := \mathcal{L}(D_j, H_j, f) \leq \epsilon$. With unknown target mixture distribution D_T, for any fixed target function f, according to the Brouwer fixed point theorem, there exists a distribution weighted combining rule that has a loss at most ϵ with respect to any mixture D_T.

Ben-David et al. [3] gave two learning bounds for empirical risk minimization. The first one considers the quality and quantity of data available from each source individually, regardless of the relationships between sources. The second bound depends directly on the $\mathcal{H}\Delta\mathcal{H}$-distance between the target and the weighted combination of source domains. Firstly, the definition of $\mathcal{H}\Delta\mathcal{H}$-distance is given as follows.

Definition 3. *For a hypothesis space $\mathcal{H}$, the symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$ is the set of hypotheses*

$$\mathcal{H}\Delta\mathcal{H} = \{h(x) \oplus h'(x) : h, h' \in \mathcal{H}\} \tag{11}$$

where $\oplus$ is the **XOR** *function. In other words, every hypothesis $g \in \mathcal{H}\Delta\mathcal{H}$ is the set of disagreements between two hypotheses in $\mathcal{H}$.*

By the definition, a distance $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T))$ between two distributions is defined:

$$d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) = 2 \sup_{h, h' \in \mathcal{H}} |\epsilon_S(h, h') - \epsilon_T(h, h')|. \tag{12}$$

It is easy to prove that the following inequality holds for any hypotheses $h, h' \in \mathcal{H}$, which is useful in the following theorems,

$$|\epsilon_S(h, h') - \epsilon_T(h, h')| \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}. \tag{13}$$

Suppose there are N distinct sources. Each source S_j is associated with an unknown distribution $\mathcal{D}_j$ over input points and an unknown labeling function f_j. Let vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ denote domain weights with $\sum_{j=1}^{N} \alpha_j = 1$.

Considering the pairwise $\mathcal{H}\Delta\mathcal{H}$-distance between each source and the target, the bound in [3] gives a trade-off between minimizing the average divergence of the target data and weighting all points equally to encourage faster convergence. The bound is given below.

Theorem 4. *Let $\mathcal{H}$ be a hypothesis space of VC dimension d. For each $j \in \{1, \ldots, N\}$, let S_j be a labeled sample of size $\beta_j m$ generated by drawing $\beta_j m$ points from D_j and labeling them according to f_j. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_{\boldsymbol{\alpha}}(h)$ for a fixed weight vector $\boldsymbol{\alpha}$ on these samples, $\hat{\epsilon}_{\boldsymbol{\alpha}} = \sum_{j=1}^{N} \alpha_j \epsilon_j(h)$ $= \sum_{j=1}^{N} \frac{\alpha_j}{m_j} \sum_{x \in S_j} |h(x) - f_j(x)|$. Let h_T^* $=\min_{h \in \mathcal{H}} \epsilon_T(h)$ be the target error minimizer, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$\begin{aligned} \epsilon_T(\hat{h}) \leq & \epsilon_T(h_T^*) + 2\sqrt{\left(\sum_{j=1}^{N} \frac{\alpha_j^2}{\beta_j}\right)\left(\frac{d\log(2m) - \log(\delta)}{2m}\right)} \\ & + \sum_{j=1}^{N} \alpha_j (2\lambda_j + d_{\mathcal{H}\Delta\mathcal{H}}(D_j, D_T)), \end{aligned} \tag{14}$$

where $\lambda_j = \min_{h \in \mathcal{H}} \{\epsilon_T(h) + \epsilon_j(h)\}$, and $\epsilon_T(h)$ reflects the probability according to the distribution D_T that a hypothesis h disagrees with a labeling function f_T and is defined as $\epsilon_T(h) = E_{x \sim D_T}[|h(x) - f_T(x)|]$.

For the bound in Theorem 4, it is not necessary to have a single hypothesis that is good for every source domain because divergence between domains is measured only between each source domain and the target domain. However, the domain structure is ignored when calculating unlabeled divergence. An alternate bound is given in the next theorem, which allows to alter the source distribution by changing $\boldsymbol{\alpha}$.

Theorem 5. *Let $\mathcal{H}$ be a hypothesis space of VC dimension d. For each $j \in \{1, \ldots, N\}$, let S_j be a labeled sample of size $\beta_j m$ generated by drawing $\beta_j m$*

points from D_j and labeling them according to f_j. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_{\boldsymbol{\alpha}}(h)$ for a fixed weight vector $\boldsymbol{\alpha}$ on these samples and h_T^* $=\min_{h\in\mathcal{H}} \epsilon_T(h)$ is the target error minimizer, then for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\begin{aligned} \epsilon_T(\hat{h}) \leq & \epsilon_T(h_T^*) + 4\sqrt{\left(\sum_{j=1}^{N} \frac{\alpha_j^2}{\beta_j}\right)\left(\frac{d\log(2m) - \log(\delta))}{2m}\right)} \\ & + 2\gamma_{\boldsymbol{\alpha}} + d_{\mathcal{H}\Delta\mathcal{H}}(D_{\boldsymbol{\alpha}}, D_T), \end{aligned} \tag{15}$$

where $\gamma_{\boldsymbol{\alpha}} = \min_h\{\epsilon_T(h) + \epsilon_{\boldsymbol{\alpha}}(h)\} = \min_h\{\epsilon_T(h) + \sum_{j=1}^{N} \alpha_j \epsilon_j(h)\}$.

In Theorem 5, a hypothesis h^* is demanded to exist which has low error on both the $\boldsymbol{\alpha}$-weighted convex combination of sources and the target domain. Instead of measuring the $\mathcal{H}\Delta\mathcal{H}$-divergence between the target and each source domain, the bound measures the divergence between the target and a mixture of sources, which may be significantly tighter.

3. Well-developed algorithms

In this section, we investigate some well developed algorithms for multi-source domain adaptation. We categorize the algorithms into two groups of approaches, i.e., feature representation approaches and combination of prelearned classifiers.

As we mentioned before, the source domains and the target domains are different, and certain features among the domains are domain-specific while others are common. Therefore, there may exist mappings from the original feature spaces to a latent feature space that is shared between domains. Feature representation approaches [8] for domain adaptation change the feature representations to better describe shared characteristics among the domains. This kind of approach aims to make the source and target domain distributions similar, either by penalizing or removing features whose statistics vary between domains or by learning a feature space embedding or projection in which a distribution divergence statistic is minimized.

The second group of approaches introduced here utilize prelearned classifiers trained both on the source domains and (if any) the target domain. These prelearned classifiers are weighted and combined to obtain a final classifier for the target domain. The core issue here is how to assign a weight for each prelearned classifier according to the relationship between the source domain and the target domain.

3.1. Feature representation approaches

Chattopadhyay et al. [29] introduced a framework called conditional probability based multi-source domain adaptation from the smoothness assumption on the probability distribution of the target domain data. Assume that there are M source domains with plenty of labeled samples in each source domain, and the target domain consists of plenty of unlabeled data D_u^T with sample size n_u and a few labeled data D_l^T with sample size n_l. The framework assigns different weights to different source domains based on the conditional probability differences. Denote the weight factor by β^s for the sth source domain, which is learned on the unlabeled target domain samples as follows. For each source, a hypothesis h^s is learned. Using these M source hypotheses to predict the unlabeled target domain data D_u^T gets a $n_u \times M$ matrix H^S with each row of H^S given by $H_i^S = [h_i^1, h_i^2, \ldots, h_i^M]$. H_i^S is a vector consisting of the predicted labels of M source hypotheses for the ith sample of target domain data. Let $\beta = [\beta^1, \ldots, \beta^M]$ be the weight vector, and the optimization for β can be done by minimizing the difference in predicted labels between two nearby points in the target domain using the following objective:

$$\min_{\beta} \sum_{i,j=1}^{n_u} (H_i^S \beta H_j^S \beta)^2 W_{ij} \tag{16}$$

where H^S is defined above, and W_{ij} is the similarity between the two target domain data samples.

Optimizing Eq. 16 enforces that nearby points in the marginal distribution of the target data have similar conditional probabilities, and the proposed weighting scheme is likely to give higher weights to those sources which have similar conditional probability distributions to the target data. The multi-source domain adaptation framework is given as follows:

$$\min_{f^T \in H_K} \gamma_A \left\|f^T\right\|_K^2 + \frac{1}{n_l}\sum_{i=1}^{n_l}(f_i^T - y_i^T)^2 + \frac{\theta}{2n_u}\sum_{j=n_l+1}^{n_T}\left\|f_j^T - \sum_{s=1}^{M}\beta^s f_j^s\right\| + \frac{\gamma_I}{n_T^2}(f^T)^\top L f^T \tag{17}$$

where the first term controls the complexity of the classifier f^T in the reproducing kernel Hilbert space H_K, and γ_A controls the penalty factor. The second term is the empirical error of f^T on the few labeled target domain data D_l^T, where f_i^T is the ith label predicted by f^T, and y_i^T is the ground truth. The third term is the empirical error on the unlabeled target data, which are labeled using the conditional-probability-based weighting scheme. $\sum_{s=1}^{M}\beta^s f_j^s$ is the estimated label for the unlabeled target data x_j based on the M source domain classifiers f^s. The fourth term is a manifold-based regularizer based on the smoothness assumption on the target domain data, where L is the graph Laplacian associated with the target domain data $D^T = D_u^T \cup D_l^T$ and $n_T = n_u + n_l$. γ_I controls the importance of f^T in the intrinsic geometry of the data marginal probability of x.

Based on the work in [29], Sun et al. [30] proposed a two-stage domain adaptation methodology for multi-source problems. Following the settings in [29] the source data samples in [30] are re-weighted in the first stage based on the marginal probability differences as

$$\begin{aligned} &\min_{\alpha^s} \left\| \frac{1}{n_s}\sum_{i=1}^{n_s}\alpha_i^s \Phi(x_i^s) - \frac{1}{n_T}\sum_{i=1}^{n_T}\Phi(x_i^t)\right\|_H^2 \\ &\text{s.t.} \quad \alpha_i^s \geq 0, \end{aligned} \tag{18}$$

where $\Phi(x)$ is a feature map onto a reproducing kernel Hilbert space H, n_s is the number of samples in the sth source domain, $n_T = n_l + n_u$ is the sample size of the target domain, and α^s is the n_s dimensional weight vector. In the second stage, the source domains are reweighted using the weighting scheme introduced in [29]. After the two stage weighting, the target classifier can be learned based on the reweighted source data and a few labeled target domain data. Formally, the classifier for the target domain is given as

$$f^* = \arg\min_{h} \mu \sum_{s=1}^{M} \frac{\beta^s}{n_s} \sum_{i=1}^{n_s} \alpha_i^s \mathcal{L}(h(x_i^s), y_i^s) + \sum_{j=1}^{n_l} \frac{1}{n_l} \mathcal{L}(h(x_j^T), y_j^t). \qquad (19)$$

Duan et al. [31, 10, 32] introduced a data-dependent regularizer into the objective of SVR (support vector regression) using the ϵ-insensitive loss. The regularizer is defined below:

Definition 4. *Let $D_l^T \cup D_u^T$ be the target domain, where $D_l^T = \{(x_1, y_1), \ldots, (x_{n_l}^T, y_{n_l}^T)\}$ and $D_u^T = x_{n_l+1}^T, \ldots, x_{n_T}^T$. Let D_j be the jth source domain, where $D^j = \{(x_i^j, y_i^j)|_{i=1}^{n_j}\}$, y_i^j is the label of x_i^j, $j = 1, \ldots, M$, and M is the total number of source domains. The data-dependent regularizer for domain adaptation is defined as:*

$$\Omega(f_u^T) = \frac{1}{2} \sum_{j=1}^{M} \gamma_j \sum_{i=n_l+1}^{n_T} (f^T(x_i^T) - f^j(x_i^T))^2 \qquad (20)$$

where $f^T(x_i^T)$ is the decision value of x_i^T form the target classifier, $f^j(x_i^T)$ is the decision value of x_i^T form the jth source classifier, and γ_j is a pre-defined weight for measuring the relevance between the jth source domain and the target domain.

In this definition, when γ_j is large, it means that the jth source domain and the target domain are relevant, so $f^j(x_i^T)$ should be close to $f^T(x_i^T)$. It can be regarded as the vector β in [29], and can be prelearned based on the Laplacian graph of the unlabeled target samples. An alternative to define γ_j is to use the so called maximum margin discrepancy (MMD) criterion proposed in [41].

In [31, 32], the authors proposed a method which simultaneously minimize the structural risk functional of least-squares SVM as well as the data-dependent regularizer defined above. The method, named domain adaptation machine (DAM), is formulated as

$$\min_{f^T} \Omega(f^T) + \frac{1}{2} \sum_{i=1}^{n_l} (f_i^t - y_i^t)^2 + \Omega(f_u^T), \qquad (21)$$

where $\Omega(\cdot)$ is the regularizer, and the second term is the empirical error of the target classifier f^T on the target labeled samples D_l^T. To solve this optimization problem, using the ϵ-insensitive loss function, the authors gave a sparse solution for the problem (referred as FastDAM in [32]). Assume that the regularizer $\Omega(f^T) = \frac{1}{2\theta} \|\mathbf{w}\|^2$ for the penalty of function complexity of f^T. The optimization problem in Eq. 21 is then rewritten as:

$$\min_{f^T,\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 + C\sum_{i=1}^{n_T} l_\epsilon(\mathbf{w}^\top \phi(x_i) + b - f_i^T) + \theta(\frac{1}{2}\sum_{i=1}^{n_l}(f_i^t - y_i^t)^2 + \Omega(f_u^T)) \quad (22)$$

where C is another tradeoff parameter to control the difference between $f^T(x)$ and $\mathbf{w}^\top \phi(x_i)+b$, θ is a tradeoff parameter to control the empirical error from the target domain as well as the smoothness regularizer, and $l_\epsilon(t)$ is the ϵ-insensitive loss: $l_\epsilon(t) = \begin{cases} |t| - \epsilon, & if\ |t| > \epsilon \\ 0, & otherwise. \end{cases}$ Since the ϵ-insensitive loss is non-smooth,

275 Eq. 22 is usually transformed as a constrained optimization problem as:

$$\begin{aligned} &\min_{f^T,\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 + C\sum_{i=1}^{n_T}(\xi_i + \xi_i^*) + \theta(\frac{1}{2}\sum_{i=1}^{n_l}(f_i^t - y_i^t)^2 + \Omega(f_u^T)) \\ &\text{s.t.} \quad \begin{cases} \mathbf{w}^\top \phi(x_i) + b - f_i^T \leq \epsilon + \xi_i, \xi_i \geq 0, \\ f_i^T - \mathbf{w}^\top \phi(x_i) - b \leq \epsilon + \xi_i^*, \xi_i^* \geq 0, \end{cases} \end{aligned} \quad (23)$$

where ξ_i's and ξ_i^*'s are slack variables for the ϵ-insensitive loss.

The DAM in [29, 31] uses all the source domains for multi-source domain adaptation. The only difference among the sources is the different weight assigned to each source, which reflects the relevance between the source domain and the target domain. But in practice it may be more beneficial to choose a few relevant source domain rather than use all of them. To this end, Duan et al. [10] proposed a data-dependent regularizer for domain selection, which is an extension to Definition 4:

$$\Omega(f_u^T) = \frac{1}{2}\sum_{j=1}^{M} d_j \sum_{i=n_l+1}^{n_T} (f^T(x_i^T) - f^j(x_i^T))^2 \quad (24)$$

where $d_j \in \{0, 1\}$ is a domain selection indicator for the jth source domain. Some selected source domains share a similar decision boundary with the target domain, thus avoiding the negative adaptation.

3.2. Combination of prelearned classifiers

When some labeled target data are available, Schweikert et al. [33] proposed a classification method for multi-source domain adaptation by combining the prelearned source classifiers and target classifier through a simple weight scheme. Using a so-called multiple convex combination, the final classifier $f(x)$ for the target domain is formulated as:

$$f(x) = \alpha f^T(x) + \frac{1-\alpha}{M} \sum_{k=1}^{M} f^k(x), \tag{25}$$

where α balances the two terms, M is the number of source domains, and f^k is the pre-computed source classifier using the labeled training data from the kth source domain, and f^T is the target classifier obtained by learning an SVM classifier using the labeled training samples from the target domain.

The multi-source domain adaptation framework proposed by Sun and Shi [34] combines the prelearned source classifiers to obtain a good target classifier. There is no need for the labeled target data to exist in their framework. The framework is based on the Bayesian learning principle: the probability of which class a target example belongs to is proportional to the product of prior and likelihood assigned to this example. Suppose there are M source domains S^s with pre-defined source classifier f^s, $s = \{1, \ldots, M\}$, and the target domain T. The target domain only has plenty of unlabeled samples. The prior is constructed with the Laplacian matrix on the unlabeled target data as follows

$$prior^s = \frac{1}{\sum_{i,j=1}^{n_u} (y_i^s - y_j^s)^2 W_{ij}} = \frac{1}{2(Y^s)^\top L Y^s}, \tag{26}$$

where n_u is the size of the target unlabeled samples, $Y^s = \{y_1, \ldots, y_{n_u}\}$, y_i^s is the predicted labels for the ith sample x_i in the target domain, W is the weight matrix where W_{ij} measures the closeness between x_i and x_j, and L is the Laplacian matrix constructed on the unlabeled target domain. The different prior of each source shows the fitness between each source domain and the target domain. The bigger the prior is, the better the corresponding source classifier is. For the likelihood, the authors employed the mean Euclidean distance of the

k-nearest neighbors of instance x_i to reflect the similarity between the target domain and the source domain. It is intuitive that the higher probability of the instance from the target domain occurring in the source domain, the better the source classifier is. For each instance x_i in the target domain, the probability of it occurring in the source domain S^s is inversely proportional to the the mean Euclidean distance of the k-nearest neighbors in source domain S^s, so the likelihood that x_i occurs in S^s is defined as

$$Like_i^s = \frac{k}{\sum_j \left\| x_i - x_j^s \right\|} \tag{27}$$

where x_j^s, which comes from the sth source, is among the k-nearest neighbors of x_i. After the prior and the likelihood are defined, the posterior is proportional to the product of the prior and the likelihood:

$$post_i^s \propto prior_i^s \times Like_i^s \tag{28}$$

where $post_i^s$ is the posterior of x_i based on the sth source domain. The posteriors are used to weight the source classifiers.

Yang et al. [12] explored classifier adaptation techniques based on support vector machines. A "delta function" in the form of $\delta f(x) = w^\top \phi(x)$ is added into the standard SVM (support vector machine) objective, and they got the adaptive support vector machine (A-SVM) as: $f(x) = f^S(x) + \delta f(x) = f^S(x) + w^\top \phi(x)$, where f^S is the pre-computed classifier from the source domain. For multi-source settings, the adapted classifier can be extended as

$$f(x) = \sum_{k=1}^{M} \alpha_k f_k^S(x) + \delta f(x) = \sum_{k=1}^{M} \alpha_k f_k^S(x) + w^\top \phi(x), \tag{29}$$

where $\alpha_k \in (0,1)$ is the weight of each source classifier $f_k^S(x)$ and $\sum_{k=1}^{M} \alpha_k = 1$. Once the model is learned from the source domains, the prediction process does not involve any source domain data, so the A-SVM model is efficient for online application. In [12], equal weights were used for all source classifiers in the experiments.

Tu and Sun [35] proposed a multi-source domain adaptation framework called cross-domain representation learning framework, which combines class-

Algorithm 1: **Cross-domain representation-learning framework**

Input: Datasets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_{n_s}$ from different domains, where example $X_i \in \mathbb{R}^d$ $(i = 1, 2, \ldots, n)$ is the raw data representation. Label information of labeled subsets. Domain information of $X_1, X_2, \ldots, X_n$.

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

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

Keys: 1) How to define the $Q_d(F_T)$ by modifying $Q_c(F_T)$. 2) How to define the combination of $Q_d(F_T)$ and $Q_c(F_T)$.

separate objectives and domain-merge objectives simultaneously to learn a data representation model. The learned model can catch data characteristics which are not only helpful for later tasks but also robust to the domain differences. The framework takes into consideration both maximizing the differences between classes and minimizing the difference between domains. Here, maximizing the differences between classes can be considered as the class-separate objective for the classification task, and minimizing the difference between domains can be naturally regarded as one of the domain-merged objectives. The framework is described in Algorithm 1. Three implementations are proposed in [35]: domain-merge and class-separate correlation feature selection (DMCS_CFS), domain-merge and class-separate Fisher discriminant analysis (DMCA_FDA), domain-merge and class-separate pairwise constraints based distance metric learning (DMCS_PCDML).

Recently, Xu and Sun introduced transfer learning methods with multi-view adaboost for both single-source and multi-source domain adaptation [36, 37]. When there are only one source domain [37], the multi-view transfer learning method with adaboost is described as follows. Suppose there are n labeled target samples $X = (x_1, y_1), \ldots, (x_n, y_n)$, and m labeled source examples

$S = (x_{n+1}, y_{n+1}), \ldots, (x_{n+m}, y_{n+m})$. To use the adaboost method, initialize a weight vector $W_t \in R^{m+n}$ for the samples in the source and target domain, $t = \{1, \ldots, T\}$ where T is the iteration number. All of the features are divided into to views V_1 and V_2. In the tth iteration, two weaker learners from these two views h_t^{V1} and h_t^{V2} are learned . Then the empirical accuracies on the target samples as well as a proposed parameter $agree_t$ are used to update the weight vector. The parameter $agree_t$ indicates the percentage of the samples predicted to be the same class by the two classifiers h_t^{V1} and h_t^{V2}. In [36], this method was extended to the multi-source setting.

Another transfer learning framework proposed by Xu and Sun is called part-based transfer learning [38, 39]. Since many collections of data consist of a number of parts of factors which contain different information of the data, learning through different parts can get more various kinds of knowledge for the task (classification, regression, etc.).

In [39], a transfer learning method for one source domain and one target domain based on partly transfer learning was proposed, namely part-based transfer learning. Suppose one source domain with n labeled training samples and one target domain with m labeled training samples and large numbers of unlabeled test samples are given. The feature spaces are the same for both the source and target domains. The method is described as follows. Firstly, divide the source and target feature spaces into N parts according to certain criteria. For example, suppose the dimension of the feature space is ten and the feature space is divided into three parts. One alternative way is to split the first nine features into three parts and add the last feature into each part to create three interdependent four-dimensional parts.

After the division, N part-based source classifiers can be trained on the N source parts, and simultaneously their optimal model parameters are got. Then, train N part-based target classifiers for the target training samples using the N optimal model parameters obtained by training the N part-based source classifiers accordingly. The empirical classification accuracies of the N part-based target classifiers are used to weight the N target classifiers to obtain the final target classifier for the target domain. This step transfers the source information to the target domain by the optimal model parameters obtained in the source training step.

When given multiple sources, the method in [39] can be easily extended to the multi-source domain adaptation setting. Suppose there are M source domains and a target domain. The feature space can be divided into N parts. For each source domain, N part-based source classifiers as well as the optimal model parameters are obtained. Therefore, for each part of the target domain, M part-based target classifiers can be trained according to the M corresponding optimal model parameters transferred from the part-based source classifiers. These M part-based target classifiers are then weighted according to the empirical classification accuracies to form the ensemble part-based target classifier. Thus N ensemble part-based target classifiers can be trained. Finally, such N ensemble part-based target classifiers are weighted according to their accuracies to gain the final target classifier trained from the M source domains and the target training domain. In [38, 39], the parameters learned from the source domain(s) were transferred directly to the target domain. This may not be suitable in many complex situations because of the diversity between the source domains and the target domain. To this end, using the parameters learned from the source domain to initialize the parameters for the target classifier is an alternative and might be more suitable for transferring. After this initialization, some learning algorithms such as neural networks can be used to update them until getting a satisfying solution.

4. Model Evaluation

4.1. Performance measurements

The most popular performance measurement in the classification task for multi-source domain adaptation is the classification accuracy [26, 5, 11, 30, 34]. Let h and f be the labeling functions that map the target unlabeled data points to their true labels and prediction labels, respectively. Then the classification accuracy is defined as

$$Accu = \frac{|\{\mathbf{x}|\mathbf{x} \in D_T \wedge h(\mathbf{x}) = f(\mathbf{x})\}|}{|D_T|}, \tag{30}$$

where $\mathbf{x}$ is the data point and D_T is the target domain. Intuitively, the higher the $Accu$ is, the better the target classifier is.

In regression, the mean squared error (MSE), which measures the average of the squares of the "errors" between the estimator and what is estimated, is often used for performance measurement [28]. Let $\hat{Y}$ be a vector of n predictions, and Y be the vector of true values. Then the MSE of the predictor is defined as

$$MSE = \sum_{i=1}^{n} \frac{1}{n}(\hat{Y} - Y)^2. \tag{31}$$

4.2. Public datasets

In this subsection, some publicly available datasets are listed in Table 2 for the convenience of use by other researchers: sentiment classification dataset, 20 newsgroups dataset, TRECVID dataset, email spam dataset, objective recognition dataset, and wifi dataset. Methods evaluated on each dataset are also listed in the table and researchers can use them for comparison in their new research. We test each URL in the table and make sure that the datasets can be downloaded through the URLs.

5. Conclusions and open problems

In this survey, we presented a theoretical investigation as well as some well-developed algorithms for multi-source domain adaptation.

In the future, several important research issues need to be considered. Firstly, it would be interesting to investigate algorithms that choose a convex combination of multiple sources to minimize the bound in [3] as possible approaches to adaptation from multiple sources. In addition, the error bound given in [28] is defined on a fixed target distribution. Therefore, investigating the possibility of using an arbitrary target distribution to generalize the error bound is also an interesting problem to be considered.

Table 2: Publicly available datasets for multi-source domain adaptation.

Dataset	References	Source	Brief description
Sentiment Classification dataset	Ben-David et al. [3] Mansour et al. [28, 7] Sun et al. [30] Sun and Shi [34] Tu and Sun [35]	`http://www.cs.jhu.edu/~mdredze/datasets/sentiment`	Sentiment classification data consist of product reviews from several different product types taken from Amazon.com.
20 Newsgroups dataset	Sun et al. [30] Duan et al. [32]	`http://www.qwone.com/~jason/20Newsgroups`	The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.
TRECVID dataset	Duan et al. [31, 32] Yang et al. [12]	`http://www-nlpir.nist.gov/projects/trecvid`	The TRECVID dataset contains 61901 key frames extracted from 108 hours of video programs form six different broadcast channels, including three English channels, two Chinese channels, and one Arabic channel.
Email Spam dataset	Duan et al. [32]	`http://www.ecmlpkdd2006.org/challenge.html`	The email spam dataset contains a set of 4000 publicly available labeled emails as well as three email sets (each has 2500 emails) annotated by three different users.
Objection Recognition dataset	Kulis et al. [11]	`https://www.eecs.berkeley.edu/~jhoffman/domainadapt/`	This dataset contains images from 31 object categories and three domains. These three domains differ from each other by the objective pose, image background, and resolution, but have the same objective categories.
Wifi dataset	Pan et al. [20]	`http://www.cse.ust.hk/~qyang/ICDMDMC07/`	The data were collected and organized at Hong Kong University of Science and Technology and used for indoor location estimation.

Secondly, how can prior knowledge about domain similarity be included in the learning procedure? The work by Sun and Shi [34] steps forward by proposing a prior based on the Laplacian matrix. A nature question to ask is thus if there is other types of appropriate priors. More effective algorithmic implementations for the Bayesian framework are interesting research topics. In addition, for the part-based methods, the authors used a simple criterion to divide the feature space. In the future, more heuristic criteria could be explored to generate the part models.

Thirdly, most existing domain adaptation algorithms assume that the feature space between the source domains and the target domains is the same. But, in many applications, the feature spaces among domains may be different. Such problem are gaining more and more interests and need further explorations.

Finally, so far, many domain adaptation techniques have been used to applications with a limited variety, such as text classification and image classification problems. Only a little work has considered video classification [18, 10]. In the future, more work on other challenging applications is needed, such as brain-computer interface signal classification [42, 43], video and speech based applications [30], activity detection, social network analysis, and logical inference [8].

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Project 61370175, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

[1] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, Lecture Notes in Computer Science 1398 (1998) 137–142.

[2] J. Shawe-Taylor, S. Sun, A review of optimization methodologies in support vector machines, Neurocomputing 74 (17) (2011) 3609–3618.

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, Machine Learning 79 (2010) 151–175.

[4] H. Daumé III, Frustratingly easy domain adaptation, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, pp. 256–263.

[5] M. Dredze, A. Kulesza, K. Crammer, Multi-domain learning by confidence-weighted parameter combination, Machine Learning 79 (1-2) (2010) 123–149.

[6] Y. Mansour, M. Mogri, A. Rostamizadeh, Domain adaptation: Learning bounds and algorithms, arXiv preprint arXiv:0902.3430, 2009.

[7] Y. Mansour, M. Mohri, A. Rostamizadeh, Multiple source adaptation and the Rényi divergence, in: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009, pp. 367–374.

[8] S. Pan, Y. Qiang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359.

[9] C. Arndt, Information measures: Information and its description in science and engineering, Springer, 2001.

[10] L. Duan, D. Xu, S. Chang, Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1338–1345.

[11] B. Kulis, K. Saenko, T. Darrell, What you saw is not what you get: Domain adaptation using asymmetric kernel transforms, in: Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1785–1792.

[12] J. Yang, R. Yan, A. Hauptmann, Cross-domain video concept detection using adaptive svms, in: Proceedings of the 15th International Conference on Multimedia, 2007, pp. 188–197.

[13] A. Aue, M. Gamon, Customizing sentiment classifiers to new domains: A case study, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing, 2005, pp. 1–7.

[14] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 440–447.

[15] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: Sentiment classification using machine learning techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2002, pp. 79–86.

[16] W. Tu, S. Sun, Dynamical ensemble learning with model-friendly classifiers for domain adaptation, in: Proceedings of the 21st International Conference on Pattern Recognition, 2012, pp. 1181–1184.

[17] J. Jiang, C. Zhai, Instance weighting for domain adaptation in NLP, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 264–271.

[18] L. Duan, I. Tsang, D. Xu, S. Maybank, Domain transfer SVM for video concept detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1375–1381.

[19] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, Q. He, Transfer learning from multiple source domains via consensus regularization, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, 2008, pp. 103–112.

[20] S. Pan, V. Zheng, Q. Yang, D. Hu, Transfer learning for wifi-based indoor localization, in: Proceedings of the Workshop on Transfer Learning for Complex Task of the 23rd AAAI Conference on Artificial Intelligence, 2008, pp. 43–48.

[21] J. R. Moreno-Torres, T. Raeder, R. O. Alaiz-RodríGuez, N. V. Chawla, F. Herrera. A unifying view on dataset shift in classification, Pattern Recognition, 45(2010) 521–530.

[22] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, Advances in Neural Information Processing Systems 19 (2007) 137–144.

[23] M. Dredze, K. Crammer, Online methods for multi-domain learning and adaptation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2008, pp. 689–697.

[24] C. Chelba, A. Acero, Adaptation of maximum entropy capitalizer: Little data can help a lot, Computer Speech & Language 20 (4) (2006) 382–399.

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

[26] J. Blitzer, R. McDonald, F. Pereira, Domain adaptation with structural correspondence learning, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2006, pp. 120–128.

[27] K. Crammer, M. Kearns, J. Wortman, Learning from multiple sources, Journal of Machine Learning Research 9 (2008) 1757–1774.

[28] Y. Mansour, M. Mohri, A. Rostamizadeh, Domain adaptation with multiple sources, Advances in Neural Information Processing Systems 21 (2009) 1041–1048.

[29] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, J. Ye, Multisource domain adaptation and its application to early detection of

fatigue, ACM Transactions on Knowledge Discovery from Data 6 (4) (2012) Article No. 18.

[30] Q. Sun, R. Chattopadhyay, S. Panchanathan, J. Ye, A two-stage weighting framework for multi-source domain adaptation, Advances in Neural Information Processing Systems 24 (2011) 505–513.

[31] L. Duan, I. W. Tsang, D. Xu, T. S. Chua, Domain adaptation from multiple sources via auxiliary classifiers, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 289–296.

[32] L. Duan, D. Xu, I. Tsang, Domain adaptation from multiple sources: A domain-dependent regularization approach, IEEE Transactions on Neural Networks and Learning Systems 23 (3) (2012) 504–518.

[33] G. Schweikert, G. Rätsch, C. Widmer, B. Schölkopf, An empirical analysis of domain adaptation algorithms for genomic sequence analysis, Advances in Neural Information Processing Systems 21 (2009) 1433–1440.

[34] S. Sun, H. Shi, Bayesian multi-source domain adaptation, in: Proceedings of the International Conference on Machine Learning and Cybernetics, 2013, pp. 24–28.

[35] W. Tu, S. Sun, Cross-domain representation-learning framework with combination of class-separate and domain-merge objectives, in: Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining, 2012, pp. 18–25.

[36] Z. Xu, S. Sun, Multi-source transfer learning with multi-view adaboost, Lecture Notes in Computer Science 7665 (2012) 332–339.

[37] Z. Xu, S. Sun, Multi-view transfer learning with adaboost, in: Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, 2011, pp. 399–402.

[38] S. Sun, Z. Xu, M. Yang, Transfer learning with part-based ensembles, Lecture Notes in Computer Science 7872 (2013) 271–282.

[39] Z. Xu, S. Sun, Part-based transfer learning, Lecture Notes in Computer Science 6677 (2011) 434–441.

[40] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. Wortman, Learning bounds for domain adaptation, Advances in Neural Information Processing Systems 20 (2008) 129–136.

[41] K. Borgwardt, A. Gretton, M. Rasch, H. Kriegel, B. Schölkopf, A. J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics 22 (14) (2006) 49–57.

[42] S. Sun, J. Zhou, A review of adaptive feature extraction and classification methods for eeg-based brain-computer interfaces, in: Proceedings of the International Joint Conference on Neural Networks, 2014, pp. 1746–1753.

[43] W. Tu, S. Sun, A subject transfer framework for EEG classification, Neurocomputing 82 (2012) 109–116.