# A Literature Survey on Domain Adaptation of Statistical Classifiers

Jing Jiang
jiang4@cs.uiuc.edu

Last modified in March 2008

## Contents



## 1 Introduction

Domain adaptation of statistical classifiers is the problem that arises when the data distribution in our test domain is different from that in our training domain. The need for domain adaptation is prevalent in many real-world classification problems. For example, spam filters can be trained on some public collection of spam and ham emails. But when applied to an individual person's inbox, we may want to "personalize" the spam filter, i.e. to adapt the spam filter to fit the person's own distribution of emails in order to achieve better performance.

Although the domain adaptation problem is a fundamental problem in machine learning, it only started gaining much attention very recently (Daumé III and Marcu, 2006; Blitzer et al., 2006; Ben-David et al., 2007; Daumé III, 2007; Jiang and Zhai, 2007a; Satpal and Sarawagi, 2007; Jiang and Zhai, 2007b; Blitzer

et al., 2008). However, some special kinds of domain adaptation problems have been studied before under different names including class imbalance (Japkowicz and Stephen, 2002), covariate shift (Shimodaira, 2000), and sample selection bias (Heckman, 1979; Zadrozny, 2004). There are also some closely-related but not equivalent machine learning problems that have been studied extensively, including multi-task learning (Caruana, 1997) and semi-supervised learning (Zhu, 2005; Chapelle et al., 2006).

In this literature survey, we review some existing work in both the machine learning and the natural language processing communities related to domain adaptation. The goal of this survey is twofold. First, there have been a number of methods proposed to address domain adaptation, but it is not clear how these methods are related to each other. This survey thus tries to organize the existing work and lay out an overall picture of the domain adaptation problem with its possible solutions. Second, a systematic literature survey naturally reveals the limitations of current work and points out promising directions that should be explored in the future.

Because domain adaptation is a relatively new topic that is still constantly attracting attention, our survey is necessarily incomplete. Nevertheless, we try to cover the major lines of work that we are aware of up to the date this survey is written. This survey will also be updated periodically.

## 2 Notations and Overview

### 2.1 Notations

We first introduce some notations that are needed in the discussion in this survey. We refer to the training domain where labeled data is abundant as the *source* domain, and the test domain where labeled data is not available or very little as the *target* domain. Let $X$ denote the input variable (i.e. an observation) and $Y$ the output variable (i.e. a class label). We use $P(X, Y)$ to denote the true underlying joint distribution of $X$ and $Y$, which is unknown. In domain adaptation, this joint distribution in the target domain differs from that in the source domain. We therefore use $P_t(X, Y)$ to denote the true underlying joint distribution in the target domain, and $P_s(X, Y)$ to denote that in the source domain. We use $P_t(Y)$, $P_s(Y)$, $P_t(X)$ and $P_s(X)$ to denote the true marginal distributions of $Y$ and $X$ in the target and the source domains, respectively. Similarly, we use $P_t(X|Y)$, $P_s(X|Y)$, $P_t(Y|X)$ and $P_s(Y|X)$ to denote the true conditional distributions in the two domains. We use lowercase $x$ to denote a specific value of $X$, and lowercase $y$ to denote a specific class label. A specific $x$ is also referred to as an observation, an unlabeled instance or simply an instance. A pair $(x, y)$ is referred to as a labeled instance. Here, $x \in \mathcal{X}$, where $\mathcal{X}$ is the input space, i.e. the set of all possible observations. Similarly, $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the class label set. Without any ambiguity, $P(X = x, Y = y)$ or simply $P(x, y)$ should refer to the joint probability of $X = x$ and $Y = y$. Similarly, $P(X = x)$ (or $P(x)$), $P(Y = y)$ (or $P(y)$), $P(X = x|Y = y)$ (or $P(x|y)$) and $P(Y = y|X = x)$ (or $P(y|x)$) also refer to probabilities rather than distributions.

We assume that there is always a relatively large amount of *labeled* data available in the source domain. We use $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ to denote this set of labeled instances in the source domain. In the target domain, we assume that we always have access to a large amount of *unlabeled* data, and we use $D_{t,u} = \{x_i^{t,u}\}_{i=1}^{N_{t,u}}$ to denote this set of unlabeled instances. Sometimes, we may also have a small amount of labeled data from the target domain, which is denoted as $D_{t,l} = \{(x_i^{t,l}, y_i^{t,l})\}_{i=1}^{N_{t,l}}$. In the case when $D_{t,l}$ is not available, we refer to the problem as *unsupervised domain adaptation*, while when $D_{t,l}$ is available, we refer to the problem as *supervised domain adaptation*.

### 2.2 Overview

Recently, there have been a number of studies related to domain adaptation. However, the motivating ideas behind these methods are different. To connect the existing work and hence to better understand the problem, in the following sections, we organize the existing work into several categories from our own viewpoint. First, in Section 3, we consider a line of work that is based on instance weighting. In Section 4, we look at some work that bears strong resemblance to semi-supervised learning. In Section 5, we review another line of work that is based on changing the representation of $X$. Section 6 reviews work using Bayesian priors, and Section 7 reviews work related to multi-task learning. In Section 8, ensemble methods for domain adaptation are considered.

The categories are ordered in this way so that methods in Section 3, Section 4 and Section 5 are generally applicable to unsupervised domain adaptation problems, while methods in Section 6 and Section 7 can only handle supervised domain adaptation problems.

## 3 Instance Weighting

One general approach to addressing the domain adaptation problem is to assign instance-dependent weights to the loss function when minimizing the expected loss over the distribution of data. To see why instance weighting may help, let us first briefly review the empirical risk minimization framework for standard supervised learning (Vapnik, 1999), and then informally derive an instance weighting solution to domain adaptation. Let $\Theta$ be a model family from which we want to select an optimal model $\theta^*$ for our classification task. Let $l(x, y, \theta)$ be a loss function. Strictly speaking, we want to minimize the following objective function in order to obtain the optimal model $\theta^*$ for the distribution $P(X, Y)$:

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} P(x, y) l(x, y, \theta).$$

Because $P(X, Y)$ is unknown, we can use the empirical distribution $\tilde{P}(X, Y)$ to approximate $P(X, Y)$. Let $\{(x_i, y_i)\}_{i=1}^N$ be a set of training instances randomly sampled from $P(X, Y)$. We then minimize the following empirical risk in order to find a good model $\hat{\theta}$:

$$\begin{aligned} \hat{\theta} &= \arg\min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \tilde{P}(x, y) l(x, y, \theta) \\ &= \arg\min_{\theta \in \Theta} \sum_{i=1}^{N} l(x_i, y_i, \theta). \end{aligned}$$

Now consider the setting of domain adaptation. Ideally, we want to find an optimal model for the target domain that minimizes the expected loss over the target distribution:

$$\theta_t^* = \arg\min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} P_t(x, y) l(x, y, \theta).$$

However, our training instances, $D_s = \{(x_i^s, y_i^s\}_{i=1}^{N_s}$, are randomly sampled from the source distribution

$P_s(X, Y)$. We can rewrite the equation above as follows:

$$\begin{aligned}
\theta_t^* &= \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \frac{P_t(x,y)}{P_s(x,y)} P_s(x,y) l(x,y,\theta) \\
&\approx \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \frac{P_t(x,y)}{P_s(x,y)} \tilde{P}_s(x,y) l(x,y,\theta) \\
&= \arg\min_{\theta\in\Theta} \sum_{i=1}^{N_s} \frac{P_t(x_i^s, y_i^s)}{P_s(x_i^s, y_i^s)} l(x_i^s, y_i^s, \theta). \qquad (1)
\end{aligned}$$

As we can see, weighting the loss for the instance $(x_i^s, y_i^s)$ with $\frac{P_t(x_i^s,y_i^s)}{P_s(x_i^s,y_i^s)}$ provides a well-justified solution to the domain adaptation problem.

It is not possible to compute the exact value of $\frac{P_t(x,y)}{P_s(x,y)}$ for a pair $(x, y)$, especially because we do not have enough labeled instances in the target domain. Section 3.1 reviews one line of work in which $P_t(X|Y) = P_s(X|Y)$ is assumed, while Section 3.2 reviews another line of work in which $P_t(Y|X) = P_s(Y|X)$ is assumed.

### 3.1 Class Imbalance

One simple assumption we can make about the connection between the distributions of the source and the target domains is that given the same class label, the conditional distributions of $X$ are the same in the two domains. However, the class distributions may be different in the source and the target domains. Formally, we assume that $P_s(X|Y = y) = P_t(X|Y = y)$ for all $y \in \mathcal{Y}$, but $P_s(Y) \neq P_t(Y)$. This difference is referred to as the *class imbalance* problem in some work (Japkowicz and Stephen, 2002).

When this class imbalance assumption is made, the ratio $\frac{P_t(x,y)}{P_s(x,y)}$ that we derived in Equation (1) can be rewritten as follows:

$$\begin{aligned}
\frac{P_t(x,y)}{P_s(x,y)} &= \frac{P_t(y)}{P_s(y)} \frac{P_t(x|y)}{P_s(x|y)} \\
&= \frac{P_t(y)}{P_s(y)}.
\end{aligned}$$

Therefore, we only need to use $\frac{P_t(y)}{P_s(y)}$ to weight the instances. This approach has been explored in (Lin et al., 2002). Alternatively, we can re-sample the training instances from the source domain so that the re-sampled data roughly has the same class distribution as the target domain. In re-sampling methods, under-represented classes are over-sampled, and over-represented classes are under-sampled (Kubat and Matwin, 1997; Chawla et al., 2002; Zhu and Hovy, 2007).

For classification algorithms that directly model the probability distribution $P(Y|X)$ such as logistic regression classifiers, it can be shown theoretically that the estimated probability $P_s(y|x)$ can be transformed into $P_t(y|x)$ in the following way (Lin et al., 2002; Chan and Ng, 2005):

$$P_t(y|x) = \frac{r(y) P_s(y|x)}{\sum_{y'\in\mathcal{Y}} r(y') P_s(y'|x)},$$

where $r(y)$ is defined as

$$r(y) = \frac{P_t(y)}{P_s(y)}.$$

Now we can first estimate $P_s(y|x)$ from the source domain, and then derive $P_t(y|x)$ using $P_s(Y)$ and $P_t(Y)$.

For other classification algorithms that do not directly model $P(Y|X)$, such as naive Bayes classifiers and support vector machines, if $P(Y|X)$ can be obtained through careful calibration, the same trick can be applied. Chan and Ng (2006) applied this method to the domain adaptation problem in word sense disambiguation (WSD) using naive Bayes classifiers.

In practice, one needs to know the class distribution in the target domain in order to apply the methods described above. In some studies, it is assumed that this distribution is known a priori (Lin et al., 2002). However, in reality, we may not have this information. Chan and Ng (2005) proposed to use the EM algorithm to estimate the class distribution in the target domain.

### 3.2 Covariate Shift

Another assumption one can make about the connection between the source and the target domains is that given the same observation $X = x$, the conditional distributions of $Y$ are the same in the two domains. However, the marginal distributions of $X$ may be different in the source and the target domains. Formally, we assume that $P_s(Y|X = x) = P_t(Y|X = x)$ for all $x \in \mathcal{X}$, but $P_s(X) \neq P_t(X)$. This difference between the two domains is called *covariate shift* (Shimodaira, 2000).

At first glance, it may appear that covariate shift is not a problem. For classification, we are only interested in $P(Y|X)$. If $P_s(Y|X) = P_t(Y|X)$, why would the classifier learned from the source domain not perform well on the target domain even if $P_s(X) \neq P_t(X)$? Shimodaira (2000) showed that this covariate shift becomes a problem when *misspecified* models are used. Suppose we consider a parametric model family $\{P(Y|X, \theta)\}_{\theta \in \Theta}$ from which a model $P(Y|X, \theta^*)$ is selected to minimize the expected classification error. If none of the models in the model family can exactly match the true relation between $X$ and $Y$, that is, there does not exist any $\theta \in \Theta$ such that $P(Y|X = x, \theta) = P(Y|X = x)$ for all $x \in \mathcal{X}$, then we say that we have a misspecified model family. The intuition of why covariate shift under model misspecification becomes a problem is as follows. With a misspecified model family, the optimal model we select depends on $P(X)$, and if $P_t(X) \neq P_s(X)$, then the optimal model for the target domain will differ from that for the source domain. The intuitive is that the optimal model performs better in dense regions of $X$ than in sparse regions of $X$, because the dense regions dominate the average classification error, which is what we want to minimize. If the dense regions of $X$ are different in the source and the target domains, the optimal model for the source domain will no longer be optimal for the target domain.

Under covariate shift, the ratio $\frac{P_t(x,y)}{P_s(x,y)}$ that we derived in Equation (1) can be rewritten as follows:

$$\begin{aligned} \frac{P_t(x,y)}{P_s(x,y)} &= \frac{P_t(x)}{P_s(x)} \frac{P_t(y|x)}{P_s(y|x)} \\ &= \frac{P_t(x)}{P_s(x)}. \end{aligned}$$

We therefore want to weight each training instance with $\frac{P_t(x)}{P_s(x)}$.

Shimodaira (2000) first proposed to re-weight the log likelihood of each training instance $(x, y)$ using $\frac{P_t(x)}{P_s(x)}$ in maximum likelihood estimation for covariate shift. It can be shown theoretically that if the support of $P_t(X)$ (the set of $x$'s for which $P_t(X = x) > 0$) is contained in the support of $P_s(X)$, then the optimal model that maximizes this re-weighted log likelihood function asymptotically converges to the optimal model for the target domain.

A major challenge is how to estimate the ratio $\frac{P_t(x)}{P_s(x)}$ for each $x$ in the training set. In some work, a principled method of using non-parametric kernel density estimation is explored (Shimodaira, 2000; Sugiyama

and Müller, 2005). In some other work, it is proposed to transform this density ratio estimation into a problem of predicting whether an instance is from the source domain or from the target domain (Zadrozny, 2004; Bickel and Scheffer, 2007). Huang et al. (2007) transformed the problem into a kernel mean matching problem in a reproducing kernel Hilbert space. Bickel et al. (2007) proposed to learn this ratio together with the classification model parameters.

### 3.3 Change of Functional Relations

Both class imbalance and covariate shift simplify the difference between $P_s(X, Y)$ and $P_t(X, Y)$. It is still possible that $P_t(X|Y)$ differs from $P_s(X|Y)$ or $P_t(Y|X)$ differs from $P_s(Y|X)$.

Jiang and Zhai (2007a) considered the case when $P_t(Y|X)$ differs from $P_s(Y|X)$, and proposed a heuristic method to remove "misleading" training instances from the source domain, where $P_s(y|x)$ is very different from $P_t(y|x)$. To discover these "misleading" training instances, some labeled data from the target domain is needed. This method therefore is only suitable for *supervised* domain adaptation.

## 4 Semi-Supervised Learning

If we ignore the domain difference, and treat the labeled source domain instances as labeled data and the unlabeled target domain instances as unlabeled data, then we are facing a semi-supervised learning (SSL) problem. We can then apply any SSL algorithms (Zhu, 2005; Chapelle et al., 2006) to the domain adaptation problem. The subtle difference between SSL and domain adaptation is that (1) the amount of labeled data in SSL is small but large in domain adaptation, and (2) the labeled data may be noisy in domain adaptation if we do not assume $P_s(Y|X = x) = P_t(Y|X = x)$ for all $x$, whereas in SSL the labeled data is all reliable.

There has been some work extending semi-supervised learning methods for domain adaptation. Dai et al. (2007a) proposed an EM-based algorithm for domain adaptation, which can be shown to be equivalent to a semi-supervised EM algorithm (Nigam et al., 2000) except that Dai et al. proposed to estimate the trade-off parameter between the labeled and the unlabeled data using the KL-divergence between the two domains. Jiang and Zhai (2007a) proposed to not only include weighted source domain instances but also weighted unlabeled target domain instances in training, which essentially combines instance weighting with bootstrapping. Xing et al. (2007) proposed a bridged refinement method for domain adaptation using label propagation on a nearest neighbor graph, which has resemblance to graph-based semi-supervised learning algorithms (Zhu, 2005; Chapelle et al., 2006).

## 5 Change of Representation

As has been pointed out, the cause of the domain adaptation problem is the difference between $P_t(X, Y)$ and $P_s(X, Y)$. Note that while the representation of $Y$ is fixed, the representation of $X$ can change if we use different features. Such a change of representation of $X$ can affect both the marginal distribution $P(X)$ and the conditional distribution $P(Y|X)$. One can assume that under some change of representation of $X$, $P_t(X, Y)$ and $P_s(X, Y)$ will become the same.

Formally, let $g : \mathcal{X} \rightarrow \mathcal{Z}$ denote a transformation function that transforms an observation $x$ represented in the original form into another form $z = g(x) \in \mathcal{Z}$. Define variable $Z$ and an induced distribution of $Z$

that satisfies $P(z) = \sum_{x \in \mathcal{X}, g(x)=z} P(x)$. The joint distribution of $Z$ and $Y$ is then

$$P(z, y) = \sum_{x \in \mathcal{X}, g(x)=z} P(x, y).$$

If we can find a transformation function $g$ so that under this transformation, we have $P_t(Z, Y) = P_s(Z, Y)$, then we no longer have the domain adaptation problem because the two domains have the same joint distribution of the observation and the class label. The optimal model $P(Y|Z, \theta^*)$ we learn to approximate $P_s(Y|Z)$ is still optimal for $P_t(Y|Z)$.

Note that with a change of representation, the entropy of $Y$ conditional on $Z$ is likely to increase from the entropy of $Y$ conditional on $X$, because $Z$ is usually a simpler representation of the observation than $X$, and thus encodes less information. In another word, the Bayes error rate usually increases under a change of representation. Therefore, the criteria for good transformation functions include not only the distance between the induced distributions $P_t(Z, Y)$ and $P_s(Z, Y)$ but also the amount of increment of the Bayes error rate.

Ben-David et al. (2007) first formally analyzed the effect of representation change for domain adaptation. They proved a generalization bound for domain adaptation that is dependent on the distance between the induced $P_s(Z, Y)$ and $P_t(Z, Y)$.

A special and simple kind of transformation is feature subset selection. Satpal and Sarawagi (2007) proposed a feature subset selection method for domain adaptation, where the criterion for selecting features is to minimize an approximated distance function between the distributions in the two domains. Note that to measure the distance between $P_s(Z, Y)$ and $P_t(Z, Y)$, we still need class labels in the target domain. To solve this problem, in (Satpal and Sarawagi, 2007), predicted labels for the target domain instances are used.

Blitzer et al. (2006) proposed a structural correspondence learning (SCL) algorithm that makes use of the unlabeled data from the target domain to find a low-rank representation that is suitable for domain adaptation. It is empirically shown in (Ben-David et al., 2007) that the low-rank representation found by SCL indeed decreases the distance between the distributions in the two domains. However, SCL does not directly try to find a representation $Z$ that minimizes the distance between $P_s(Z, Y)$ and $P_t(Z, Y)$. Instead, SCL tries to find a representation that works well for many related classification tasks for which labels are available in both the source and the target domains. The assumption is that if a representation $Z$ gives good performance for the many related classification tasks in both domains, then $Z$ is also a good representation for the main classification task we are interested in in both domains. The core algorithm in SCL is from (Ando and Zhang, 2005).

## 6 Bayesian Priors

Most of the work reviewed in the previous sections does not require labeled data from the target domain. In this section and the next section, we review two kinds of methods that work for supervised domain adaptation, i.e. when a small amount of labeled data from the target domain is available.

When we use the maximum a posterior (MAP) estimation approach for supervised learning, we can encode some prior knowledge about the classification model into a Bayesian prior distribution $P(\theta)$, where $\theta$ is the model parameter. More specifically, instead of maximizing

$$\prod_{i=1}^{N} P(y_i|x_i; \theta),$$

we maximize

$$P(\theta)\prod_{i=1}^{N} P(y_i|x_i;\theta).$$

In domain adaptation, the prior knowledge can be drawn from the source domain. More specifically, we first construct a Bayesian prior $P(\theta|D_s)$, which is dependent on the labeled instances from the source domain. We then maximize the following objective function:

$$P(\theta|D_s)P(D_{t,l}|\theta) \quad = \quad P(\theta|D_s)\prod_{i=1}^{N_{t,l}} P(y_i^t|x_i^t;\theta).$$

Li and Bilmes (2007) proposed a general Bayesian divergence prior framework for domain adaptation. They then showed how this general prior can be instantiated for generative classifiers and discriminative classifiers. Chelba and Acero (2004) applied this kind of a Bayesian prior for the task of adapting a maximum entropy capitalizer across domains.

# 7 Multi-Task Learning

Multi-task learning, sometimes known as transfer learning, is a machine learning topic highly related to domain adaptation. The original definition of multi-task learning considers a different setting than domain adaptation. In multi-task learning, there is a single distribution of the observation, i.e. a single $P(X)$. There are, however, a number of different output variables $Y_1, Y_2, \ldots, Y_M$, corresponding to $M$ different tasks. Therefore, there are $M$ different joint distributions $\{P(X, Y_k)\}_{k=1}^{M}$. Note that the class label sets are different for these $M$ different tasks. We assume that these different tasks are related. When learning $M$ conditional models $\{P(Y_k|X, \theta_k)\}_{k=1}^{M}$ for the $M$ tasks, we impose a common component shared by $\{\theta_k\}_{k=1}^{M}$. There have been a number of studies on multi-task learning (Caruana, 1997; Ben-David and Schuller, 2003; Micchelli and Pontil, 2005; Xue et al., 2007).

Strictly speaking, domain adaptation is a different problem than multi-task learning because we have only a single task but different domains. However, domain adaptation can be treated as a special case of multi-task learning, where we have two tasks, one on the source domain and the other on the target domain, and the class label sets of these two tasks are the same. If we have some labeled data from the target domain, we can then directly apply some existing multi-task learning algorithm.

Indeed, some domain adaptation methods proposed recently are essentially multi-task learning algorithms. Daumé III (2007) proposed a simple method for domain adaptation based on feature duplications. The idea is to make a domain-specific copy of the original features for each domain. An instance from domain $k$ is then represented by both the original features and the features specific to domain $k$. It can be shown that when linear classification algorithms are used, this feature duplication based method is equivalent to decomposing the model parameter $\theta_k$ for domain $k$ into $\theta_c + \theta'_k$, where $\theta_c$ is shared by all domains. This formulation then is very similar to the regularized multi-task learning method proposed by Evgeniou and Pontil (2004). Similarly, Jiang and Zhai (2007b) proposed a two-stage domain adaptation method, where in the first generalization stage, labeled instances from $K$ different source training domains are used together to train $K$ different models, but these models share a common component, and this common model component only applies to a subset of features that are considered generalizable across domains.

## 8 Ensemble Methods

In previous sections, only learning algorithms that return single classification models are considered. Ensemble methods are a type of learning algorithms that combine a set of models to construct a complex classifier for a classification problem. Ensemble methods include bagging, boosting, mixture of experts, etc. There has been some work using ensemble methods for domain adaptation.

One line of work uses mixture models. It can be assumed that there are a number of different component distributions $\{P^{(k)}(X,Y)\}_{k=1}^{K}$, each of which modeled by a simple model. The distribution of $X$ and $Y$ in either the source domain or the target domain is then a mixture of these component distributions. The source and the target domains are related because they share some of these component distributions. However, the mixture coefficients are different in the two domains, making the overall distributions different.

Daumé III and Marcu (2006) proposed a mixture model for domain adaptation, in which three mixture components are assumed, one shared by both the source and the target domains, one specific to the source domain, and one specific to the target domain. Labeled data from both the source and the target domains is needed to learn this three-component mixture model using the conditional expectation maximization (CEM) algorithm. Storkey and Sugiyama (2007) considered a more general mixture model in which the source and the target domains share more than one mixture components. However, they did not assume any target domain specific component, and as a result, no labeled data from the target domain is needed. The mixture model is learned using the expectation maximization (EM) algorithm.

Boosting is a general ensemble method that combines multiple weak learners to form a complex and effective classifier. Dai et al. (2007b) proposed to modify the widely-used *AdaBoost* algorithm to address the domain adaptation problem. With some labeled data from the target domain, the idea here is to put more weight on mistakenly classified target domain instances but less weight on mistakenly classified source domain instances in each iteration, because the goal is to improve the performance of the final classifier on the target domain only.

## References


Rie Ando and Tong Zhang. A framework for learning predictive structure from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, November 2005.

Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Proceedings of the 16th Annual Conference on Learning Theory*, Washington D.C., USA, August 2003.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 137–144. MIT Press, Cambridge, Massachusetts, USA, 2007.

Steffen Bickel and Tobias Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 161–168. MIT Press, Cambridge, Massachusetts, USA, 2007.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th Annual International Conference on Machine Learning*, pages 81–88, Corvallis, Oregon, USA, June 2007.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia, July 2006.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, Massachusetts, USA, 2008.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, July 1997.

Yee Seng Chan and Hwee Tou Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 89–96, Sydney, Australia, July 2006.

Yee Seng Chan and Hwee Tou Ng. Word sense disambiguation with distribution estimation. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1010–1015, Edingurgh, Scotland, July 2005.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

Ciprian Chelba and Alex Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 285–292, Barcelona, Spain, July 2004.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classifiers for text classification. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 540–545, Vancouver, British Columbia, Canada, July 2007a.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th Annual International Conference on Machine Learning*, pages 193–200, Corvallis, Oregon, USA, June 2007b.

Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007.

Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, May 2006.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117, Seattle, Washington, USA, August 2004.

James J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–161, January 1979.

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, Cambridge, MA, 2007.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–450, November 2002.

Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 254–271, Prague, Czech Republic, June 2007a.

Jing Jiang and ChengXiang Zhai. A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the ACM 16th Conference on Information and Knowledge Management*, pages 401–410, 2007b.

Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the 14th Annual International Conference on Machine Learning*, pages 179–186, Nashville, Tennessee, USA, July 1997.

Xiao Li and Jeff Bilmes. A Bayesian divergence prior for classifier adaptation. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, March 2007.

Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46(1–3):191–202, January 2002.

Charles A. Micchelli and Massimiliano Pontil. Kernels for multi-task learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 921–928. MIT Press, Cambridge, Massachusetts, USA, 2005.

Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2–3):103–134, May 2000.

Sandeepkumar Satpal and Sunita Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 224–235, Warsaw, Poland, September 2007.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000.

Amos J. Storkey and Masashi Sugiyama. Mixture regression for covariate shift. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1337–1344. MIT Press, Cambridge, Massachusetts, USA, 2007.

Masashi Sugiyama and Klaus-Robert Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4):249–279, 2005.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, second edition, 1999.

Dikan Xing, Wenyuan Dai, Gui-Rong Xue, and Yong Yu. Bridged refinement for transfer learning. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 324–335, Warsaw, Poland, September 2007.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, May 2007.

Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21th Annual International Conference on Machine Learning*, pages 114–121, Banff, Canada, July 2004.

Jingbo Zhu and Eduard Hovy. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 783–790, Prague, Czech Republic, June 2007.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison, 2005.