A Probabilistic Covariate Shift Assumption for Domain Adaptation

Tameem Adel
University of Waterloo
thesham@uwaterloo.ca

Alexander Wong
University of Waterloo
a28wong@uwaterloo.ca

Abstract

The aim of domain adaptation algorithms is to establish a learner, trained on labeled data from a source domain, that can classify samples from a target domain, in which few or no labeled data are available for training. Covariate shift, a primary assumption in several works on domain adaptation, assumes that the labeling functions of source and target domains are identical. We present a domain adaptation algorithm that assumes a relaxed version of covariate shift where the assumption that the labeling functions of the source and target domains are identical holds with a certain probability. Assuming a source deterministic large margin binary classifier, the farther a target instance is from the source decision boundary, the higher the probability that covariate shift holds. In this context, given a target unlabeled sample and no target labeled data, we develop a domain adaptation algorithm that bases its labeling decisions both on the source learner and on the similarities between the target unlabeled instances. The source labeling function decisions associated with probabilistic covariate shift, along with the target similarities are concurrently expressed on a similarity graph. We evaluate our proposed algorithm on a benchmark sentiment analysis (and domain adaptation) dataset, where state-of-the-art adaptation results are achieved. We also derive a lower bound on the performance of the algorithm.

Introduction

In machine learning, domain adaptation refers to the situation when one learns from samples drawn from a certain domain, and tests the resulting hypothesis on samples drawn from another domain. For many application domains, obtaining enough labeled data for training is not easy due to cost, availability, etc. One possible approach of addressing this challenge is to leverage labeled data that may be available from similar domains, and this is basically where the concept of domain adaptation is paramount. In the domain adaptation literature, the training domain is commonly referred to as the source domain, whereas the test domain is commonly referred to as the target domain. The hypothesis performance in the target domain is the main metric in domain adaptation (Ben-David et al. 2007). In order for domain adaptation to achieve reasonable classification performance, there must be some sort of similarity between the two domains, so that the learning hypothesis can have a chance of performing better than a random classifier on target data. Such performance depends both on the hypothesis performance in the source domain, and on the relationship between the two domains.

In some works, existing domain adaptation algorithms have been grouped into two categories: i) conservative and ii) adaptive. A “conservative” domain adaptation algorithm is one that learns only from source labeled data, without making use of data from the target domain, whereas “adaptive” algorithms make use of target generated data (Ben-David and Urner 2012; 2014). With respect to this taxonomy, our proposed algorithm is adaptive since it adapts its target learner based on target unlabeled data. One example of a domain adaptation problem is learning from images captured under certain lighting conditions or taken by a certain type of camera, and performing an object recognition task on images captured under different lighting conditions or taken by another camera type, respectively (Saenko et al. 2010). Another common example is to train a spam filter on emails belonging to one address, and test it on a different address.

In domain adaptation, understanding the relationship between the source and target domains is fundamental for the learner. The two extremes in the source-target domain relationship spectrum are: i) If there is no relationship between the two domains at all, there is no basis for domain adaptation, and ii) if the source and target domains are identical then there is no need for domain adaptation. Covariate shift is a common assumption that formulates this relationship in the majority of previous works on domain adaptation, e.g. (Sugiyama and Mueller 2005; Huang et al. 2006; Ben-David and Urner 2014). Covariate shift refers to the assumption that the labeling functions of the source and target domains are identical. Practically speaking, covariate shift is a reasonable assumption for some, but not all, domain adaptation problems. As an example of an adaptation problem where covariate shift does not hold, let us assume that there is a sentiment analysis dataset containing the music and films domains. The sentence ‘I love the story but not the music’, is an example of an input pattern that has different rates depending on the domain (Thet et al. 2009). Another example where covariate shift does not hold is in remote sensing applications where an object is moving and the image used for training (source) shows an object that slightly moved before acquiring another image (target).

A majority of domain adaptation algorithms rely on covariate shift, while the rest mostly seek a common feature representation where covariate shift holds, or have access to target labeled data. In this paper, we introduce a domain adaptation algorithm that takes a different strategy based on a relaxed or probabilistic version of covariate shift. Assuming a binary classifier, for each point in the learning space, probabilistic covariate shift generically states that source and target labels are identical with a probability proportionate to: i) the distance from the source decision boundary in the case of a source large margin learner, or to ii) the source labeling degree of certainty in the case of a source probabilistic classifier. The work presented here focuses on the case of a source large margin learner.

The proposed domain adaptation algorithm learns from similarities between target unlabeled instances, as well as the corresponding source labels associated with probabilistic covariate shift. In order to label the target sample, we establish a similarity graph whose edge weights are founded on the combination of similarities between target instances and probabilities that their respective source labels hold in the target (according to probabilistic covariate shift). We assume a source deterministic binary classifier, and prior knowledge that the source hypothesis class is that of large margin classifiers. The learner makes use of a source labeled sample as well as a target unlabeled sample in the labeling task.

Our contributions are as follows: First, we introduce a domain adaptation algorithm that neither requires labeled data from the target domain nor relies on the standard covariate shift assumption. Second, the proposed algorithm achieves state-of-the-art adaptation results on the Amazon reviews sentiment analysis dataset. Third, under the assumption of a weight ratio between the source and target marginal distributions, ϕ-Lipschitz property with respect to the target distribution, and probabilistic covariate shift, we derive a theoretical lower bound on the performance of the algorithm.

Related Work

As per the training samples available to the learner, the framework followed in this paper was formalized by Ben-David et al. (Ben-David et al. 2007). It assumes that source data and a target unlabeled sample (no target labeled data) are available to the learner. It further notes that feature representation of an adaptation problem is sound if it achieves low source domain error and minimizes the distance between the source and target marginal distributions. In this framework, a few additive measures of distance between source and target marginals were introduced via the available training samples, e.g. d_A (Kifer, Ben-David, and Gehrke 2004) and discrepancy distance (Mansour, Mohri, and Rostamizadeh 2009). Various other studies were proposed in the same framework (Blitzer, Dredze, and Pereira 2007; Glorot, Bordes, and Bengio 2011; Chen, Xu, and Weinberger 2012). Other domain adaptation learners have access to target labeled samples as well, e.g. (Chen, Weinberger, and Blitzer 2011; Blitzer et al. 2008). The former iteratively learned a target predictor along with an associated subset of source and target features via an optimization problem. Under covariate shift, the latter minimized a convex combination of source and target empirical risk to derive a uniform convergence bound (Blitzer et al. 2008; Ben-David et al. 2010). On the other hand, Ben-David and Urner (Ben-David and Urner 2014) presented a conservative adaptation learner, which learns from source samples only.

One major difference between the framework of (Ben-David et al. 2007) and the work presented in this paper is the standard vs. probabilistic covariate shift assumption. Most domain adaptation algorithms were founded on the covariate shift assumption, e.g. (Bickel, Bruckner, and Scheffer 2007; Huang et al. 2006; Kifer, Ben-David, and Gehrke 2004; Mansour, Mohri, and Rostamizadeh 2009). Bickel, Bruckner, and Scheffer (Bickel, Bruckner, and Scheffer 2007) mapped covariate shift based learning onto an integrated optimization problem and established a kernel logistic regression classifier for solving it. To handle the source and target marginal distributional difference, Huang et al. (Huang et al. 2006) proposed a nonparametric method to produce resampling weights without distribution estimation. In addition, some domain adaptation paradigms, under the covariate shift setup, corrected sample selection bias by importance weighting (Cortes et al. 2008; Cortes, Mansour, and Mohri 2010; Sugiyama, Krauledat, and Mueller 2007).

Examples of domain adaptation algorithms not assuming covariate shift include (Bergamo and Torresani 2010) and (Schweikert et al. 2009) where target labeled data constitute part of the training data. Other algorithms targeted a problem where labeling functions of multiple sources are not identical due to noise (Crammer, Kearns, and Wortman 2006; 2008). However, their problem is different from adaptation since all training marginal distributions are identical.

A few domain adaptation algorithms can be adjusted and used in transfer learning. Transfer learning refers to applying the learning knowledge obtained from a source task(s) to develop a hypothesis for a target task (Ben-David and Schuller 2003; Dai et al. 2007; Maurer 2005). In transfer learning, the learner typically has access to target labeled data, unlike the learner in our problem definition.

Methodology

Notation Let X be an instance set and Y be a label set. We address a binary labeling problem; $Y = \{1, -1\}$. For a distribution, P, over $X \times \{0, 1\}$, we denote a learner by $l : X \to Y$, and its error probability by $Err_P(l) = Pr_{(x,y)\sim P}(y \neq l(x))$.

Let P_S and P_T be the source and target distributions, respectively, over $X \times \{0, 1\}$. Also, let D_S and D_T be the marginal distributions of P_S and P_T, respectively, over X.

The domain adaptation learner receives as input a sample, S, consisting of n source labeled instances drawn according to P_S. The learner also bases its labeling decisions on a sample, T, of m target unlabeled instances drawn according to D_T. Assume that l_S belongs to a hypothesis class of large margin learners. The aim is to establish a target learner, l_T, to label T with a minimized error probability, $Err_{P_T}(l_T)$.

Source Learning

As we are interested in source large margin classifiers here, we learn from the source domain using support vector machines (SVMs) with a soft margin. The dual form of soft margin SVMs (1), which is formulated into a quadratic programming problem, is solved using sequential minimal optimization (SMO) (Platt 1999).

$$maximize \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i \cdot x_j \tag{1}$$

$$w.r.t. \;\; \alpha_i. \quad subject\ to: \; 0 \leq \alpha_i \leq C, \;\; \sum_{i=1}^{n} \alpha_i y_i = 0$$

Where α_i, for $i = 1, 2, ..., n$, denote Lagrange multipliers, and $\alpha_i = 0$ for all training instances except the support vectors. It is a linear SVM learner with $C = 1$. We denote the number of support vectors by n_{sv} and the bias term by $b = \frac{1}{n_{sv}} \sum_{i=1}^{n_{sv}} (\sum_{j=1}^{n} \alpha_j y_j (x_j \cdot x_i) - y_i)$. Hence, the output of the classifier for an instance x is computed as:

$$sign[(\sum_{i=1}^{n} \alpha_i y_i (x_i \cdot x)) + b] \tag{2}$$

Assumptions Controlling Domain Adaptation

As far as the source domain is concerned, it is an ordinary supervised learning problem. However, the primary task is to make use of source labeled data along with target unlabeled data so that the target data can be accurately labeled. Our main goal here is to answer the question as to whether it is possible to successfully learn without target labeled data and with an assumption that can be more realistic than the standard covariate shift in many applications. We address domain adaptation problems satisfying the following:

Weight Ratio For the subsets of the domain X where $D_S(X) \neq 0$, denote the weight ratio, C_R, of the target and source marginal distributions, D_T and D_S, as:

$$C_R(D_T, D_S) = \inf_{D_S(x) \neq 0} \frac{D_T(x)}{D_S(x)} \tag{3}$$

Lipschitz Condition A function $f : R^d \rightarrow R$ is ϕ-Lipschitz if (Shalev-Shwartz and Ben-David 2014):

$$\forall x_1, x_2 \in [0,1]^d : \quad |f(x_1) - f(x_2)| \leq \phi \cdot \|x_1 - x_2\| \tag{4}$$

In our case, the deterministic labeling function, f, is denoted by, $f : R^d \rightarrow \{0, 1\}$. The ϕ-Lipschitz condition is assumed only when we derive the error bound. Moreover, it is not assumed to hold in the dataset used in the experiments.

Probabilistic Covariate Shift The assumption that covariate shift always holds, $l_S(x) = l_T(x) \; \forall x \in X$, may be reasonable in some domain adaptation tasks, yet it is not realistic in many others. Assuming a source large margin classifier, we introduce a generalized version of covariate shift where source and target labeling functions of x are identical with a probability proportionate to its distance from the source decision boundary. In other words, the farther an instance is from the boundary, the higher our confidence of its source and target labels being identical. A measure proportionate to distance from the boundary, $|f_{svm}|$, is computed as the absolute value of the SVM classifier's output shown in (2). We formalize the probabilistic covariate shift assumption by: $\forall \psi > 0$ and x where $f_{svm}(x) \neq 0$:

$$Pr_{x \sim D_T}(\exists x : l_S(x) \neq l_T(x) \; \wedge \; \frac{1}{|f_{svm}(x)|} \leq \psi) \leq \upsilon(\psi) \tag{5}$$

The condition $f_{svm}(x) \neq 0$ is added due to the fraction $\frac{1}{|f_{svm}(x)|}$, but instances, x, with $f_{svm}(x) = 0$ are still the least likely to have identical source and target labels. $\upsilon(\psi)$ is a monotonically increasing function of ψ. Probabilistic covariate shift generalizes the standard covariate shift since setting $\upsilon(\psi) = 0$, for all ψ results in the standard covariate shift. From the decision boundary perspective, probabilistic covariate shift states that the target boundary is "probably" close to the source boundary; as such, probability that the former is at a certain location, is inversely proportionate to the distance between the source boundary and such location.

Target Learning

We aim at labeling a target sample, T, based on probabilistic covariate shift (an instance's source label is not guaranteed to be identical to its target label) and with no target labeled data available for learning. The information at our disposal consists of two main aspects: i) the global assumption that the target boundary is more likely to be close, rather than far, to the source boundary, and ii) the local assumption that nearby target instances are likely to have the same label. Hence, we base our target labeling decisions on these two aspects. We establish a similarity graph where both global and local labeling information are expressed so that the target sample can be labeled accordingly. Similarity graphs originally derive pairwise similarity scores based on local neighborhoods between unlabeled nodes. Other versions of similarity graphs have been developed in order to handle labeled and unlabeled nodes in a transductive sense, e.g. (Blum and Chawla 2001). Here we address a learning problem where we have instances that are neither unlabeled nor labeled with certainty. Instead, each target instance label has an associated probability or degree of certainty, and we want to assign weights on edges between the graph nodes (target instances) reflecting both the global and local labeling aspects. We propose a KNN probabilistic mincut similarity graph (6) where the local aspect is commonly expressed (7) and the global aspect is reflected by the expression introduced in (8). Similar to the original unsupervised similarity graphs, a spectral ratiocut solution to the KNN probabilistic mincut optimization problem is subsequently presented.

KNN Probabilistic Mincut We construct a weighted similarity graph where each target instance is a vertex (node). Edges between vertices are weighted based on the following rules. There is no edge between two vertices if neither of them is a nearest neighbor of the other. Otherwise, a symmetric weighted KNN function is utilized for edge weighting. An edge between x_i and x_j, $w(x_i, x_j)$, is assigned a weight based on two factors (6), symmetric KNN similarity between x_i and x_j, and the corresponding labeling probabilities, $Pr_{x_i \sim D_T}(l_T(x_i) = 1)$ and $Pr_{x_j \sim D_T}(l_T(x_j) = 1)$, w.r.t. probabilistic covariate shift. Symmetric KNN similarity depends on the two ordered similarity pairs $sim(x_i, x_j)$ and $sim(x_j, x_i)$. $sim(x_i, x_j)$ is calculated by (7). Gaussian similarity is chosen as the similarity function w_{ij}.

$$w(x_i, x_j) = label\_sim(x_i, x_j) \times [sim(x_i, x_j) + sim(x_j, x_i)] \tag{6}$$

$$sim(x_i, x_j) = \begin{cases} \frac{w_{ij}}{\sum_{x_j \in knn(x_i)} w_{ij}} & x_j \in knn(x_i) \\ 0 & x_j \notin knn(x_i) \end{cases} \tag{7}$$

On the other hand, the global labeling information is expressed via $label\_sim(x_i, x_j)$. For an instance, x, the target labeling probability is based on probabilistic covariate shift, since: $Pr(l_T(x) = 1) = Pr(l_S(x) \neq l_T(x))$ if $l_S(x) = 0$, and $Pr(l_T(x) = 1) = 1 - Pr(l_S(x) \neq l_T(x))$ if $l_S(x) = 1$.

Pairwise similarities of labeling probabilities, $label\_sim(x_i, x_j)$, are expressed as shown in (8).

$$label\_sim(x_i, x_j) = 1 \ + 4\, C_S \times \tag{8}$$
$$(Pr_{x_i \sim D_T}(l_T(x_i) = 1) - \frac{1}{2})\ (Pr_{x_j \sim D_T}(l_T(x_j) = 1) - \frac{1}{2})$$

The constant C_S controls the weight given to source labeling probabilities; if $C_S = 0$, then $label\_sim(x_i, x_j) = 1$ so that only the local neighborhood counts and the problem turns into a partitioning problem on a KNN similarity graph. We use $C_S = 1$ in our experiments. The limits are as follows: If $Pr(l_T(x_i) = 1) = Pr(l_T(x_j) = 1) = 1$ or $Pr(l_T(x_i) = 1) = Pr(l_T(x_j) = 1) = 0$, then $label\_sim(x_i, x_j) = 2$. If $Pr(l_T(x_i) = 1) = 1$ and $Pr(l_T(x_j) = 1) = 0$, or vice versa, then $label\_sim(x_i, x_j) = 0$. In case $Pr(l_T(x_i) = 1) = Pr(l_T(x_j) = 1) = 0.5$, or at least one of them is equal to 0.5, meaning there is no labeling information to derive, then $label\_sim(x_i, x_j) = 1$.

In order to assign labels to target instances via the similarity graph, the graph is cut into two partitions. A minimum cut is sought such that the summation of edge weights connecting the two partitions is minimized.

Probabilistic mincut graph aims at minimizing the total weight of edges connecting the two partitions, i.e. edges whose removal would disconnect the the two partitions, T^+ and T^-. Equation (9) depicts the probabilistic mincut graph optimization problem.

$$minimize_{T^+, T^-} \sum_{\forall x_i \in T^+, x_j \in T^-} w(x_i, x_j) \tag{9}$$

KNN Probabilistic (Spectral) Ratiocut As can be seen in (6-9), the KNN probabilistic mincut graph minimizes the cut by minimizing the sum of weights across the two partitions (groups). One major problem with the probabilistic mincut graph algorithm is that it may lead to degenerate cuts, i.e. target instances being split into two very unbalanced groups; T^+ and T^- can be unbalanced in terms of size. The intuition behind this problem is similar to the one in Blum and Chawla (Blum and Chawla 2001). This deficiency can be overcome by minimizing the average, rather than sum, of weights across the two groups. Dividing the KNN Probabilistic Mincut optimization objective function in (9) by $|T^+|\ |T^-|$ leads to the ratiocut objective function (different only by a constant) (Hagen and Kahng 1992). In its original unsupervised form, the ratiocut problem is NP-hard but it can be efficiently solved using spectral grouping techniques. We follow an equivalent path here. Let W be the adjacency matrix consisting of $w(x_i, x_j)$, and Dg be the degree matrix where $Dg_{ii} = \sum_{j=1}^{n} w(x_i, x_j)$. A spectral formulation of $\frac{\sum w(x_i, x_j)}{|T^+|\ |T^-|}$ is shown in (10) (Luxburg 2007). The matrix L is the unnormalized Laplacian.

$$minimize_F\ F^T L F, \text{ for } L = Dg - W,\ f \in \{T^+, T^-\}$$
$$subject\ to:\ F \perp 1,\ \ \|F\| = \sqrt{n} \tag{10}$$

Performance Lower Bound

We derive a lower bound on the target domain performance of our algorithm, which we refer to as ProbCS. In this section, we construct a 1-NN probabilistic mincut graph. Vertices of the graph represent a target unlabeled sample, T. Its adjacency matrix is calculated by (6). We assume a weight ratio, C_R, between the marginals, D_T and D_S. Probabilistic covariate shift controls the relationship between the deterministic binary labeling functions l_S and l_T by (5). Also, l_T satisfies the ϕ-Lipschitz property.

Theorem 1 The average distance between a target instance and the source SVM boundary is proportionate to the ProbCS lower bound, whereas the average distance between target instances is inversely proportionate to the same bound.

Proof The probability that ProbCS makes a correct labeling decision on a target instance, $Corr_{P_T}$(ProbCS), is equal to the probability that $x, x \in T$, is ultimately assigned a label, $l_{\text{ProbCS}}(x)$, that is equal to its correct label $l_T(x)$:

$$E_{T \sim D_T^m}[Corr_{P_T}(\text{ProbCS})] = Pr_{y = l_T(x), \acute{y} = l_{\text{ProbCS}}(x)}(y = \acute{y})$$
$$E_{T \sim D_T^m}[Err_{P_T}(\text{ProbCS})] = Pr(y \neq \acute{y}) \tag{11}$$

Refer to probabilistic covariate shift as PCS. For $x, x \in T$, l_T(SVM) denotes the most likely label of x based on SVM and PCS only, before taking target local neighborhoods into consideration. We use l_T(SVM) only in theory so that we can separately analyze the errors resulting from the 1-NN probabilistic mincut graph and those resulting from PCS and the source SVM classifier.

Theorem 2 Assume that l_T(SVM) is the ground truth for a target 1-NN classifier (we will get back to this assumption later). The total cut of the 1-NN probabilistic mincut graph is equal to the number of leave-one-out (LOO) cross-validation classification errors of a binary 1-NN classifier acting on the same data and same domain.

Proof LOO cross-validation error of a binary 1-NN classifier learning from a dataset of size m is equal to the number of instances, z, where $l(z) \neq l(NN(z))$. On the other hand, recall that the total cut of the 1-NN probabilistic mincut graph is defined as the sum of weights of edges connecting the two partitions, T^+, T^-. Let $sim(x_i, x_j) = 1$ if x_j is the nearest neighbor of x_i, and $sim(x_i, x_j) = 0$ otherwise. Thus, the total cost of a minimum cut is equal to $\sum w(x_i, x_j)$ across the cut, which in turn is equal to $\sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} sim(x_i, x_j)$, where $x_i \in T^+$ and $x_j \in T^-$, or vice versa. The latter summation is exactly equal to the LOO cross-validation error of 1-NN. Equivalence can be proven for larger values of k, but we use $k = 1$ for simplicity.

Based on Theorem 2, and on the fact that LOO cross-validation error is an almost unbiased estimate of generalization error of KNN (Luntz and Brailovsky 1969; Elisseeff and Pontil 2002), $E_{T \sim D_T^m}[Err_{P_T}(\text{1NN-Prob-Mincut})]$ can be estimated by $E_{T \sim D_T^m}[Err_{P_T}(\text{1NN})]$. Back to (11), denote by A the event that $l_T(\text{SVM}[x]) = l_T(x)$, and by B the event that 1-NN leads to a correct classification. Therefore:

$$Pr(A) \equiv Pr(l_T(\text{SVM})[x] = l_T(x))$$
$$Pr(B|A) \equiv 1 - E_{T \sim D_T^m}[Err_{P_T}(\text{1NN})]$$
$$Pr(y = \acute{y}) = Pr(A, B) = Pr(A) \times Pr(B|A) \tag{12}$$

The rest of this section will be dedicated to the calculation of $Pr(A)$ and $Pr(B|A)$. We begin by analyzing

$E_{T\sim D_T^m}[Err_{P_T}(\text{1NN})]$. This is the expected error of a 1-NN classifier on one domain (the target), assuming that it learns from a correctly labeled sample of size m. Denote $Pr(l_T(x)\text{=}1)$ by $pl_T(x)$.

$$\begin{aligned} &Pr(l_T(x) \neq l_T(\text{1NN})[x]) = \\ &pl_T(x)(1 - pl_T(\text{1NN})[x]) + pl_T(\text{1NN})[x](1 - pl_T(x)) \\ &\quad \leq 2pl_T(x)(1 - pl_T(x)) + |pl_T(\text{1NN})[x] - pl_T(x)| \end{aligned} \tag{13}$$

Where the inequality is the outcome of standard algebraic manipulations, same as those in the proof of Theorem 7 in Ben-David and Urner (Ben-David and Urner 2014).

$$\begin{aligned} &E_{T\sim D_T^m}[Err_{P_T}(\text{1NN})] = E[Pr(l_T(x) \neq l_T(\text{1NN})[x])] \\ &\quad \leq E[2pl_T(x)(1 - pl_T(x)) + |pl_T(\text{1NN})[x] - pl_T(x)|] \end{aligned}$$

Due to l_T being deterministic, the first term is always 0. Assuming $X = [0,1]^d$ and using ϕ-Lipschitz (4) leads to:

$$E_{T\sim D_T^m}[Err_{P_T}(\text{1NN})] \leq \phi E_{T\sim D_T^m}[\|NN(x) - x\|] \tag{14}$$

Equation (14) determines the expected error of a target 1-NN learner, which, as stated by Theorem 2, is equivalent to the expected error of a target 1-NN probabilistic mincut graph. The intuition that the average distance between an instance x and its NN, $E[\|NN(x) - x\|]$ is proportionate to the 1-NN expected error, is supported by the bound.

Back to (12), we move on to $Pr(A)$. Denote by U the event: PCS doesn't change $l_S(\text{SVM})[x]$, and by V the complementary event: PCS changes $l_S(\text{SVM})[x]$.

$$\begin{aligned} &E_{T\sim D_T^m}(l_T(\text{SVM})[x] = l_T(x)) = \\ &E[((l_T(x)\text{=}l_S(\text{SVM})[x]), U) \vee ((l_T(x) \neq l_S(\text{SVM})[x]), V)] \\ &\geq E_{T\sim D_T^m}((l_T(x) = l_S(\text{SVM})[x]), U) \end{aligned} \tag{15}$$

Using the PCS formalization (5) and Markov's inequality:

$$E(V) \leq \upsilon(\psi) + \frac{\psi}{E(|f_{\text{svm}}(x)|)},\ E(U) \geq 1 - E(V) \tag{16}$$

We assume that $E_{x\sim D_S}(l_T(x) = l_S(\text{SVM})[x]) = 0.5$, and assume that the events $l_T(x) = l_S(\text{SVM})[x]$ and U are independent. The latter assumption is rigid (leading to a less tight bound), because the intuition behind assuming PCS (related to U) should be based on some prior knowledge (leading to expected correlation between $l_T(x)$ and $l_S(\text{SVM})[x]$). Based on these two assumptions, and from (3) and (16):

$$\begin{aligned} &E(l_T(\text{SVM})[x] = l_T(x)) = \\ &\geq E_{x\sim D_T}(l_T(x) = l_S(\text{SVM})[x]) \times E(U) \\ &\geq C_R \times E_{x\sim D_S}(l_T(x) = l_S(\text{SVM})[x]) \times E(U) \\ &\geq 0.5\ C_R \times (1 - \upsilon(\psi) - \frac{\psi}{E(|f_{\text{svm}}(x)|)}) \end{aligned} \tag{17}$$

From (12), (14) and (17) into (11):

$$\begin{aligned} &E_{T\sim D_T^m}[Corr_{P_T}(\text{ProbCS})] \geq (1 - \upsilon(\psi) - \frac{\psi}{E(|f_{\text{svm}}(x)|)}) \\ &\times 0.5\ C_R \times (1 - \phi\ E_{T\sim D_T^m}[\|NN(x) - x\|]) \end{aligned} \tag{18}$$

Hence, the farther the expected distance from a target instance to the source SVM boundary, $E(|f_{\text{svm}}(x)|)$, the higher (better) the ProbCS lower bound. Also, the smaller the average distance between a target instance x and its nearest neighbor, $E(\|NN(x) - x\|)$, the higher the ProbCS lower bound. This concludes the proof of Theorem 1.

Experiments

To evaluate the performance of the proposed algorithm, we run our experiments on the Amazon reviews dataset (Blitzer, Dredze, and Pereira 2007). This is a sentiment analysis dataset that has often been used for evaluation in domain adaptation. The Amazon reviews dataset originally contains more than 340,000 reviews from 22 different domains, where each domain represents a product type. The dataset as a whole is very heterogeneous and extremely unbalanced. This is the reason why its most common use is in the form constructed by Blitzer, Dredze, and Pereira (Blitzer, Dredze, and Pereira 2007), which consists of 4 different domains: i) books, ii) DVDs, iii) electronics, and iv) kitchen appliances. Each review originally has a rating (0-5 stars), but again we apply the convention, e.g. (Glorot, Bordes, and Bengio 2011; Chen, Xu, and Weinberger 2012), of turning these ratings into binary labels by assigning a positive (negative) label to reviews with rating > 3 (≤ 3). The updated form of Amazon reviews is more balanced as per the number of positive and negative reviews. Two more common things we follow in order to provide consistent comparisons with previous algorithms are: i) a pre-processing step where features are reweighted with standard tf-idf (Salton and Buckley 1988), ii) we select the 5,000 most frequent features (vocabulary terms of unigrams and bigrams). Information about the Amazon reviews dataset is displayed in Table 1.

Table 1: Statistics of the Amazon reviews sentiment analysis dataset.

Domain	**# Labeled rev.**	**# Unlabeled rev.**	**% +ve**
Books	2000	4465	50%
DVDs	2000	5945	50%
Electronics	2000	5681	50%
Kitchen appl.	2000	3586	50%

Similar to Glorot, Bordes, and Bengio (Glorot, Bordes, and Bengio 2011; Chen, Xu, and Weinberger 2012), our metric is a measure of the discrepancy error, i.e. the error due to the difference between the source and target domains. Let the transfer error, $e(S,T)$, be the test error obtained by a learner trained on S and tested on T, and let the in-domain error, $e(T,T)$, be the test error obtained by a learner trained on a target sample T_1 and tested on another target sample T_2. Also, denote by $e_b(T,T)$, the in-domain error of the baseline linear SVM. The metric used to compare domain adaptation algorithms is referred to as the transfer loss, $e(S,T) - e_b(T,T)$. The lower the algebraic transfer loss value, the better the adapted classifier performance compared to the one-domain baseline classifier.

We have 12 adaptation tasks, moving from one domain (source) to another (target), e.g. $B \rightarrow D$. Our domains are B (books), D (DVDs), E (Electronics) and K (Kitchen appliances). The strategy we pursue in order to implement ProbCS on the Amazon reviews dataset is as follows:

1. We train a linear SVM on S, the labeled 2000 instances (reviews) of a source domain.
2. Regarding probabilistic covariate shift formulated in (5), we define $\upsilon(\psi)$ as shown in (19):
$$\upsilon(\psi) = \begin{cases} 0.5 & f_{svm} \leq |1| \\ 0 & f_{svm} > |1| \end{cases} \tag{19}$$
The above results from our assumption that some Amazon reviews can be invariably indicated as positive (or negative) in several domains. Thus, we make the assumption that target reviews not within the source SVM margin can be safely classified by the source SVM, and set $\upsilon(\psi) = 0$

in this part of the target domain. For target reviews within the source SVM margin, we do not make further assumptions as per their target labels and we set $\upsilon(\psi) = 0.5$. We experimented other values, $f_{svm} \leq |2|, |0.5|$, etc, and results support the intuition as $f_{svm} \leq |1|$ performs best. Results are notably worse when standard covariate shift is assumed to hold within the source SVM margin.

3. From (19): For each instance x_i in T, set the corresponding $\upsilon(\psi)$ according to $f_{svm}(x_i)$. Note that for x_i not within the SVM margin, $\upsilon(\psi) = 0$, thus $Pr(l_S(x_i) \neq l_T(x_i)) = 0$ and they retain their source labels. This is later utilized in deciding which group is positive, T^+, and which is negative, T^-. The group containing instances $x_i,\ x_i \in T,\ f_{svm}(x_i) > +1$ is T^+, and the one containing instances $x_i,\ x_i \in T,\ f_{svm}(x_i) < -1$ is T^-.
4. We construct the KNN similarity graph whose vertices represent instances of T, and the adjacency matrix is calculated by (6). Afterwards, a spectral solution to the probabilistic ratiocut optimization problem is computed by (10). As a result, T^+ and T^- are obtained. Instances of T^+ (T^-) are assigned the label $+1$ (-1).

In Figure 1, linear SVM is a domain adaptation learner, trained on source data assuming standard covariate shift; $\upsilon(\psi) = 0$, for all ψ. In the 12 domain adaptation tasks, we calculated number of source test instances within the source linear SVM margin, just to compare it with number of target test instances within the same margin. The latter was always bigger, which supports the probabilistic covariate shift intuition that the labeling functions are different. Assume, for example, no target test instances lay in the margin, this would have suggested that the source classifier is good enough for target data and hardly needs adaptation.

Results of linear SVM vs. ProbCS, which are in favour of ProbCS, as shown in Figure 1, support the intuition behind assuming probabilistic covariate shift, rather than standard covariate shift, on the Amazon reviews dataset. On the other hand, an experiment was performed based on spectral ratiocut with an ordinary KNN weighting function, not taking source labeling probabilities into consideration. The intuition behind this experiment is to check whether the local neighborhood assumption is enough to group target instances. Results of this experiment are far worse than all the adaptation algorithms. As a result (and also due to space), we did not add this experiment to Figure 1. The last two experiments suggest that, for Amazon reviews, probabilistic covariate shift, not its standard correspondent, and the utilized values of $\upsilon(\psi)$ are plausible assumptions. Another experiment was performed based on the same values of $\upsilon(\psi)$ by running transductive SVM (Joachims 1999b) using (Joachims 1999a). In addition to being slow, performance was considerably worse than ProbCS in the 12 tasks, supporting the intuition that the local neighborhood assumption is useful for adaptation learning from Amazon reviews.

Besides linear SVM, the other 4 algorithms we compare our results to represent state-of-the-art adaptation results on the Amazon reviews dataset. First, structural correspondence learning (SCL) (Blitzer, McDonald, and Pereira 2006) utilizes unlabeled data from both domains to find correspondences among their features. Second, CODA (Chen, Weinberger, and Blitzer 2011) is briefly described in the Related Work section. Finally, SDA (Glorot, Bordes, and Bengio 2011) and mSDA (Chen, Xu, and Weinberger 2012) are two adaptation algorithms based on learning robust feature representations by stacked denoising autoencoders (SDAs). mSDA (Chen, Xu, and Weinberger 2012) marginalizes noise and is more efficient than SDA in terms of computational cost and scalability. For CODA, SDA and mSDA, we used the implementation provided by the authors. For SCL, we report the results reported in (Chen, Xu, and Weinberger 2012) because the settings are identical. Parameter values are determined by 10-fold cross-validation ($K = 7$ for ProbCS).

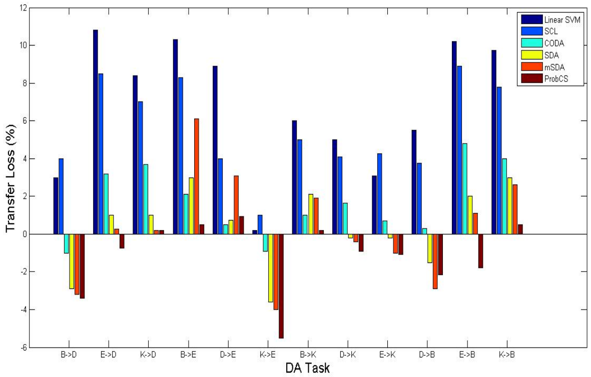

Figure 1: Comparison of 6 domain adaptation algorithms based on the transfer loss. In 9 out of the 12 adaptation tasks, ProbCS achieves the best performance (lowest transfer loss).

Transfer loss values, for the 12 domain adaptation tasks and 6 domain adaptation algorithms, are reported in Figure 1. ProbCS achieves the lowest transfer loss in 9 tasks and joint lowest with mSDA in $K \to D$. A negative transfer loss value indicates an improvement achieved by the corresponding domain adaptation algorithm (notwithstanding the domain discrepancy) over a baseline linear SVM trained and tested on the same domain ($e(S, T) < e_b(T, T)$).

Conclusion

We present a domain adaptation algorithm that learns without any target labeled data, assuming probabilistic covariate shift, where the assumption that the labeling functions of the source and target domains are identical holds with a certain probability. Probabilistic covariate shift assumes that the target decision boundary may not be the same as the source boundary but is probably not too far. In other words, the probability that the target boundary is at a certain distance from the source one, is inversely proportionate to the distance. Target learning is performed on the basis of a similarity graph representing probabilistic covariate shift-based source labels and similarities between target unlabeled instances. Results on a benchmark sentiment analysis dataset indicate state-of-the-art adaptation results. A lower bound on the performance of the proposed algorithm is also presented. One direction for future research is in generative models where modeling can be based on prior knowledge along with probabilistic covariate shift.

References

Ben-David, S., and Schuller, R. 2003. Exploiting task relatedness for multiple task learning. *Conference on Learning Theory (COLT)* 567–580.

Ben-David, S., and Urner, R. 2012. On the hardness of domain adaptation and the utility of unlabeled target samples. *Algorithmic Learning Theory (ALT)* 139–153.

Ben-David, S., and Urner, R. 2014. Domain adaptation - can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence* 70(3):185–202.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems (NIPS)* 137–144.

Ben-David, S.; Blitzer, S.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. 2010. A theory of learning from different domains. *Machine learning* 79(2):151–175.

Bergamo, A., and Torresani, L. 2010. Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach. *Advances in neural information processing systems (NIPS)* 181–189.

Bickel, S.; Bruckner, M.; and Scheffer, T. 2007. Discriminative learning for differing training and test distributions. *International Conference on Machine Learning (ICML)* 81–88.

Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Wortman, J. 2008. Learning bounds for domain adaptation. *Advances in neural information processing systems (NIPS)* 129–136.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. *Association for Computational Linguistics (ACL)*.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. *EMNLP* 120–128.

Blum, A., and Chawla, S. 2001. Learning from labeled and unlabeled data using graph mincut. *International Conference on Machine Learning (ICML)* 19–26.

Chen, M.; Weinberger, K.; and Blitzer, J. 2011. Co-training for domain adaptation. *Advances in neural information processing systems (NIPS)* 2456–2464.

Chen, M.; Xu, Z.; and Weinberger, K. 2012. Marginalized denoising autoencoders for domain adaptation. *International Conference on Machine Learning (ICML)*.

Cortes, C.; Mohri, M.; Riley, M.; and Rostamizadeh, A. 2008. Sample selection bias correction theory. *Algorithmic Learning Theory (ALT)* 38–53.

Cortes, C.; Mansour, Y.; and Mohri, M. 2010. Learning bounds for importance weighting. *Advances in neural information processing systems (NIPS)* 442–450.

Crammer, K.; Kearns, M.; and Wortman, J. 2006. Learning from data of variable quality. *Advances in neural information processing systems (NIPS)* 18.

Crammer, K.; Kearns, M.; and Wortman, J. 2008. Learning from multiple sources. *Journal of Machine Learning Research (JMLR)* 9:1757–1774.

Dai, W.; Yang, Q.; Xue, G.; and Yu, Y. 2007. Boosting for transfer learning. *International Conference on Machine Learning (ICML)* 193–200.

Elisseeff, A., and Pontil, M. 2002. Leave-one-out error and stability of learning algorithms with applications. *NATO-ASI Series on Learning Theory and Practice*.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. *International Conference on Machine Learning (ICML)* 513–520.

Hagen, L., and Kahng, A. 1992. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. on Computer-Aided Design* 11:1074–1085.

Huang, J.; Gretton, A.; Borgwardt, K.; Schoelkopf, B.; and Smola, A. 2006. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems (NIPS)* 601–608.

Joachims, T. 1999a. SVM-Light: Support vector machine. *Cornell Univ.*

Joachims, T. 1999b. Transductive inference for text classification using support vector machines. *International Conference on Machine Learning (ICML)* 200–209.

Kifer, D.; Ben-David, S.; and Gehrke, J. 2004. Detecting change in data streams. *VLDB* 180–191.

Luntz, A., and Brailovsky, V. 1969. On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetica* 3.

Luxburg, U. V. 2007. A tutorial on spectral clustering. *Stat. and Computing* 17:395–416.

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2009. Domain adaptation: Learning bounds and algorithms. *Conference on Learning Theory (COLT)*.

Maurer, A. 2005. Algorithmic stability and meta-learning. *Journal of Machine Learning Research (JMLR)* 967–994.

Platt, J. C. 1999. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods* 185–208.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. *European Conference on Computer Vision (ECCV)*.

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

Schweikert, G.; Ratsch, G.; Widmer, C.; and Scholkopf, B. 2009. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. *Advances in neural information processing systems (NIPS)* 1433–1440.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Sugiyama, M., and Mueller, K. 2005. Generalization error estimation under covariate shift. *Workshop on Information-Based Induction Sciences*.

Sugiyama, M.; Krauledat, M.; and Mueller, K. 2007. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research (JMLR)* (8):985–1005.

Thet, T. T.; Jin-Cheon, N.; Christopher, K.; and Subbaraj, S. 2009. Sentiment analysis of movie reviews on discussion boards using a linguistic approach. *CIKM workshop on Topic-sentiment analysis for mass opinion* 1:81–84.