Domain Adaptation with Coupled Subspaces

John Blitzer
Google Research

Dean Foster
University of Pennsylvania

Sham Kakade
University of Pennsylvania

Abstract

Domain adaptation algorithms address a key issue in applied machine learning: How can we train a system under a *source* distribution but achieve high performance under a different *target* distribution? We tackle this question for divergent distributions where crucial predictive target features may not even have support under the source distribution. In this setting, the key intuition is that that if we can link target-specific features to source features, we can learn effectively using only source labeled data. We formalize this intuition, as well as the assumptions under which such coupled learning is possible. This allows us to give finite sample target error bounds (using only source training data) and an algorithm which performs at the state-of-the-art on two natural language processing adaptation tasks which are characterized by novel target features.

1 Introduction

The supervised learning paradigm of training and testing on identical distributions has provided a powerful abstraction for developing and analyzing learning algorithms. In many natural applications, though, we train our algorithm on a source distribution, but we desire high performance on target distributions which differ from that source [20, 32, 6, 28]. This is the problem of domain adaptation, which plays a central role in fields such as speech recognition [25], computational biology [26], natural language processing [8, 11, 16], and web search [9, 15].[1]

In this paper, we address a domain adaptation setting that is common in the natural language processing literature. Our target domain contains crucial predictive features such as words or phrases that do not have support under the source distribution. Figure 1 shows two tasks which exemplify this condition. The left-hand side is a product review prediction task [7, 12, 28]. The instances consist of reviews of different different products from Amazon.com, together with the rating given to the product by the reviewer (1-5 stars). The adaptation task is to build a regression model (for number of stars) from reviews of one product type and apply it to another. In the example shown, the target domain (kitchen appliances) contains phrases like *a breeze* which are positive predictors but not present in the source domain.

The right-hand side of Figure 1 is an example of a part of speech (PoS) tagging task [31, 8, 19]. The instances consist of sequences of words, together with their tags (noun, verb, adjective, preposition etc). The adaptation task is to build a tagging model from annotated Wall Street Journal (WSJ) text and apply it to biomedical abstracts (BIO). In the example shown, BIO text contains words like *opioid* that are not present in the WSJ.

While at first glance using unique target features without labeled target data may may seem impossible, there is a body of empirical work achieving good performance in this setting [8, 16, 19]. Such approaches are often referred to as *unsupervised* adaptation methods [17], and the intuition they have in common is that it is possible to use unlabeled target data to couple the weights for novel features to weights for features which are common across domains. For example, in the sentiment data set, the phrase *a breeze* may co-occur with the words *excellent* and *good* and the phrase *highly recommended*. Since these words are used to express positive sentiment about books, we build a representation from unlabeled target data which couples the weight for *a breeze* with the weights for these features.

In contrast to the empirical work, previous theoretical work in unsupervised adaptation has focused on two settings. Either the source and target distributions share support [18, 20, 10], or they have low divergence for a specific hypothesis class [6, 28]. In the first setting, instance weighting algorithms can achieve asymptotically target-optimal performance. In the second, it is possible to give finite sample error bounds for specific hypothesis classes (although the models are not in general target-optimal).

[1] Jiang [22] provides a good overview of domain adaptation settings and models

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

Sentiment Classification		Part of Speech Tagging			
Books		**Financial News**			
Positive:	*packed with fascinating info*	NN	VB	VB	NN
Negative:	*plot is very predictable*	*funds*	*are*	*attracting*	*investors*
Kitchen Appliances		**Biomedical Abstracts**			
Positive:	*a breeze to clean up*	NN	PP	ADJ	NN
Negative:	*leaking on my countertop*	*expression*	*of*	*opioid*	*receptors*

Figure 1: Examples from two natural language processing adaptation tasks, where the target distributions contain words (in red) that do not have support under the source distribution. Words colored in blue and red are unique to the source and target domains, respectively. Sentiment classification is a binary (positive vs. negative) classification problem. Part of speech tagging is a sequence labeling task, where NN indicates noun, PP indicates preposition, VB indicates verb, etc.

Neither setting addresses the use of target-specific features, though, and instance weighting is known to perform poorly in situations where target-specific features are important for good performance [21, 22].

The primary contribution of this work is to formalize assumptions that: 1) allow for transferring an accurate classifier from our source domain to an accurate classifier on the target domain and 2) are capable of using novel features from the target domain. Based on these assumptions, we give a simple algorithm that builds a coupled linear subspace from unlabeled (source and target) data, as well as a more direct justification for previous "shared representation" empirical domain adaptation work [8, 16, 19]. We also give finite source sample target error bounds that depend on how the covariance structure of the coupled subspace relates to novel features in the target distribution.

We demonstrate the performance of our algorithm on the sentiment classification and part of speech tagging tasks illustrated in Figure 1. Our algorithm gives consistent performance improvements from learning a model on source labeled data and testing on a different target distribution. Furthermore, incorporating small amounts of target data (also called *semi-supervised* adaptation) is straightforward under our model, since our representation automatically incorporates target data along those directions of the shared subspace where it is needed most. In both of these cases, we perform comparable to state-of-the-art algorithms which also exploit target-specific features.

2 Setting

Our input $X \in \mathcal{X}$ are vectors, where $\mathcal{X}$ is a vector space. Our output $Y \in \mathbb{R}$. Each domain $D = d$ defines a joint distribution $\Pr[X, Y|D = d]$ (where the domains are either source $D = s$ or target $D = t$). Our first assumption is a stronger version of the covariate shift assumption [18, 20, 10]. That is, there exists a *single* good linear predictor for both domains:

Assumption 1. *(Identical Tasks) Assume there there is a vector β so that for $d \in s, t$:*

$$\mathbb{E}[Y|X, D = d] = \beta \cdot X$$

This assumption may seem overly strong, and for low-dimensional spaces it often is. As we show in section 5.5, though, for our tasks it holds, at least approximately.

Now suppose we have a labeled training data $T = \{(x, y)\}$ on the source domain s, and we desire to perform well on our target domain t. Let us examine what is transferred by using the naive algorithm of simply minimizing the square loss on the source domain.

Roughly speaking, using samples from the source domain s, we can estimate β in only those directions in which X varies on domain s. To make this precise, define the *principal subspace* $\mathcal{X}_d$ for a domain d as the (lowest dimensional) subspace of $\mathcal{X}$ such that $X \in \mathcal{X}_d$ with probability 1.

There are three natural subspaces between the source domain s and target domain t; the part which is shared and the parts specific to each. More precisely, define the shared subspace for two domains s and t as $\mathcal{X}_{s,t} = \mathcal{X}_s \cap \mathcal{X}_t$ (the intersection of the principal subspaces, which is itself a subspace). We can decompose any vector x into the vector $x = [x]_{s,t} + [x]_{s,\perp} + [x]_{t,\perp}$, where the latter two vectors are the projections of x which lie off the shared subspace (Our use of the "$\perp$" notation is justified since one can choose an inner product space where these components are orthogonal, though our analysis does not explicitly assume any inner product space on $\mathcal{X}$). We can view the naive algorithm as fitting three components, $[w]_{s,t}$, $[w]_{s,\perp}$, and $[w]_{t,\perp}$, where the prediction is of the form:

$$[w]_{s,t} \cdot [x]_{s,t} + [w]_{s,\perp} \cdot [x]_{s,\perp} + [w]_{t,\perp} \cdot [x]_{t,\perp}$$

Here, with only source data, this would result in an unspecified estimate of $[w]_{t,\perp}$ as $[x]_{t,\perp} = 0$ for $x \in \mathcal{X}_s$. Furthermore, the naive algorithm would only learn weights on $[x]_{s,t}$ (and it is this weight, on what is shared, which is what transfers to the target domain).

Certainly, without further assumptions, we would not expect to be able to learn how to utilize $[x]_{t,\perp}$ with only training data from the source. However, as discussed in the introduction, we might hope that with unlabeled data, we would be able to "couple" the learning of features in $[x]_{t,\perp}$ to those on $[x]_{s,t}$.

2.1 Unsupervised Learning and Dimensionality Reduction

Our second assumption specifies a means by which this coupling may occur. Given a domain d, there are a number of semi-supervised methods which seek to find a projection to a subspace $\mathcal{X}_d$, which loses little predictive information about the target. In fact, much of the focus on un- (and semi-)supervised dimensionality reduction is on finding projections of the input space which lose little predictive power about the target. We idealize this with the following assumption.

Assumption 2. *(Dimensionality Reduction) For $d \in \{s,t\}$, assume there is a projection operator [2] Π_d and a vector β_d such that*

$$\mathbb{E}[Y|X, D = d] = \beta_d \cdot (\Pi_d X)\,.$$

Furthermore, as Π_t need only be specified on $\mathcal{X}_t$ for this assumption, we can specify the target projection operator so that $\Pi_t[x]_{s,\perp} = 0$ (for convenience).

Implicitly, we assume that Π_s and Π_t can be learned from unlabeled data, and being able to do so is is crucial to the practical success of an adaptation algorithm in this setting. Practically, we already know this is possible from empirical adaptation work [8, 16, 19].

3 Adaptation Algorithm

Under Assumptions 1 and 2 and given labeled source data and unlabeled source and target data, the high-level view of our algorithm is as follows: First, estimate Π_s and Π_t from unlabeled source and target data. Then, use Π_s and Π_t to learn a target predictor from source data. We begin by giving one algorithm for estimating Π_s and Π_t, but we emphasize that any Π_s and Π_t which satisfy Assumption 2 are appropriate under our theory. Our focus here is to show how we can exploit these projections to achieve good target results, and in Section 5.1 we analyze and evaluate the structural correspondence learning [8] method, as well.

[2] Recall, that M is a projection operator if M is a linear and if M is idempotent, i.e. $M^2 x = Mx$

Input: Unlabeled source and target data x_s, x_t.
Output: Π_s, Π_t

1. For source and target domains d,
   a. $\forall x_d$, Divide x_d into multiple views $x_d^{(1)}$ and $x_d^{(2)}$.
   b. Choose $k < min(D_1, D_2)$ features from each view $\left(x_{d,i_j}^{(1)}\right)_{j=1}^{k}$.
   c. Construct the $D_1 \times k$ and $D_2 \times k$ cross-correlation matrices C^{12} & C^{21}, where $C_{ij}^{12} = \frac{x_{d,i}^{(1)} x_{d,i_j}^{(2)}}{\sqrt{x_{d,i}^{(1)} x_{d,i}^{(1)} x_{d,i_j}^{(2)} x_{d,i_j}^{(2)}}}$.
   d. Let $\Pi_d = \begin{bmatrix} \Pi_d^{(1)} & 0 \\ 0 & \Pi_d^{(2)} \end{bmatrix}$, where $\Pi_d^{(1)}$ is the outer product of the top left singular vectors of C^{12} (likewise with $\Pi_d^{(2)}$ and C^{21}.
2. Return Π_s and Π_t.

Figure 2: Algorithm for learning Π_s and Π_t.

3.1 Estimating Π_s, Π_t and $[x]_{s,\perp}$

Figure 2 describes the algorithm we use for learning Π_s and Π_t. It is modeled after the approximate canonical correlation analysis algorithm of Ando et al. [2, 23, 14], which also forms the basis of the SCL domain adaptation algorithm [8]. CCA is a multiple-view dimensionality reduction algorithm, so we begin by breaking up each instance into two views (1a). For the sentiment task, we split the feature space up randomly, with half of the features in one view and half in the other. For the PoS task, we build representations for each word in the sequence by dividing up features into those that describe the current word and those that describe its context (the surrounding words).

After defining multiple views, we build the the cross-correlation matrix between views. For our tasks, where features describe words or bigrams, each view can be hundreds of thousands of dimensions. The cross-correlation matrices are dense and too large to fit in memory, so we adopt an approximation technique from Ando et al. [2, 14]. This requires choosing k representative features and building a low-rank cross-correlation matrix from these (1b). Normally, we would normalize by whitening using the within-view covariance matrix. Instead of this, we use simple correlations, which are much faster to compute (requiring only a single pass over the data) and worked just as well in our experiments (1c). The singular value decomposition of the cross-correlation matrix yields the top canonical correlation directions, which we combine to form Π_s and Π_t (1d).

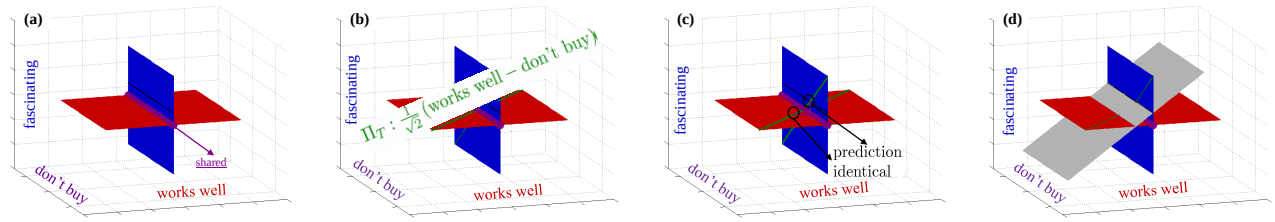

Figure 3: Depiction of how Equation 1 allows us to build an optimal target predictor from source data. (a) defines a 3-dimensional space, where the purple z-axis is shared across source and target domains. (b) shows a particular projection Π_t which couples the target-specific feature *works well* with the shared feature *don't buy*. Under Assumptions 1 and 2, (c) shows that the optimal predictor must assign weight to *works well*, even though it is not observed in the source domain. (d) shows the level set of a linear predictor consistent with our assumptions.

3.2 Estimating a Target Predictor from Source Data

Given Π_t and Π_s, our algorithm fits a linear predictor of the following form from source labeled data:

$$w_t \Pi_t x \; + \; w_s \Pi_s [x]_{s,\perp} \tag{1}$$

where w_t and w_s are the parameters. Recall that $[x]_{s,\perp}$ is the part of the source domain which cannot be represented by the target project Π_t. Computing this exactly is difficult, but we can approximate it here as follows: Let P_{st} be a $D \times D$ diagonal matrix with

$$P_{st,ii} = \begin{cases} 1, & x_i \text{ exists in } \mathcal{X}_s, \mathcal{X}_t \\ 0, & \text{otherwise} \end{cases}$$

Then set $[x]_{s,\perp}$ to be $(I - P_{s,t})\Pi_s$.[3]

Before we move on, we note that the combination of Figure 2 and Equation 1 is our algorithm, which we henceforth refer to as *coupled*. We simply apply the predictor from Equation 1 to the target data. Figure 3 gives some intuition about why this predictor can perform optimally on the target domain. Suppose we want to predict sentiment, and we have two domains in a three-dimensional space, shown in Figure 3(a). The source domain (blue plane) has the features *fascinating* and *don't buy*. The target domain (red plane) has the features *works well* and *don't buy*. Since we have never observed the phrase *works well* in the source, this direction is novel (i.e. it lies in $[x]_{t,\perp}$).

Now suppose we find directions Π_s and Π_t, the green lines in Figure 3(b). Π_t couples *works well* with the negative of *don't buy*. Since *don't buy* is shared with the source domain, we can effectively map source points (containing *fascinating*) to target points (containing *works well*). Under Assumption 1 and 2, we know that the projections of these points onto the shared space must have the same predictions, since they map to the same point. Any linear predictor consistent with both assumptions (e.g. that from Figure 3(d)) is forced to put weight on the novel part of the target domain, $[x]_{t,\perp}$.

Since Figure 3 is three-dimensional, we cannot directly represent $\Pi_s[x]_{s,\perp}$, those source directions which are predictive, but may not be shared with the target. Although they won't appear in the target, we must estimate weights for them in order to correctly calibrate the weights for the shared subspace $\mathcal{X}_{s,t}$. Finally, there may be directions $\Pi_t[x]_{t,\perp}$ that cannot be learned, even from an infinite amount of source data, which do not appear in Equation 1. These directions essentially bias our source predictor with respect to the target domain.

The high-level argument from the previous paragraphs can be formalized in the following *soundness* lemma, which shows that

1. An optimal source linear predictor can always be written in the form of Equation 1.
2. With infinite source data, an optimal target linear predictor always has w_t from Equation 1 as the weight for the shared part of each instance $[x]_{s,t}$.

Lemma 3. *(Soundness) For $d = s$ and $d = t$, we have that:*

$$\mathbb{E}[Y|X, D = d] = \beta_t \Pi_t x \; + \; \beta_s \Pi_s [x]_{s,\perp}$$

Proof. First, by our projection assumption, the optimal predictors are:

$$\begin{aligned} \mathbb{E}[Y|X, D = s] &= \beta_s \Pi_s [x]_{s,t} \; + \; \beta_s \Pi_s [x]_{s,\perp} \; + \; 0 \\ \mathbb{E}[Y|X, D = t] &= \beta_t \Pi_t [x]_{s,t} \; + \; 0 \; + \; \beta_t \Pi_t [x]_{t,\perp} \end{aligned}$$

Now, in our domain adaptation setting (where $E[Y|X, D = d]$ is linear in X), we have must have that the weights on $x_{s,t}$ agree, so that:

$$\beta_s \Pi_s [x]_{s,t} = \beta_t \Pi_t [x]_{s,t}$$

for all x.

[3] This approximation is not exact because these source-unique features may also be partially coupled with the shared subspace, but it performs well in practice.

For $d = t$, the above holds since $[x]_{s,\perp} = 0$ for $x \in \mathcal{X}_t$. For $d = s$, we have $\Pi_t x = \Pi_t[x]_{s,t} + \Pi_t[x]_{s,\perp} = \Pi_t[x]_{s,t}$ for $x \in \mathcal{X}_s$, since Π_t is null on $[x]_{s,\perp}$ (as discussed in Assumption 2). □

In the next section, we will prove two important consequences of Lemma 3, demonstrating when we can learn a perfect target predictor from only source training data and at what rate (in terms of source data) this predictor will converge.

4 Learning Bounds for the Coupled Representation

We begin by stating when we converge to a perfect target predictor on the target domain with a sufficiently large labeled source sample.

Theorem 4. *(Perfect Transfer) Suppose $\Pi_t \mathcal{X}_{s,t} = \Pi_t \mathcal{X}_t$. Then any weight vector (w_t, w_s) on the coupled representation which is optimal on the source, is also optimal on the target.*

Proof. If (w_t, w_s) provides an optimal prediction on s, then this uniquely (and correctly) specifies the linear map on $\mathcal{X}_{s,t}$. Hence, w_t is such that $w_t \Pi_t[x]_{s,t}$ is correct for all x, e.g. $w_t \Pi_t[x]_{s,t} = \beta[x]_{s,t}$ (where β is as defined in Assumption 1). This implies that w_t has been correctly specified in $\dim(\Pi_t \mathcal{X}_{s,t})$ directions. By assumption, this implies that all directions for w_t have been specified, as $\Pi_t \mathcal{X}_{s,t} = \Pi_t \mathcal{X}_t$ □

Our next theorem describes the ability of our algorithm to generalize from finite training data (which could consist of only source samples or a mix of samples from the source and target). For the theorem, we condition on the inputs x in our training set (e.g. we work in a fixed design setting). In the fixed design setting, the randomization is only over the Y values for these fixed inputs. Define the following two covariance matrices:

$$\begin{aligned} \Sigma_t &= \mathbb{E}[\,(\Pi_t x)(\Pi_t x)^\top | D = t], \\ \Sigma_{s \to t} &= \frac{1}{n} \sum_{x \in T_s} (\Pi_t x)(\Pi_t x)^\top \end{aligned}$$

Roughly speaking, $\Sigma_{s \to t}$ specifies how the training inputs vary in the relevant target directions.

Theorem 5. *(Generalization) Assume that $\mathrm{Var}(Y|X) \leq 1$. Let: our coordinate system be such that $\Sigma_t = I$; $\mathcal{L}_t(w)$ be the square loss on the target domain; and $(\hat{w}_t, \hat{w}_s)$ be the empirical risk minimizer with a training sample of size n. Then our expected regret is:*

$$\mathbb{E}[\mathcal{L}_t(\hat{w}_t, \hat{w}_s)] - \mathcal{L}_t(\beta_t, \beta_s) \leq \frac{\sum_i \frac{1}{\lambda_i}}{n}$$

where λ_i are the eigenvalues of $\Sigma_{s \to t}$ and the expectation is with respect to random samples of Y on the fixed training inputs.

The proof is in Appendix A. For the above bound to be meaningful we need the eigenvalues λ_i to be nonzero – this amounts to having variance in all the directions in $\Pi_t \mathcal{X}_t$ (as this is the subspace corresponding to target error covariance matrix Σ_t). It is possible to include a bias term for our bound (as a function of β_t) in the case when some $\lambda_i = 0$, though due to space constraints, this is not provided. Finally, we note that incorporating target data is straightforward under this model. When $\Sigma_t = I$, adding target data will (often significantly) reduce the inverse eigenvalues of $\Sigma_{s \to t}$, providing for better generalization. We demonstrate in Section 5 how simply combining source and target labeled data can provide improved results in our model.

We briefly compare our bound to the adaptation generalization results of Ben-David et al. [4] and Mansour et al. [27]. These bounds factor as an approximation term that goes to 0 as the amount of source data goes to infinity and a bias term that depends on the divergence between the two distributions. If perfect transfer (Theorem 4) is possible, then our bound will converge to 0 without bias. Note that Theorem 4 can hold even when there is large divergence between the source and target domains, as measured by Ben-David et al. [4] and Mansour et al. [27]. On the other hand, there may be situations where for finite source samples our bound is much larger due to small eigenvalues of $\Sigma_{s \to t}$.

5 Experiments

We evaluate our coupled learning algorithm (Equation 1) together with several other domain adaptation algorithms on the sentiment classification and part of speech tagging tasks illustrated in Figure 1. The sentiment prediction task [7, 28, 12] consists of reviews of four different types of products: books, DVDs, electronics, and kitchen appliances from Amazon.com. Each review is associated with a rating (1-5 stars), which we will try to predict. The smallest product type (kitchen appliances) contains approximately 6,000 reviews. The original feature space of unigrams and bigrams is on average approximately 100,000 dimensional. We treat sentiment prediction as a regression problem, where the goal is to predict the number of stars, and we measure square loss.

The part-of-speech tagging data set [8, 19, 30] is a much larger data set. The two domains are articles from the Wall Street Journal (WSJ) and biomedical abstracts from MEDLINE (BIO). The task is to annotate words with one of 39 tags. For each domain, we have approximately 2.5 million words of raw text (which we use to learn Π_s and Π_t), but the labeling conditions are quite asymmetric. The WSJ corpus contains the Penn Treebank corpus of 1 million an-

notated words [29]. The BIO corpus contains only approximately 25 thousand annotated words, however.

We model sentences using a first-order conditional random field (CRF) tagger [24]. For each word, we extract features from the word itself and its immediate one-word left and right context. As an example context, in Figure 1, the window around the word *opioid* is *of* on the left and *receptors* on the right. The original feature space consists of these words, along with character prefixes and suffixes and is approximately 200,000 dimensional. Combined with 39^2 tags, this gives approximately 300 million parameters to estimate in the original feature space. The CRF does not minimize square loss, so Theorem 5 cannot be used directly to bound its error. Nonetheless, we can still run the *coupled* algorithm from Equation 1 and measure its error.

There are two hyper-parameters of the algorithm from Figure 2. These are the number of features k we choose when we compute the cross-correlation matrix and the dimensionality of Π_s and Π_t. k is set to 1000 for both tasks. For sentiment classification, we chose a 100-dimensional representation. For part of speech tagging, we chose a 200-dimensional representation for each word (left, middle, and right). We use these throughout all of our experiments, but in preliminary investigation the results of our algorithm were fairly stable (similar to those of Ando and Zhang [1]) across different settings of this dimensionality .

5.1 Adaptation Models

Here we briefly describe the models we evaluated in this work. Not all of them appear in the subsequent figures.

Naïve. The most straightforward model ignores the target data and trains a model on the source data alone.

Ignore source-specific features. If we believed that the gap in target domain performance was primarily due to source-specific features, rather than target-specific features, we might consider simply discarding those features in the source domain which don't appear in the target. Our theory indicates that these can still be helpful (Lemma 3 no longer holds without them), and discarding these features never helped in any experiment. Because of this, we do not report any numbers for this model.

Instance Weighting. Instance weighting approaches to adaptation [20, 5] are asymptotically optimal and can perform extremely well when we have low-dimensional spaces. They are not designed for the case when new target domain features appear, though. Indeed, sample selection bias correction theory [20, 28] does not yield meaningful results when distributions do not share support. We applied the instance weighting method of Bickel [5] to the sentiment data and did not observe consistent improvement over the naïve baseline. For the part of speech tagging, we did not apply instance weighting, but we note the work of Jiang [21], who experimented with instance weighting schemes for this task and saw no improvement over a naïve baseline. We do not report instance weighting results here.

Use Π_t. One approach to domain adaptation is to treat it as a semi-supervised learning problem. To do this, we simply estimate a prediction $w_t \Pi_t x$ for $x \in \mathcal{X}_s$, ignoring source-specific features. According to Equation 1, this will perform worse than accounting for $[x]_{s,\perp}$, but it can still capture important target-specific information. We note that this is essentially the semi-supervised algorithm of Ando et al. [2], treating the target data as unlabeled.

Coupled. This method estimates Π_s, Π_t, and $[x]_{s,\perp}$ using the algorithm in Figure 2. Then it builds a target predictor following Equation 1 and uses this for target prediction.

Correspondence. This is our re-implementation of the structural correspondence learning (SCL) algorithm of [8]. This algorithm learns a projection similar to the one from Figure 2, but with two differences. First, it concatenates source and target data and learns a single projection Π. Second, it only uses, as its k representative features from each view, features which are shared across domains.

One way to view SCL under our theory is to divide Π into Π_s and Π_t by copying it and discarding the target-specific features from the first copy and the source-specific features from the second copy. With this in hand, the rest of SCL is just following Equation 1. At a high level, *correspondence* can perform better than *coupled* when the shared space is large and *coupled* ignores some of it. *Coupled* can perform better when the shared space is small, in which case it models domain-specific spaces $[x]_{s,\perp}, [x]_{t,\perp}$ more accurately.

5.2 Adaptation with Source Only

We begin by evaluating the target performance of our coupled learning algorithm when learning only from labeled source data. Figure 4 shows that all of the algorithms which learn some representation for new target features never perform worse than the naïve baseline. *Coupled* never performs worse than the semi-supervised Π_t approach, and *correspondence* performs worse only in one pair (*DVDs* to *electronics*). It is also worth mentioning that certain pairs of domains overlap more than others. Book and DVD reviews tend to share vocabulary. So do kitchen appliance and electronics reviews. Across these two groups (e.g. books versus kitchen appliances), reviews do not share a large amount of vocabulary. For the eight pairs of domains which do not share significant vocabulary, the error bars of *coupled* and the naïve baseline do not overlap, indicating that *coupled* consistently outperforms the baseline.

Figure 5 illustrates the coupled learner for part of speech tagging. In this case, the variance among experiments is much smaller due to the larger training data. Once again, *coupled* always improves over the naïve model. Because

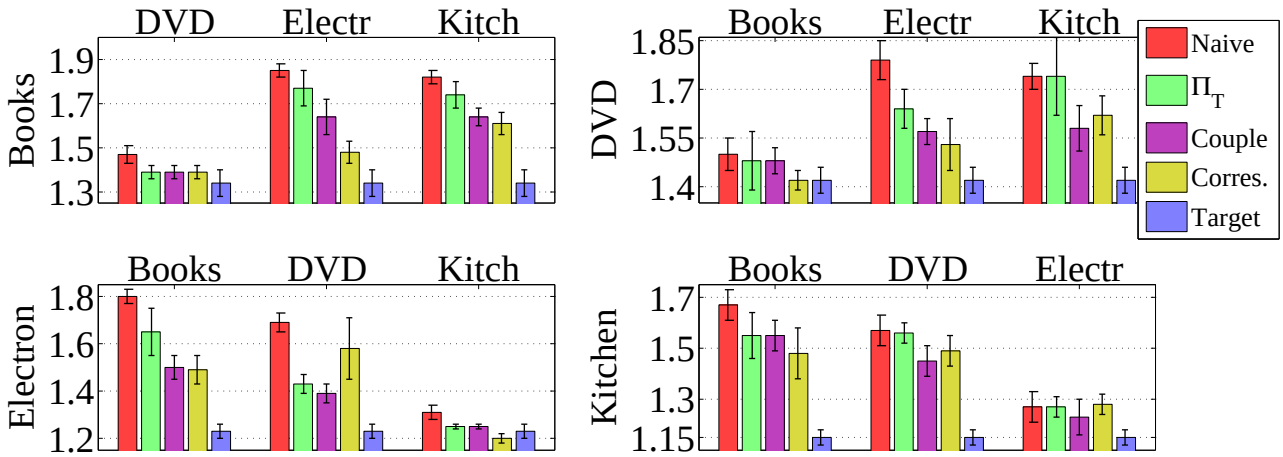

Figure 4: Squared error for the sentiment data (1-5 stars). Each of the four graphs shows results for a single target domain, which is labeled on the Y-axis. Clockwise from top left are books, dvds, kitchen, and electronics. Each group of five bars represents one pair of domains, and the error bars indicate the standard deviation over 10 random draws of source training and target test set. The red bar is the naïve algorithm which does not exploit Π_t or Π_s. The green uses $\Pi_t x$ but not $\Pi_s[x]_{s,\perp}$. The purple is the coupled learning algorithm from Equation 1. The yellow is our re-implementation of SCL [8], and the blue uses labeled *target* training data, serving as a ceiling on improvement.

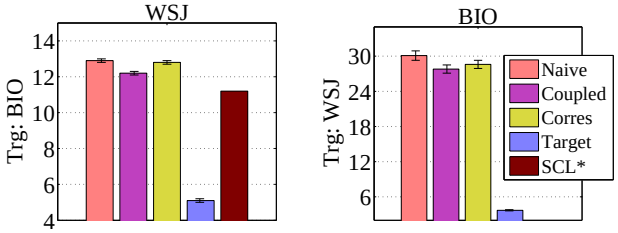

Figure 5: Per-token error for the part of speech tagging task. Left is from WSJ to BIO. Right is from BIO to WSJ. The algorithms are the same as in Figure 4.

of data asymmetry, the WSJ models perform much better on BIO than vice versa. Finally, we also report, for the WSJ→BIO task, the SCL error reported by Blitzer et al. [8]. This error rate is much lower than ours, and we believe this to be due to differences in the features used. They used a 5-word (rather than 3-word) window, included bigrams of suffixes, and performed separate dimensionality reductions for each of 25 feature types. It would almost certainly be helpful to incorporate similar extensions to *coupled*, but that is beyond the scope of this work.

5.3 Adaptation with Source and Target

Our theory indicates that target data can be helpful in stabilizing predictors learned from the source domain, especially when the domains diverge somewhat on the shared subspace. Here we show that our coupled predictors continue to consistently improve over the naïve predictors, even when we do have labeled target training data. Figure 6 demonstrates this for three selected domain pairs. In the case of part of speech tagging, we use all of the available target labeled data, and in this case we see an improvement over the target only model. Since the relative relationship between *coupled* and *correspond* remain constant, we do not depict that here. We also do not show results for all pairs of domains, but these are representative.

Finally, we note that while Blitzer et al. [8, 7] successfully used labeled target data for both of these tasks, they used two different, specialized heuristics for each. In our setting, combining source and target data is immediate from Theorem 5, and simply applying the *coupled* predictor outperforms the baseline for both tasks.

5.4 Use of target-specific features

Here we briefly explore how the coupled learner puts weight on unseen features. One simple test is to measure the relative mass of the weight vector that is devoted to target-specific features under different models. Under the naïve model, this is 0. Under the shared representation, it is the proportion of $w_t\Pi_t$ devoted to genuinely unique features. That is, $\frac{||[w_t\Pi_t]_{t,\perp}||_2^2}{||w_t\Pi_t||_2^2}$. This quantity is on average 9.5% across all sentiment adaptation task pairs and 32% for part of speech tag adaptation. A more qualitative way to observe the use of target specific features is shown in figure 5.4. Here we selected the top target-specific words (never observed in the source) that received high weight under $w_t\Pi_t$. Intuitively, the ability to assign high weight to words like *illustrations* when training on only kitchen appliances can help us generalize better.

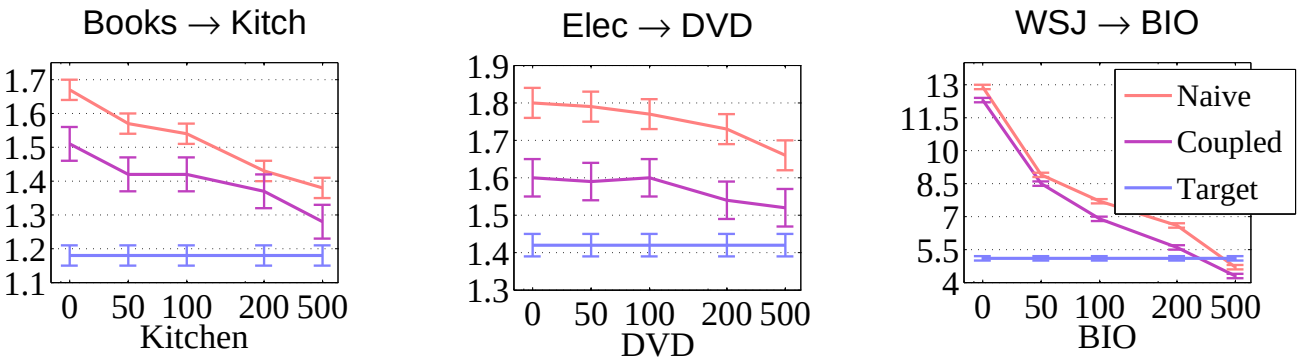

Figure 6: Including target labeled data. Each figure represents one pair of domains. The x axis is the amount of target data.

Adaptation	Negative Target Features	Positive Target Features
Books to Kitch	*mush, bad quality, broke, warranty, coffeemaker*	*dishwasher, evenly, super easy, works great, great product*
Kitch to Books	*critique, trite, religious, the publisher, the author*	*introduction, illustrations, good reference, relationships*

Figure 7: Illustration of how the coupled learner (Equation 1) uses unique target-specific features for the pair of sentiment domains *Books* and *Kitchen*. We train a model using only source data and then find the most positive and negative features that are target specific by examining the weights under $[w_t \Pi_t]_{t,\perp}$.

5.5 Validity of Assumptions

Our theory depends on Assumptions 1 and 2, but we do not expect these assumptions to hold exactly in practice. Both assumptions state a linear mean for $(Y|X)$, and we note that for standard linear regression, much analysis is done under the linear mean assumption, even though it is difficult to test if it holds. In our case, the spirit of our assumptions can be tested independently of the linear mean assumption: Assumption 1 is an idealization of the existence of a single good predictor for both domains, and Assumption 2 is an idealization of the existence of projection operators which do not degrade predictor performance. We show here that both assumptions are reasonable for our domains.

Assumption 1. We empirically test that there there is one simultaneously good predictor on each domain. To see that this is approximately true, we train by mixing both domains, $w^* = \text{argmin}_w [\mathcal{L}_s(w) + \mathcal{L}_t(w)]$, and compare that with a model trained on a single domain. For the domain pair books and kitchen appliances, training a joint predictor on books and kitchen appliance reviews together results in a 1.38 mean squared error on books, versus 1.35 if we train a predictor from books alone. Other sentiment domain pairs are similar. For part-of-speech tagging, measuring error on the Wall Street Journal, we found 4.2% joint error versus 3.7% WSJ-only error. These relatively minor performance differences indicate that one good predictor does exist for both domains.

Assumption 2. We test that the projection operator causes little degradation as opposed to using a complete representation. Using the projection operator, we train as usual, and we compare that with a model trained on the original, high-dimensional feature space. With large amounts of training data, we know that the original feature space is at least as good as the projected feature space. For the electronics domain, the reduced-dimensional representation achieves a 1.23 mean squared error versus a 1.21 for the full representation. Other sentiment domain pairs are similar. For the Wall Street Journal, the reduced dimensional representation achieves 4.8% error versus 3.7% with the original. These differences indicate that we found a good projection operator for sentiment, and a projections operator with minor violations for part of speech tagging.

6 Conclusion

Domain adaptation algorithms have been extensively studied in nearly every field of applied machine learning. What we formalized here, for the first time, is how to adapt from source to target when crucial target features do not have support under the source distribution. Our formalization leads us to suggest a simple algorithm for adaptation based on a low-dimensional coupled subspace. Under natural assumptions, this algorithm allows us to learn a target predictor from labeled source and unlabeled target data.

One area of domain adaptation which is beyond the scope of this work, but which seen much progress recently, is supervised and semi-supervised adaptation [3, 13, 17]. This work focuses explicitly on using labeled data to relax our single-task Assumption 1. Since these methods also make use of shared subspaces, it is natural to consider combinations of them with our coupled subspace approach, and we look forward to exploring these possibilities further.

References

[1] R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, 2005.

[2] R. Ando and T. Zhang. Two-view feature generation model for semi-supervised learning. In *ICML*, 2007.

[3] A. Arygriou, C. Micchelli, M. Pontil, and Y. Yang. A spectral regularization framework for multi-task structure learning. In *NIPS*, 2007.

[4] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007.

[5] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *ICML*, 2007.

[6] J. Blitzer, K. Crammer, A. Kulesza, and F. Pereira. Learning bounds for domain adaptation. In *NIPS*, 2008.

[7] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

[8] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *EMNLP*, 2006.

[9] K. Chen, R. Liu, C.K. Wong, G. Sun, L. Heck, and B. Tseng. Trada: tree based ranking function adaptation. In *CIKM*, 2008.

[10] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. In *ALT*, 2008.

[11] Hal Daumé, III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[12] M. Dredze and K. Crammer. Online methods for multi-domain learning and adaptation. In *EMNLP*, 2008.

[13] Jenny Rose Finkel and Christopher D. Manning. Hierarchical bayesian domain adaptation. In *NAACL*, 2009.

[14] D. Foster, R. Johnson, S. Kakade, and T. Zhang. Multi-view dimensionality reduction via canonical correlation analysis. Technical Report TR-2009-5, TTI-Chicago, 2009.

[15] Jianfeng Gao, Qiang Wu, Chris Burges, Krysta Svore, Yi Su, Nazan Khan, Shalin Shah, and Hongyan Zhou. Model adaptation via model interpolation and boosting for web search ranking. In *EMNLP*, 2009.

[16] Honglei Guo, Huijia Zhu, Zhili Guo, Xiaoxun Zhang, Xian Wu, and Zhong Su. Domain adaptation with latent semantic association for named entity recognition. In *NAACL*, 2009.

[17] A. Saha H. Daume III, A. Kumar. Co-regularization based semi-supervised domain adaptation. In *Neural Information Processing Systems 2010*, 2010.

[18] J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47:153–161, 1979.

[19] F. Huang and A. Yates. Distributional representations for handling sparsity in supervised sequence-labeling. In *ACL*, 2009.

[20] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Schoelkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2007.

[21] J. Jiang and C. Zhai. Instance weighting for domain adaptation. In *ACL*, 2007.

[22] Jing Jiang. A literature survey on domain adaptation of statistical classifiers, 2007.

[23] S. Kakade and D. Foster. Multi-view regression via canonical correlation analysis. In *COLT*, 2007.

[24] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[25] C. Legetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9:171–185, 1995.

[26] Q. Liu, A. Mackey, D. Roos, and F. Pereira. Evigan: a hidden variable model for integrating gene evidence for eukaryotic gene prediction. *Bioinformatics*, 5:597–605, 2008.

[27] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009.

[28] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*, 2009.

[29] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19:313–330, 1993.

[30] PennBioIE. Mining the bibliome project, 2005.

[31] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *EMNLP*, 1996.

[32] G. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *SIGIR*, 2008.

A Appendix

We now prove Theorem 5.

Proof. First, note that our expected regret, when $\Sigma_t = I$ is just:

$$\mathbb{E}\|\hat{w}_t - \beta_t\|^2$$

We can also choose our coordinate system so that $[x]_{s,\perp}$ is (statistically) uncorrelated with $[x]_{s,t}$. Then our estimate of w_t is just $\Sigma_{s\to t}^{-1}\widehat{\mathbb{E}}[(\Pi_t X)Y]$, where

$$\mathbb{E}[(\Pi_t X)Y] = \frac{1}{n}\sum_{(x,y)\in T}(\Pi_t x)y$$

where (x, y) are the values in our training set. Define η_x for x in our training set by $y_x = \mathbb{E}[Y|x] + \eta_x$, where y_x is the value on training sample x. By assumption, $\mathbb{E}\eta_x^2 \leq 1$. If we rotate to a coordinate system where $\Sigma_{s\to t}$ is diagonal, then:

$$\begin{aligned}
\mathbb{E}\|\hat{w}_t - \beta_t\|^2 &= \mathbb{E}||\widehat{\mathbb{E}}[(\Pi_t X)Y] - \mathbb{E}[(\Pi_t X)Y]||^2_{\Sigma_{s\to t}^{-2}} \\
&= \mathbb{E}\left[\sum_i \frac{(\widehat{\mathbb{E}}[(\Pi_t X)_i Y] - \mathbb{E}[(\Pi_t X)_i Y])^2}{\lambda_i^2}\right] \\
&= \mathbb{E}\left[\sum_i \frac{\left(\frac{1}{n}\sum_{x\in T}\eta_x(\Pi_t x)_i\right)^2}{\lambda_i^2}\right] \\
&= \mathbb{E}\left[\sum_i \frac{\frac{1}{n^2}\sum_{x\in T}\eta_x^2(\Pi_t x)_i^2}{\lambda_i^2}\right] \\
&\leq \sum_i \frac{\frac{1}{n^2}\sum_{x\in T}(\Pi_t x)_i^2}{\lambda_i^2} \\
&= \frac{1}{n}\sum_i \frac{1}{\lambda_i}
\end{aligned}$$

where the third to last step uses independence and the final step uses the definition of λ_i. □