Domain Adaptation with Coupled Subspaces

Anonymous Author(s)
Affiliation
Address
email

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. The key intuition we formalize is how to link the learning of weights for these target-specific features to source features via a coupled subspace. We formalize the assumptions under which such coupled learning is possible and give finite sample target error bounds (using only source training data). Our algorithm yields good performance on two natural language processing adaptation data sets which are characterized by the presence of novel 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 [18, 29, 4, 25]. This is the problem of domain adaptation, which plays a central role in fields such as speech recognition [22], computational biology [23], natural language processing [6, 9, 14], and web search [7, 13].[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 an example of a product review classification task [5, 10, 25]. The instances in this task 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 at all in the source domain.

The right-hand side of Figure 1 is an example of a part of speech (PoS) tagging task [28, 6, 17]. The instances consist of words and their left and right contexts, together with their tags (noun, verb, adjective, adverb, etc).[2] The adaptation task is to build a tagging model from annotated Wall Street Journal (WSJ) text and apply it to text of another genre, such as biomedical abstracts (BIO). In the example shown, BIO text contains words like *opioid* that we'd like to assign tags to, but which are not present in the WSJ.

While at first glance this may seem impossible, there is a body of empirical work achieving good performance in this setting, by coupling the learning of these novel features to those features which are shared across domains [6, 14, 17]. For example, in the sentiment data set, the phrase *a breeze*

[1] Jiang [20] provides a good overview of domain adaptation settings and models

[2] PoS tagging is often treated as a sequence-modeling task, but the error is measured on a per-token basis

Sentiment Classification	
	Books
Positive:	*packed with fascinating info*
Negative:	*plot is very predictable*
	Kitchen Appliances
Positive:	*a breeze to clean up*
Negative:	*leaking on my countertop*

Part of Speech Tagging			
Financial News			
NN	VB	VB	NN
funds	*are*	*attracting*	*investors*
Biomedical Abstracts			
NN	PP	ADJ	NN
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.

may co-occur with the words *excellent* and *good* and the phrase *highly recommended*. Since these words are also used to express positive sentiment about books, we can build a representation from unlabeled target data which couples the weight for *a breeze* with the weights for these features.

In this work, we formalize assumptions that provably permit the design of algorithms for domain adaptation, which: 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. We also give finite target error bounds (using only source training data or a mix of source and target training data) that depend on how the covariance structure of the coupled subspace relates to novel features in the target distribution (which we precisely specify).

Our work differs from previous treatment of the domain adaptation setting [20] in that we focus on the issue of how to make use of novel features in the target domain. Our bound is similar in spirit to both sample selection bias correction work [15, 18, 8] and recent theoretical analyses of domain adaptation [4, 25]. It differs from the former by accommodating source and target distributions which don't share support and from the latter by giving rates which asymptotically approach 0, even when the distributions diverge on the coupled subspace.

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. Incorporating small amounts of target data is trivial under our model, since our representation automatically incorporates target data along those directions of the shared subspace where it is needed most.

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 $\Pr[X, Y|D = d]$ — while our theory extends to multiple domains, we only consider the source domain $D = s$ and target $D = t$. In the covariate shift model, $\Pr[X|D = d]$ may vary with the domain D, while the conditional distribution $\Pr[Y|X, D = d]$ is not a function of the domain. We consider a slight modification of this setting, under the prevalent assumption that we are working in a high enough dimensional feature space so that a certain linearity assumption is appropriate. Formally, we have:

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

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

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. Let us define the subspaces in which the source and target inputs vary. 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 intro, 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* [3] *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 our adaptation algorithm. Our theory is agnostic to the details of the specific learning algorithm, though. Indeed, the empirical adaptation work that does learn shared representations covers a whole host of techniques, all of which work well for their particular task [6, 14, 17]. We discuss our specific algorithm in further detail in Section 4.1.

3 Generalization under the Coupled Representation

The idea of the algorithm is to construct a shared representation so that learning on the source will force learning on the novel part of the target subspace, e.g. on $[x]_{t,\perp}$. Our algorithm fits a linear predictor of the form:

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

where w_t and w_s are the parameters.

First, the following lemma shows that this representation is *sound*, meaning that our shared representation supports optimal predictions *simultaneously* on both the source and target domains, with $w_t = \beta_t$ and $w_s = \beta_s$.

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

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

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

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

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

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.

For $d = t$, the above clearly holds as $[x]_{s,\perp} = 0$ 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). □

Now, let us provide a little intuition for this representation, before formally showing how perfect transfer is possible using only source data. Note that for our source domain, where $x \in \mathcal{X}_s$, we have that $\Pi_t x = \Pi_t[x]_{s,t}$ (since $\Pi_t[x]_{s,\perp} = 0$, as explained in Assumption 2). Hence, our prediction with x in the source is of the form:

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

Thus, if we seek to use $[x]_{s,t}$ in our prediction, then we *must* place a non-zero weight on at least some components in w_t. However, now the learned weight w_t could have an effect on all of $[x]_{t,\perp}$.

Our first theorem specifies when perfect transfer is possible using only source data (e.g. do we converge to a perfect predictor on the target domain with a sufficiently large sample on the source domain). Using the notation, $\Pi_t\mathcal{X}_t = \{\Pi_t x : x \in \mathcal{X}_t\}$, we have that $\Pi_t\mathcal{X}_t$ is the reduced dimensional predictive subspace for the target domain. Also, note that $\Pi_t\mathcal{X}_{s,t}$ is the subspace of $\Pi_t\mathcal{X}_t$ which is due to the variation in the shared subspace. Note that $\Pi_t\mathcal{X}_{s,t}$ could actually equal $\Pi_t\mathcal{X}_t$. For example, say $\Pi_t X$ is simply the one dimensional projection $3X_1 + 5X_3$, where both X_1 and X_3 are in $\mathcal{X}_t$. Now note that we only need to varyX_1 (or just X_3 or any one-dimensional linear combination of them) to cause $3X_1 + 5X_3$ to vary. In other words, roughly speaking, as long as the shared features are appropriately coupled to new features (through Π_t)), we will have that $\Pi_t\mathcal{X}_{s,t} = \Pi_t\mathcal{X}_t$.

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 is on generalization, when we have a finite training dataset (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), and denote these by T_{training} (of size n). As in the fixed design setting, the randomization is only over the Y values for these fixed inputs. Define the following two covariance matrices:

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

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:*

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

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 to be nonzero — this amounts to having variance in all the directions in $\Pi_t \mathcal{X}_t$ (as this is subspace corresponding to target error covariance matrix Σ_t). Note that some target data could greatly reduce the inverse eigenvalues eigenvalues, thus providing for better generalization.

We briefly compare our bound to the adaptation generalization results of Ben-David et al. [3] and Mansour et al. [24]. The bounds they provide 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. [3] and Mansour et al. [24]. On the other hand, there may be situations where for finite source samples our bound is much larger due to small eigenvalues of $\Sigma_{s \to t}$. Finally we note that if some eigenvalues are 0 (i.e. we are missing some relevant directions), then it is possible to include a bias term for our bound (as a function of β_t), though due to space constraints, this is not provided.

4 Experiments

We evaluate our coupled learning algorithm (Equation 1) on the sentiment classification and part of speech tagging tasks illustrated in Figure 1. The sentiment classification task [5, 25, 10] 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.

The part-of-speech tagging data set [6, 17, 27] 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 annotated words [26]. The BIO corpus contains only approximately 25 thousand annotated words, however.

We model each word and its part of speech the tag of each word separately, conditioned on the word 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.

4.1 Learning Π_s and Π_t

We briefly discuss here how to find the projection operators Π_s and Π_t and how to identify the portion of the source $[x]_{s,\perp}$ that is not related to the target. For finding the projections, we use an approximation to canonical correlation analysis (CCA) [16] for large-scale, high dimensional data [1, 21, 12]. CCA is a multiple-view dimensionality reduction algorithm, so we begin by breaking up each instance into two views. For the sentiment task we define view 1 to be the first half of the document and view 2 to be the second half. For the PoS task, we create a two-view representation by dividing up each segment into content (the word itself) and context (the surrounding words).

After defining multiple views, we search for the reduced dimensional subspace of both views that maximizes the correlation between views across all of the unlabeled data. On the target domain, the output of this procedure are two orthogonal projections $\Pi_t^{(1)}$ and $\Pi_t^{(2)}$ which are also orthogonal to each other. Now we define $\Pi_t = \Pi_t^{(1)} + \Pi_t^{(2)}$. In all sentiment experiments, we set the dimensionality of Π_s and Π_t to be 40. In all the PoS tagging experiments, we set the dimensionality of Π_s and Π_t to be 200 from the context projection and 100 from the content projection, for a total of 300.

Using CCA-induced representations for input to supervised models has been shown to be useful both theoretically and empirically [1, 21]. It is not a contribution of this work, though, and space prevents us from giving a detailed description of the CCA objective and the specific approximation of Ando and Zhang [1], which we use here. We refer the reader to that work for details.

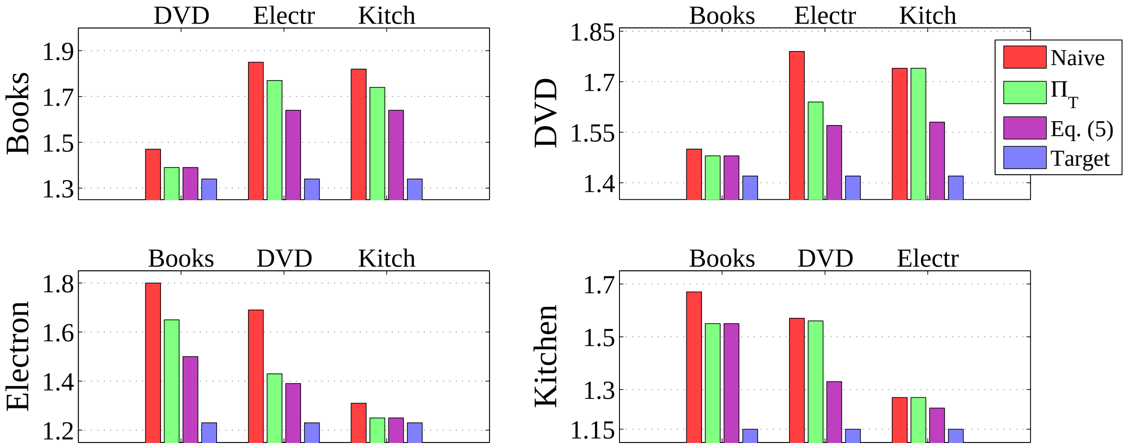

Figure 2: Squared error for the sentiment data (1-5 stars). Each of the four graphs shows results for a single target domain, which is labeled on the Y-axis. Clockwise from top left are books, dvds, kitchen, and electronics. Each group of three bars represents one pair of domains, and the error bars 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 is our coupled learning algorithm with only source data. The blue (train on target) bars are unaffected by source and constant across target domains.

We approximate $[x]_{s,\perp}$ by simply projecting onto just those source-unique features which have no support in the large target unlabeled data. In the sentiment task from Figure 1, for example, the word *fascinating* is just such a source-unique feature.

4.2 Adaptation with Source Only

We begin by evaluating the target performance of our coupled learning algorithm when learning only from labeled source data. Figure 2 shows the performance on sentiment data of the coupled model (in green) versus a naïve model (in red) which simply learns based on the original feature representation, without taking advantage of projections Π_t and Π_s. The blue bars train a model directly on the target data and can be thought of as a "lower bound" on error for adaptation. We defer comparisons with other adaptation algorithms to Section 4.6.

Since we are able to incorporate unseen features via the shared representation, our theory predicts an improvement in target performance. Our algorithm never causes an increase in error and often greatly improves improves over the naïve model. It is also worth mentioning that certain pairs of domains overlap less than others. For example, books and DVDs are highly overlapping in vocabulary usage, but books and kitchen appliances are not. Our method performs especially well (relative to the baseline) for those pairs of very dissimilar domains where there are many new, target-specific features. We explore this further in Section 4.4

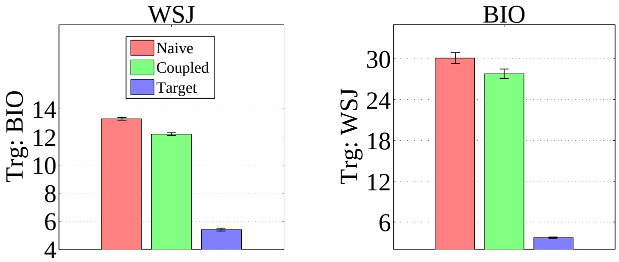

Figure 3: Per-token error for the part of speech tagging task. Left is from WSJ to BIO. Right is from BIO to WSJ. The algorithms are the same as in Figure 2.

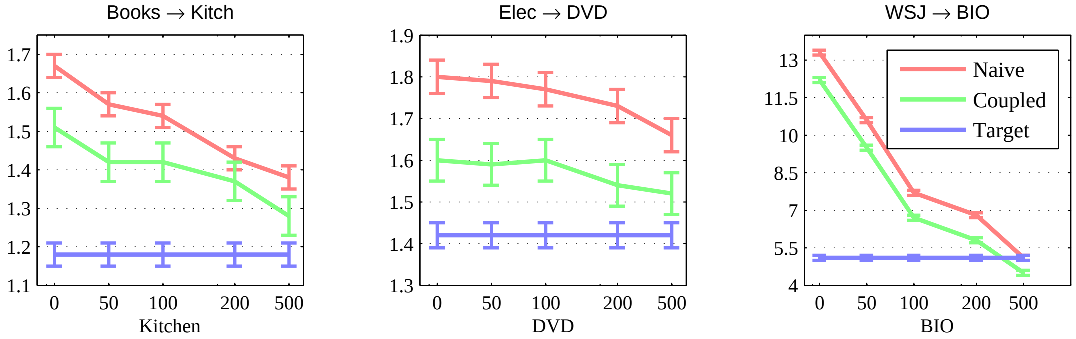

Figure 4: Including some target data. Each figure represents one pair of domains. The x axis indicates the amount of target data.

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*	*wonderful book, introduction to* *illustrations, good reference, relationships*

Figure 5: 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 $[w_t \Pi_t]_{t,\perp}$.

Figure 3 illustrates the 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, our method always improves over the naïve model. Finally, we note that because of data asymmetry, our WSJ models are generally much better than our BIO models. In particular, in the right-hand plot, the BIO source model is trained on only 12,500 words, but the WSJ target model is trained on 1 million.

4.3 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. Of course, we also expect target data to help the naïve algorithm as well, but here we show that our coupled predictors continue to consistently improve over the naive predictors, even when we do have labeled target training data. Figure 4 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 a significant improvement over the target only model (blue curve). Because of space constraints, we don't show results for all pairs of domains, but these results are representative.

4.4 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 4.4. 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.

4.5 Validity of Assumptions

While our theory depends on Assumptions 1 and 2, we do not expect them to hold exactly in practice. Here we examine the extent to which they hold true for our particular tasks. The basic idea is to exploit the fact that we do possess labeled target data for the two data sets to examine source and target performance under different conditions. For Assumption 1, we compare the performance of a joint predictor with the performance of a predictor trained on just one domain. If these two differ by a small amount, then Assumption 1 approximately holds. 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. On kitchen appliances, the joint error is 1.23 versus 1.19 for the individual predictor. For part-of-speech tagging, the results are similar: 4.2 joint error versus 3.7 WSJ-only error on the Wall Street Journal.

For Assumption 2, we performed a similar set of experiments, training with both the reduced-dimensionality representation (under Π_d) and the full feature representation with all the in-domain training data we had. For the WSJ, the reduced dimensional representation achieves 4.2% error (versus 3.7% with the original feature representation). For DVDs, the reduced-dimensional representation achieves a 1.47 mean squared error versus a 1.41 mean squared error for electronics. For electronics, the reduced-dimensional representation achieves a 1.23 mean squared error versus a 1.21 for the full representation.

4.6 Alternative Adaptation Algorithms

Our coupled learning approach performs well for domain adaptation, but it is not the only algorithm designed for this problems. Of the empirical algorithms, the structural correspondence learning work of Blitzer et al. [6] is the most closely-related, but we note that on the same sentiment data set that algorithm did not give as consistent gains as ours (and indeed sometimes hurt performance). The instance weighting algorithms advocated for sample selection bias correction [18, 8] are designed for cases when the distributions share support and are thus not appropriate for our setting, but we do note that Jiang and Zhai [19] report poor performance using instance weighting for the PoS tagging task.

When labeled data does exist, there do exist other algorithms that exploit it by specifically seeking to relax Assumption 1 [2, 9, 11]. These algorithms do not exploit unlabeled target data, however, and thus are not directly comparable to our coupled learner. Indeed, they are complementary and could potentially give gains when used together with our coupled representation, although exploration of that is beyond the scope of this paper.

5 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 an optimal target predictor from labeled source data and unlabeled target data. Furthermore, we show that incorporating small amounts of labeled target data can improve the stability of our target predictor, both theoretically and empirically. We believe our analysis will lead to interesting connections to other work on adaptation and transfer learning, especially supervised shared subspace methods [2, 9].

References

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

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

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

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

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

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

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

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

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

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

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

[12] 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.

[13] 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.

[14] 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.

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

[16] H. Hotelling. The most predictable criterion. *Journal of Educational Psychology*, 1935.

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

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

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

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

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

[22] 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.

[23] 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.

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

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

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

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

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

[29] 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. □