Learning with Minimum Supervision: 
A General Framework for Transductive Transfer Learning

Mohammad Taha Bahadori  
Electrical Engineering Department  
University of Southern California  
Los Angeles, CA  
mohammab@usc.edu

Yan Liu  
Computer Science Department  
University of Southern California  
Los Angeles, CA  
yanliu.cs@usc.edu

Dan Zhang  
Computer Science Department  
Purdue University  
West Lafayette, IN  
danzhang2008@gmail.com

Abstract—Transductive transfer learning is one special type of transfer learning problem, in which abundant labeled examples are available in the source domain and only unlabeled examples are available in the target domain. It easily finds applications in spam filtering, microblogging mining and so on. In this paper, we propose a general framework to solve the problem by mapping the input features in both the source domain and target domain into a shared latent space and simultaneously minimizing the feature reconstruction loss and prediction loss. We develop one specific example of the framework, namely latent large-margin transductive transfer learning (LATTL) algorithm, and analyze its theoretic bound of classification loss via Rademacher complexity. We also provide a unified view of several popular transfer learning algorithms under our framework. Experiment results on one synthetic dataset and three application datasets demonstrate the advantages of the proposed algorithm over the other state-of-the-art ones.

I. INTRODUCTION

Transfer learning is the process of leveraging the information from other domains (i.e., source domain) to train a better model for the target domain [31], [27]. It has been extensively studied in the literature and easily found applications in many learning tasks, such as natural language processing [15], sentiment prediction [5], image classification [29], and intrusion detection [20].

Most transfer learning algorithms have been focused on the learning scenarios as follows [31]: given abundant labeled examples in the source domain and a few labeled examples in the target domain, how can we build a better model to make predictions for examples in the target domain? It has been shown that even a small amount of labeled data can help improve the prediction performance. In parallel, self-taught learning is proposed to successfully handle learning tasks in which only a limited number of labeled examples in the target domain are available, while there is an abundance of unlabeled examples in the source domain [29]. With the development of social media and crowd sourcing, we are usually confronted with another type of challenges in real applications, i.e., given abundant labeled examples in one domain, how can we build effective learning algorithms applicable to many other domains without any additional labeling efforts. For example, in spam filtering, we can get labeled email spams for a small number of users, and the goal is to build a personalized spam filtering classifier for each of the millions of users without soliciting any labels from them; in microblogging mining, the contents change rapidly over time and every hour the tweets could be discussing topics in a new domain; it is easy to obtain abundant tagged documents from social tagging websites (e.g. Delicious.com), while it is impractical to manually label tweets every hour as they come. This learning problem, i.e., abundant labeled example in source domain and unlabeled examples in the target domain, is also known as transductive transfer learning. There have been several studies to address the problem [2], [28], but it is far from being solved for real applications and is therefore the main focus of this paper.

An important problem to consider when developing transfer learning algorithms is what kind of knowledge can actually be transferred across domains. Roughly speaking, the existing algorithms for transfer learning can be summarized into two types: One is to assume a shared prediction function. In other words, we aim at seeking more meaningful feature space in which the prediction function can be applied to. Here the shared feature space can be either latent space [1], [29], [14], [26], [17], or augmented feature space [6], [15]. The other type is to assume shared function families in the original feature space: the goal is to estimate parameters/hyper-parameters for each domain [22], [7], [13], or utilize the relational information to help the estimation [24], [25], [16]. Given that we only have unlabeled examples in the target domain in transductive transfer learning, the first approach is a natural choice.

In this paper, we propose a general framework for transductive transfer learning, which learns different mapping functions (depending on the domains) and the prediction function at the same time by minimizing the reconstruction error and transductive classification error in one unified function. As an example, we develop latent transductive transfer learning (LATTL) model based on the large-margin approach. Different from most transfer learning algorithms,
which assume to have the same mapping functions across domains, LATTL relaxes this assumption and guides the search of mapping functions in the target domain via minimizing the classification loss. As a result, our algorithm is more flexible and does not significantly rely on the strong similarities between the source and target domains. In particular, we provide a theoretical bound for its classification loss in the target domain via Rademacher complexity. We also discuss its connection to other transfer learning algorithms, and demonstrate its effectiveness empirically on both synthetic dataset and application datasets, even when the similarities between the source domain and target domain are relatively low.

The rest of the paper is organized as follows: we first describe our proposed approach in Section II, then discuss the connections of our algorithm to existing transfer learning algorithms in Section III, and show experiment results in Section IV. Finally we summarize the paper and provide hints on future work.

II. RELATED WORKS

Before diving into the explicit formulation of Transductive Transfer Learning, we introduce the two most related algorithms.

A. Transductive SVM

Introduced by Vapnik [32], the Transductive Support Vector Machine (TSVM) attempts to solve the transductive labeling problem defined as following:

\[
\min_{w,b,(\xi_i)} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i + C^* \sum_{i=1}^{m} \xi_i^* \\
\text{subject to:} \\
\forall i \in \{1, \ldots, m\} \quad y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \\
\forall i \in \{m+1, \ldots, m+n\} \quad y_{i}^* (w^T x_i + b) \geq 1 - \xi_i^*, \quad \xi_i^* \geq 0,
\]

where \(w\) is the classifier and \(b\) is the bias component, \((x_i, y_i), i = 1, \ldots, m\) are the samples with their labels and \(x_{i}, i = m+1, \ldots, m+n\) comprise the unlabeled samples. \(y_{i}^*, i = m+1, \ldots, m+n\) are the labels learned for the unlabeled samples; \(C, C^*, \xi_i\) and \(\xi_i^*\) are the costs and hinge losses for the labeled and unlabeled samples, respectively.

Vapnik believes that finding a classification for an specific set of test samples should be easier than finding a general rule [32]. Another justification for the success of TSVM is the cluster assumption, [11], which states that the classifier should not be in the high density cluster region and TSVM tries to enforce this rule. However, the assumption behind TSVM is that both source and target data points are in the same domain which is not the case in our problem statement.

B. Self Taught Learning

Self taught learning (STL) [29] is proposed to solve the problem in which only a limited number of labeled examples in the target domain are available, while there is an abundance of unlabeled examples in the source domain. The algorithm works by learning a set of higher level, succinct bases in the source domain so that these bases can be considered as some “basic elements” that cover all of the examples in the source and target domains with a set of sparse activation coefficients. The concrete formulation of STL is the following:

\[
\min_{\Phi, \{a_i\}} \sum_i \|x_i - \Phi a_i\|^2 + \gamma \|a_i\|_1,
\]

where \([\Phi_1, \ldots, \Phi_r]\) are the \(r\) basis vectors for the latent space corresponding to the source domain and \(\{a_i\}\) are the representations of the source samples in the latent space. Clearly, STL does not utilize the transductive classification benefits. In this paper our goal is to aid the transductive classification by representing the source and target samples in a common latent space.

III. METHODOLOGY

A. Problem Statement and Notations

Given \(n\) number of unlabeled examples \(\{x_1,\ldots,x_n\} \in X\) in the target domain, as well as \(m\) number of labeled examples \(\{(z_1, y_1), (z_2, y_2), \ldots, (z_m, y_m)\} \in \{X, \mathcal{Y}\}\) in the source domain. \(X\) is a \(d\)-dimensional space, and \(\mathcal{Y}\) denotes the label space. And each \(y_i\) can be considered as a \(\ell\)-dimensional label vector, as in multi-label learning problems, with \(y_{ij} \in \{-1,1\}\) and \(\ell\) being the number of classes in the label space. Let \(\{u_1, \ldots, u_n\}, \{\hat{u}_1, \ldots, \hat{u}_n\} \in \mathcal{Y}\), \(u_{ij}, \hat{u}_{ij} \in \{-1,1\}, j = 1, \ldots, \ell\) be the true and predicted labels for the target data, respectively. The main goal of transductive transfer learning is to predict the labels for examples \(\{x_1, x_2, \ldots, x_n\}\) on the target domain.

B. Formulation

Our algorithm for transductive transfer learning works by finding a common latent space shared by both the source domain and the target domain, so that this common subspace can be optimal for both feature reconstruction and classification. More precisely, the objective function of our algorithm can be formulated as follows:

\[
\min_{S, \mathcal{P}} RLoss(Z, X, S) + CLoss(Z, Y, X, S, \mathcal{P}),
\]

where \(S\) is the \(r\)-dimensional shared latent space, \(\mathcal{P}\) is the parameters of classifier \(C\), the reconstruction loss \(RLoss(Z, X, S)\) measures the loss between feature vectors in the original feature space and those in the latent space for the examples in the source domain or target domain, and the classification loss \(CLoss(Z, Y, X, S, \mathcal{P})\) is the loss between the predicted labels by classifiers \(C\) and the ground truth for
the examples in the source domain plus the transductive loss for the examples in the target domain.

Equation (1) provides a general framework which we can use to define different types of transductive transfer learning algorithms. In this paper, we introduce one specific example of the framework, namely latent large-margin transductive transfer learning algorithm (LATTL), which uses regularized squared loss as the reconstruction loss and hinge loss of support vector machines for the reconstruction loss. LATTL is aimed to perform two STL-like latent space reconstructions supervised by a TSVM classification on the reconstructed spaces. Using the SVM formulation, we write the formulation of the LATTL algorithm as following:

\[
\min_{\{w_j, b, \Phi, \tilde{\Phi}, \{y_j^*\}\}} \frac{1}{2} \sum_{j=1}^{\ell} \|w_j\|^2 + C_1 \sum_{i=1}^{m} \eta_i + C_3 \sum_{i=1}^{n} \zeta_i^+ \\
+C_2 \sum_{i=1}^{m} \xi_i + C_4 \sum_{i=1}^{n} \xi_i^+
\]

subject to:

\[
\forall i \in \{1, \ldots, m\}, \quad y_{ij}(w_j^T a_i + b_j) \leq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0
\]

\[
\forall i \in \{1, \ldots, m\}, \quad y_{ij}(w_j^T a_i + b_j) \geq 1 - \xi_{ij}^+, \quad \xi_{ij}^+ \geq 0
\]

\[
\forall i \in \{1, \ldots, n\}, \quad \|x_i - \Phi a_i\|^2 + \beta \|a_i\|^2 \leq \eta_i
\]

\[
\forall i \in \{1, \ldots, m\}, \quad \|x_i - \tilde{\Phi} a_i\|^2 + \beta \|a_i\|^2 \leq \zeta_i
\]

\[
\forall i \in \{1, \ldots, \ell\}, \quad \|\Phi_i\|^2 \leq 1, \quad \|\tilde{\Phi}_i\|^2 \leq 1
\]

(2)

In this optimization problem, there are four types of variables that need to be optimized. \(\Phi, \tilde{\Phi} \in \mathbb{R}^{d \times r}\) are basis vectors for \(r\)-dimensional hidden spaces underlying the source and target domain, respectively. \(\{a_i\}\) and \(\{e_i\}\) are the representations of the source and target samples in the latent space, respectively. The vectors \(w_1, \ldots, w_\ell\) are the multi-label classifiers shared between two latent spaces, where \(b_1, \ldots, b_\ell\) are the corresponding bias values. Finally, \(y_j^*\) is the \(j^{th}\) label predicted for the target sample \(x_i\) during the learning process. The predicted labels are \(\tilde{u}_i = \text{sgn}(f(e_i))\), where \(f(e_i) = w_j^T e_i + b\). \(\beta\) is the normalization parameter, \(\xi_{ij}\) and \(\xi_{ij}^+\) are slack variables, and \(C_1\) - \(C_4\) are the cost parameters that control the reconstruction error and classification error in the source and target domains.

The classifier \(f\) is the same across different domains in the latent space while the mapping functions from the original feature space to latent space is unknown and differ from domain to domain. By simultaneously minimizing the transductive classification loss and reconstruction loss, we are able to recover the classification function \(f\) and the mapping functions. One by-product advantage of mapping the high-dimensional features into lower-dimension latent space is the ease to handle short texts, such as tweets or instant message which, leads to similar effects of latent semantic indexing.

To solve Equation (2), we use an iterative approach, similar to the one discussed in [9], and in each iteration we solve two optimization problems with each defined by fixing one set of variables \(\{\{w_j\}, b, \Phi, \tilde{\Phi}, \{y_j^*\}\}\) or \(\{\{a_i\}, \{e_i\}\}\). In particular, we have the following iterative approach:

1. Fixing \(\{a_i\}\) and \(\{e_i\}\), the resulting optimization problem can be decomposed into three independent subproblems:

\[
\min_{\Phi} \sum_{i=1}^{m} \|z_i - \Phi a_i\|^2\quad \text{subject to:} \forall i \in \{1, \ldots, s\} \|\Phi_i\|^2 \leq 1,
\]

\[
\min_{\tilde{\Phi}} \sum_{i=1}^{m} \|z_i - \tilde{\Phi} e_i\|^2\quad \text{subject to:} \forall i \in \{1, \ldots, s\} \|\tilde{\Phi}_i\|^2 \leq 1
\]

(3) (4)

And for each \(w_j\) and \(b_j\):

\[
\min_{w_j, b_j} \frac{1}{2} \|w_j\|^2 + C_2 \sum_{i=1}^{n} \xi_{ij} + C_3 \sum_{i=1}^{n} \xi_{ij}^+
\]

subject to:

\[
\forall i \in \{1, \ldots, m\}, \quad y_{ij}(w_j^T a_i + b_j) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0
\]

\[
\forall i \in \{1, \ldots, n\}, \quad y_{ij}^*(w_j^T e_i + b_j) \geq 1 - \xi_{ij}^+, \quad \xi_{ij}^+ \geq 0
\]

(5)

Problems (3) and (4) can be solved efficiently by using the Lagrange dual formulation, see [23]. Problem (5) is the TSVM formulation, which can be solved efficiently by one of the methods described in [12], [33], [35], [10].

2. Fixing \(\{w_j\}, b, \Phi, \tilde{\Phi}\) and \(\{y_j^*\}\), the resulting problem can be decomposed to two independent subproblems, that is, for each \(a_i\), we have

\[
\min_{\{a_i\}, \xi_{ij}} C_1 \left(\|z_i - \Phi a_i\|^2 + \beta \|a_i\|^2\right) + C_2 \sum_{j=1}^{\ell} \xi_{ij}
\]

subject to: \(\forall i \in \{1, \ldots, m\}\), \(y_{ij}(w_j^T a_i + b_j) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0\)

and for each \(e_i\):

\[
\min_{\{e_i\}, \xi_{ij}^+} C_3 \left(\|x_i - \tilde{\Phi} e_i\|^2 + \beta \|e_i\|^2\right) + C_4 \sum_{j=1}^{\ell} \xi_{ij}^+
\]

subject to: \(\forall i \in \{1, \ldots, n\}\), \(y_{ij}^*(w_j^T e_i + b_j) \geq 1 - \xi_{ij}^+, \quad \xi_{ij}^+ \geq 0\)

(6) (7)

Both Problems (6) and (7) are standard quadratic programming problems. If \(\ell\) is much larger than the number of bases in \(\Phi\) and \(\tilde{\Phi}\), we can solve it from its dual [8] and largely reduce the number of variables that are needed to be optimized.

Algorithm 1 describes the details of the optimization process.

**Imbalanced Datasets:** In the datasets that the number of positive labels differ significantly from the number of negative labels, large margin classifiers such as LATTL tend to classify all the samples into one class and increase the classification margin. To address this issue, we balance the source dataset by picking the equal number of positive and negative data points and then train TSVM on it.

**C. Theoretical Analysis**

In order to obtain a deeper insight to the source of different types of losses in our algorithm, we present a
Algorithm 1
Input:
1. Source domain examples: \( \{\mathbf{z}_1, \cdots, \mathbf{z}_m\} \)
2. Target domain examples: \( \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \)
3. Parameters \( C_1, C_2, C_3, C_4, \beta \) and the dimension of the hidden space \( r \).
4. Set \( \epsilon = 0.01 \)
Output: The labels for the target domain samples.
5. Randomly initialize \( \{e_i\} \) and \( \{a_i\} \).
6. repeat
7. \( \{a_i\}_{old} = \{a_i\} \)
8. Solve problem 3 and obtain \( \Phi \).
9. Solve problem 4 and obtain \( \Phi \).
10. for \( i \leftarrow 1 \) to \( \ell \) do
11. Solve Problem 5 and obtain \( w_i \) and \( h_i \) and \( y_i \).
12. for \( i \leftarrow 1 \) to \( n \) do
13. Solve Problem 6 and obtain \( \{a_i\} \).
14. for \( i \leftarrow 1 \) to \( n \) do
15. Solve Problem 7 and obtain \( \{e_i\} \).
16. until \( \|\{a_i\} - \{a_i\}_{old}\|_2 < \epsilon \)
17. return \( y_{ij} = 1 \) if \( w_i^T e_i + b_i \geq 0 \) and \( y_{ij} = 0 \) otherwise.

Theoretical analysis on the generalization error of our algorithm using the transductive Rademacher complexity bounds [3]. The transductive Rademacher complexity is the generalization of the inductive Rademacher complexity to the transductive learning settings and similar to the inductive one measures the expected correlation of the patterns created by the algorithm and the pure noise. In other words, the Rademacher complexity is a measure of how likely an algorithm can detect a pattern in pure noise. Thus, even the numeric value of the Rademacher complexity gives meaningful insight to the performance of an algorithm [30].

Let us consider the classification task for one of the classes. Denote \( \tilde{f} = (f_1, \cdots, f_{(n+m)}) \in \mathbb{R}^{m+n} \) the soft-labels (before sign function) of all points in the set \( A_{m+n} \triangleq \{\mathbf{z}_1, \cdots, \mathbf{z}_m, \mathbf{x}_1, \cdots, \mathbf{x}_n\} \). We can define \( F \subseteq \mathbb{R}^{m+n} \) as the set of all possible soft classification points that are generated by our learning algorithm. The goal of the learning algorithm is to minimize the test error \( \mathcal{L}_n(\tilde{f}) \triangleq \frac{1}{n} \sum_{i=1}^{m+n} L(f(j), y_i) \) where \( L(f, y) \) is the 0/1 loss function. The function \( \mathcal{L}_n(\tilde{f}) \) is the empirical error on the training set.

1) Preliminary: According to [18], we have the following definition of transductive Rademacher complexity and the next theorem showing its application in bounding the expected loss of transductive SVM.

**Definition Transductive Rademacher Complexity** Let \( F \subseteq \mathbb{R}^{m+n} \) and \( p \in [0, 1/2] \). Let \( \sigma = (\sigma_1, \cdots, \sigma_{m+n}) \) be a vector of iid random variables such that

\[
\sigma_i \triangleq \begin{cases} 
1, & \text{with probability } p; \\
-1, & \text{with probability } p; \\
0, & \text{with probability } 1 - 2p.
\end{cases}
\]

The (empirical) transductive Rademacher complexity with parameter \( p \) is

\[
R_{m+n}(F, p) \triangleq \left( \frac{1}{m} + \frac{1}{n} \right) \mathbb{E} \left[ \sup_{\sigma, v} \langle \sigma, v \rangle \right]
\]

**Theorem 3.1: Risk bound for Transductive Learning** Let \( F \) be the set of full-sample soft labellings of the algorithm, generated by operating it on all possible training and test set partitions and \( F \in [1, 1]^{m+n} \). Let \( p_0 = \frac{mn}{(m+n)} \sigma \), \( c_0 = \frac{\sqrt{2\ln(2\epsilon)}}{\sqrt{m+n}} \), \( Q \triangleq \left( \frac{1}{2} + \frac{1}{4} \right) \) and \( S \triangleq \frac{\sqrt{Q}}{m+n} \). With probability at least \( 1 - \delta \) over the choice of the training set from \( A_{m+n} \), for all \( f \in F \),

\[
\mathcal{L}_n(\tilde{f}) \leq \mathcal{L}(\tilde{f}) + R_{m+n}(F, p_0) + c_0 Q \sqrt{\min(m, n)} + 2 \sqrt{\frac{SQ}{2} \ln 1/\delta}.
\]
\[ \mathbb{E}_{x,u_i}[H(-u_i g_i(x))] = \sum_{u_i \in \{-1,1\}} \int_{\mathbb{R}^d} H(-u_i g_i(x)) \phi_{x,u_i}(x,u_i) dx \]
\[ = \sum_{u_i \in \{-1,1\}} \int_{\mathbb{R}^r} H(-u_i f_i(a)) \phi_{a,u_i}(a,u_i) da \]
\[ \leq \sum_{u_i \in \{-1,1\}} \int_{\mathbb{R}^r \setminus A} H(-u_i f_i(a)) \phi_{a,u_i}(a,u_i) da \]
\[ + \sum_{u_i \in \{-1,1\}} \int_{\mathbb{R}^r \setminus A} 1_{\phi_{a,u_i}(a,u_i)} da \]
\[ \leq \sum_{u_i \in \{-1,1\}} \int_{\mathbb{R}^r} H(-u_i f_i(a)) \phi_{a,u_i}(a,u_i) da \]
\[ + \frac{1}{2}\mathbb{E}[|\text{sgn}(f_i(a)) - \text{sgn}(f_i(a))|] \]
\[ = \mathbb{E}_{a,u_i}[H(-u_i f_i(a))] + \rho^i \quad (13) \]

Equation (11) is the result of change of variable from \( x \) to \( a \) and application of the transformation of random variables rule (the mapping function \( x \to a \) is assumed to be differentiable.). Equation (12) holds because the integral is absolutely integrable in \( \mathbb{R}^r \) and the set \( A \) is measurable.

The second term is due to an assumption that states \( f_i \) is a reasonably good classifier and if an error happens during the latent representation learning, with a high probability the whole classification task fails.Bounding \( \mathbb{E}[H(-u_i f_i(a))] \) with the empirical transductive Rademacher complexity of the class of functions and noting the fact that in the optimization problem 2 we can bound the empirical loss as \( \mathcal{L}(\tilde{f}_i) \leq \frac{1}{m} \sum_{j=1}^{m} \xi_{ij} \) concludes the proof.

Now we can customize Theorem (3.2) for our LATTL algorithm as following:

**Theorem 3.3:** Suppose the following conditions are hold for problem (2):

1) The source and target domains have the same vocabulary and share the same latent space. The classifier operates in the latent space and is the same for both source and target samples.

2) The loss in the mapping of the target samples to the latent space is bounded. In oeder words,

\[ \| \tilde{a} - a \|_2 \leq d, \quad (14) \]

Then the expected loss in the target to latent space mapping phase can be bounded as the following:

\[ \rho^i \leq \int_{B_i} \phi_a(t) dt \]

Where the set \( B_i \) is defined as \( B_i := \{ a | \| (w_i,a) + b_i \| \leq d \| w_i \|_2 \} \).

**Proof:**

\[ \rho^i = \frac{1}{2} \mathbb{E}[|\text{sgn}(f_i(\tilde{a})) - \text{sgn}(f_i(a))|] \]
\[ = \mathbb{P}\{ (f_i(\tilde{a}) > 0 \cap f_i(a) < 0) \cup (f_i(\tilde{a}) < 0 \cap f_i(a) > 0) \} \]
\[ \leq \mathbb{P}\{ \| w_i \|_2 < \| w_i,a + b_i \| < \} \]
\[ + \mathbb{P}\{ \| w_i,a + b_i \| < d \| w_i \|_2 < \} \]
\[ = \mathbb{P}\{ \| w_i,a + b_i \| < d \| w_i \|_2 \} \]
\[ = \int_{B_i} \phi_a(t) dt \]

In [4], theoretical analysis of general transfer learning algorithm is discussed. However, their analysis assumes that a transfer learning algorithm has advantage over non-transfer learning algorithms by finding a better estimation of target domain distribution using the distribution of the source domain and the mapping between source and target representations. In the performance analysis of LATTL, its two properties prevent us from using the bound developed by Ben-David et al: (i) LATTL does not work by learning the induced distributions, and (ii) In one phase of our algorithm LATTL uses TSVM classification and we need to include the hinge loss of TSVM in the bound. Hence we developed the bound in Theorem (3.2) to obtain more insight to the problem.

**D. Empirical Rademacher Bound**

Following the definition, the Rademacher complexity of the class of functions \( \mathcal{F}_j \) can be written as:

\[ R_{m+n}(\mathcal{F}_j, \rho_0) \triangleq \left( \frac{1}{m} + \frac{1}{n} \right) \mathbb{E}_\sigma \left[ \sup_{f \in \mathcal{F}_j} \sum_{i=1}^{m+n} \sigma_i f(a_i) \right] \quad (15) \]

\[ = \mathbb{E}_\sigma \left[ \sup_{\| w_i - \Phi_a \|_2, \| a_i \|_2 \leq \eta_i} \left( \frac{1}{m} + \frac{1}{n} \right) \sum_{i=1}^{m+n} \sigma_i (\langle w_j, a_i \rangle + b_j) \right] \]

A major difficulty in computing the value in Equation (16) is that we cannot compute the expectation with respect to the random vector \( \sigma \). Here we use the same trick as in [19] to replace the expectation with averaging. Note that due to McDiarmid theorem [30], the expression inside square brackets is concentrated around its expected value when the optimization problem (2) has a feasible solution. This is because, if the optimization problem (2) has a feasible solution we have \( \eta_j, \zeta_j < \infty \) and there are \( C^* \) and \( b^* \) such that \( \| w_j \|_2 \leq C^* < \infty \) and \( |b_j| \leq b^* < \infty \). Hence the change in the function \( h(\sigma) \triangleq (1/m+1/n) \sum_{i=1}^{m+n} \sigma_i f(a_i) \) when we flip the value of the one of \( \sigma_i \) can be bounded as following:

\[ c_j \leq \left( \frac{2}{m} + \frac{2}{n} \right) \sup_{\| w_j, a_i \| + b_j} \{ |\langle w_j, a_i \rangle + b_j| \} \]
\[ \leq \left( \frac{2}{m} + \frac{2}{n} \right) \left( C^* \frac{\max(\eta_j, \zeta_j)}{\beta} + b^* \right) \]
When \( c_j \) is bounded, the McDiarmid theorem guarantees that as \((1/m+1/n) \to 0\), with high probability the average value of \( h(\sigma) \) for different instantiations of \( \sigma \) should be a good estimate for the expected value of it. Thus we instantiate some sample vectors of \( \sigma \) and call it \( \hat{\sigma} \) and use it to compute the value in Equation (16). The final optimization problem that should be solved to obtain the bound on Rademacher complexity of class functions \( \mathcal{F}_j \) is shown in Equation (17).

\[
\max_{w_j, b_j, (a_i)} \left( \frac{1}{m} + \frac{1}{n} \right) \sum_{i=1}^{m+n} \delta_i (w_j, a_i) + b_j \\
\text{subject to:} \\
\forall i \in \{1, \ldots, m\} \quad \|w_j\|_2^2 + b_j^2 \leq B \\
\forall i \in \{(m+1), \ldots, n\} \quad \|z_i - \Phi a_i\|_2^2 + \beta \|a_i\|_2^2 \leq \eta_i \\
\forall i \in \{(m+1), \ldots, n\} \quad \|x_i - \Phi a_i\|_2^2 + \beta \|a_i\|_2^2 \leq \zeta_i
\]

(17)

The values of \( \eta_i, \zeta_i, \Phi \) and \( \hat{\Phi} \) are obtained from the solution of the optimization problem (2). Note that the optimization problem in Equation (17) is convex in \( w_j \) and \( b_j \) (while fixing \( \{a_i\} \)) and convex in \( \{a_i\} \) (while fixing \( w_j \) and \( b_j \)) and can be solved by iteration.

IV. CONNDTIONS TO EXISTING WORK

In the previous section, we introduce a general framework of transductive transfer learning that learns the latent feature mapping function and classification function in one unified model. As seen in Equation (1), the formulation consists of feature reconstruction loss and classification loss. Depending on the choice of loss function, a variety of transfer learning algorithms can be developed.

Table I exemplifies one interesting view of existing transfer learning models through this decomposition. For example, comparing STL with our general framework in Equation (1), we can see that self-taught learning only concerns about the reconstruction loss. Since there are no terms associated with the examples in target domain and classification loss, it is difficult to provide the theoretical justification on the performance of self-taught learning. Another example is EASYADAPT algorithm [15], which utilizes a simple approach, i.e., feature augmentation, as feature reconstruction. As we can see, this approach has zero reconstruction loss. However, since it increases the feature dimension, it is also exposed to the risk of over-fitting (especially for high-dimensional data).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>S-Domain</th>
<th>T-Domain</th>
<th>Mapping function</th>
<th>Reconstruction Loss (Domain)</th>
<th>Classification Loss (Domain)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-taught Learning</td>
<td>U</td>
<td>L</td>
<td>Shared</td>
<td>Squared loss with L-1 regularizer (S)</td>
<td>Hinge or log loss (S)</td>
</tr>
<tr>
<td>EASYADAPT</td>
<td>L</td>
<td>L</td>
<td>Shared</td>
<td>Squared loss with L-2 regularizer (S+T)</td>
<td>Hinge Loss (S+T)</td>
</tr>
<tr>
<td>LATTL</td>
<td>L</td>
<td>U</td>
<td>Different</td>
<td>Squared loss with L-2 regularizer (S+T)</td>
<td>Hinge Loss (S+T)</td>
</tr>
<tr>
<td>LMPROJ</td>
<td>L</td>
<td>U</td>
<td>Shared</td>
<td>Projected mean distance (S + T)</td>
<td>Hinge loss (S)</td>
</tr>
</tbody>
</table>

Table I
GENERAL FRAMEWORK OF TRANSFER LEARNING AND ITS SPECIFIC EXAMPLES: S: SOURCE (DOMAIN); T: TARGET (DOMAIN); U: UNLABELED EXAMPLES; L: LABELED EXAMPLES.

Large-margin transductive transfer learning (LMPROJ) [28] is most related to our model, but with two major difference: (1) it uses the same mapping function for the source and target domains; (2) it utilizes projected mean discrepancy distance instead of squared loss as the reconstruction loss. The basic assumption is that in latent space, the centroid of examples in the source domain should be as close to the one in the target domain as possible. This assumption is reasonable for certain data distributions, but not true for others. For example, LMPROJ does not work for cases in which the positive examples in the source domain and the target domain could be samples from mixtures of Gaussians with different centroids in the latent space. In addition, the complexity of the resulting optimization problem is significantly increased by modeling the mean discrepancy distance. Transductive SVM [32] and [21] is not specifically designed for transductive transfer learning, but could be applicable. However, it is clear that it does not have the reconstruction loss to take into consideration the different distributions of the data in the source domain and target domain.

V. EXPERIMENTS

In order to demonstrate the effectiveness of our algorithm, we compare it with other baseline algorithms, including LMPROJ, STL, TSV, and EASYADAPT, on one synthetic dataset and three application datasets, including Reuters-21578, 20News and the Landmine datasets.

**Synthetic Dataset:** A synthetic dataset is generated to examine the effectiveness of our proposed algorithm for transductive transfer learning problems. We choose a 3-class classification problem. In the source domain, each example is generated from a multivariate Gaussian mixture model of 9-dimensional features. In the target domain, the same method is used to generate the labels as well as the feature vectors. But, different from the source domain, for each generated example, approximately 70% percent of its features would be set to zero randomly to mimic the feature sparsity in many applications. In the end, the dataset consists of 2000 examples in source domain, and 473 examples in the target domain.

**Reuters-21578:** This dataset is one of the benchmark
datasets for text classification\(^1\). It consists of 21578 documents with each document labeled as one or more of 135 categories. To construct a transfer learning problem, we divide the dataset into two sub-datasets. The source domain contains the body text of each document in one subset, while the target domain contains only the title in the other subset. We further process the texts by removing stop words, stemming, and removing features appearing less than 20 times. The final dataset consists of 10376 examples in the source domain and 10305 in the target domain with 1029 features in total.

**20Newsgroup** This is another benchmark dataset for text classification. It contains four main categories, i.e., ‘comp’, ‘rec’, ‘sci’, ‘talk’, each of which also has subcategories, such as graphics, hardware, autos, etc. Based on this structure, we generate 2 transfer learning tasks. Each task involves distinguishing two higher-level categories. The source domain and the target domain contains examples from different lower-level categories. For example, one transfer learning task is to distinguish between ‘rec’ and ‘talk’. The source domain contains examples from rec.sport.baseball and talk.politics.misc; whereas the target domain contains examples from rec.sport.hockey and talk.religion.misc. The way that the transfer learning tasks are generated is similar to [20]. After stemming and stop word removal, the final dataset 3854 documents in the source domain and 3747 in the target domain with 6600 features.

**Landmine Dataset** The remote sensing problem is based on data collected from real landmines\(^2\). In this problem, there are a total of 29 sets of data, collected from different landmine fields. Each data is represented as a 9-dimensional feature vector extracted from radar images, and the class label is true mine or false mine. Since each of the 29 datasets are collected from different regions that may have different types of ground surface conditions, these datasets are considered to be dominated by different distributions. As described in [34], datasets 1 to 10 are collected regions with foliated rocks while datasets 20 to 24 are collected from regions that are bare earth or desert. Thus, we combine datasets 1 to 5 as the source domain dataset, datasets 6 to 10 as the target domain dataset with high similarity to the source domain and datasets 20 to 24 as another target domain dataset with low similarity.

### A. Accuracy Comparison

In order to compare the performance of different methods, we use the average Geometric Mean (G-mean) value which is well-suited for performance evaluation of algorithms in the imbalanced datasets. G-mean is defined by $\text{G-mean} = \sqrt{PR}$, where $P$ represents precision which is defined as the ratio of true positives in the positive predictions and $R$ represents recall, defined as the ratio of positive predictions among the true positive labels. In datasets with multiple labels per sample, we report the average G-mean across the multiple labels.

Our algorithm has six hyperparameters ($C_1 - C_4, \beta, r$) to be tuned during the performance evaluation which can be impractical for some large datasets. Hence, we propose a heuristic for faster tuning of the parameters. We first find the optimal values for $C_2$ and $C_4$ by tuning TSVM on the given dataset. In the next step, we fix the value of $C_2, C_4$ and find the optimal value of the remaining four hyperparameters. To present a more practical performance of LATTL, in all performance report tables, we report the accuracies achieved by both the fully tuned LATTL and the one tuned via TSVM parameter tuning heuristic, denoted by “LATTL (Full)” by “LATTL (TSVM)”, respectively. We use 5-fold cross-validation to tune the parameters.

In the Reuters-21578 and 20Newsgroup datasets, we use only 20\% of the unlabeled samples in Equation (5) to accelerate the TSVM operation embedded in LATTL. To have a more fair comparison, we use linear classifiers for all of the experiments.

Tables II, III, IV, V and VI compare the performance of our algorithm with the other state of the art algorithms. It is evident that our algorithm outperforms others in all datasets in the G-mean measure, and stays competitive in precision and recall. More specifically, we make the following observations:

1. Interestingly, even though SVM is not developed for transductive transfer learning, it keeps pace with LMPROJ...
and STL in performance comparison.
(2) Recall that LMPROJ finds the mapping function by minimizing the differences between the source and target domain in the latent space, and meanwhile maximizing the performance of the classifier. The algorithm works better than SVM only on the synthetic dataset, which has a similar underlying generation process as its assumption.
(3) STL attempts to learn a latent space from the samples in the source domain via sparse coding. It is evident that STL does not perform well in our settings. We believe this is due to the following reasons: STL assumes that the source samples are not labeled and therefore the learned latent space may not be as effective as the ones using labels as guidance; instead, our algorithm uses the labels in the source domain for two purposes: (i) guiding the search of latent space for the source domain. (ii) learning a more efficient latent space for the classification task for both the source domain and target domain.
(4) EASYADAPT is not directly applicable in our settings since there are no labeled examples from the target domain. Therefore we design a transductive version of EASYADAPT that utilizes the unlabeled examples in the target domain via transductive learning algorithms. In our experiment, we use EASYADAPT with TSVM, which is referred to as T-EASYADAPT. We can see that its performance is inferior to our algorithm.
(5) Our LATTL algorithm performs consistently the best among all datasets. In addition, we can notice that its improvement is much more significant (10-20%) on datasets with less similarity between the source domain and the target domain, such as synthetic dataset and Landmine 20-24.
(6) Performance of LATTL is closely related to the performance of TSVM. The improvement over the performance of TSVM stems from the fact that the source and target dataset are more similar to each other when represented in the latent space.

B. Parameter Impact Study

In this section, we design several experiments to study the impact of the parameters on the performance of LATTL. Due to the lack of space, we present only the results for a subset of the experiments carried out on the synthetic dataset. The impacts of the the parameters are as follows.

$\beta$: Figure 1(a) shows the performance of LATTL as $\beta$ is varied, while other parameters are fixed. The general trend suggests that smaller values of $\beta$ should be favorable; however a more meaningful conclusion is that the performance is almost insensitive to the value of $\beta$. In fact, in the situations full parameter tuning is expensive and some parameters should be sacrificed, $\beta$ should be the first choice and any value of $\beta \in [10^{-2}, 10^{0}]$ should be a reasonable choice.

$C_1 - C_4$: Unlike $\beta$, Figures 1(b) and 1(c) show that the performance of LATTL critically depends on the choice of $C_1 - C_4$ parameters. We run two more experiments: (1) All parameters except $C_1$ and $C_3$ are fixed and these two parameters vary, and (2) All parameters except $C_2$ and $C_4$ are fixed and these two parameters vary. Figure 1(c) shows that LATTL achieves higher performance when the hinge losses ($\xi_{ij}$) are more stressed than the transductive losses.
(ζn i ). This suggests that the main task of classification is still done using labeling information in the source domain. In contrast, Figure 1(b) implies that for reconstruction losses, selection of penalization coefficient depends on the number of samples in each domain. Since we have 2000 samples in the source domain and 473 one in the target domain, we realize that the optimal performance should be achieved for values of C1 and C3 satisfying 2000C1 ≈ 473C3.

![Figure 2. (a) The transductive Rademacher Complexity of our algorithm with different values of the latent space dimension on the Synthetic dataset. The red horizontal line shows the value of Empirical Rademacher complexity of SVM. (b) The accuracy achieved by LATTL with different latent space dimension.](image1)

![Figure 3. (a) The transductive Rademacher Complexity of our algorithm with different values of the latent space dimension on the Landmine 6-10 dataset. The red horizontal line shows the value of Empirical Rademacher complexity of SVM. (b) The accuracy achieved by LATTL with different latent space dimension.](image2)

### Table VI

<table>
<thead>
<tr>
<th></th>
<th>G-mean</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LATTL (Full)</td>
<td>0.6281</td>
<td>0.4800</td>
<td>0.8219</td>
</tr>
<tr>
<td>LATTL (TSVM)</td>
<td>0.6239</td>
<td>0.4763</td>
<td>0.8173</td>
</tr>
<tr>
<td>TSVM</td>
<td>0.6072</td>
<td>0.4615</td>
<td>0.7990</td>
</tr>
<tr>
<td>SVM</td>
<td>0.3759</td>
<td>0.5227</td>
<td>0.1483</td>
</tr>
<tr>
<td>LMPROJ</td>
<td>0.3646</td>
<td>0.5408</td>
<td>0.1413</td>
</tr>
<tr>
<td>STL</td>
<td>0.5004</td>
<td>0.4410</td>
<td>0.5679</td>
</tr>
<tr>
<td>T-EASYADAPT</td>
<td>0.3263</td>
<td>0.5000</td>
<td>0.2130</td>
</tr>
</tbody>
</table>

**Performance Comparison in the Landmine Tasks 20-24 Combined Dataset.**

### C. Dimension of the Latent space and the Empirical Rademacher Complexity

In order to study the effect of latent space dimension, we use our risk bound analysis to provide more insight. Since our analysis bounds the generalized error rate, in this section we change our accuracy measure to average correct labeling \((1 - \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|)\). We compute the summation of the empirical Rademacher complexity for all labels for 200 realizations of the Rademacher variables. Figures 2 and 3 show the mean Rademacher complexity values with different latent space dimensionality \(r\) on the synthetic dataset and Landmine 6-10 dataset, respectively. As expected, the Rademacher complexity of LATTL is upper bounded by SVM, suggesting the superior classification performance of LATTL. The Rademacher complexity plot shows that as we map the data points to a higher dimensional latent space, the classification task become easier. However, considering the accuracy plot, we can deduce that, as expected, the reconstruction loss should increase as we increase the dimensions of the latent space. In the Landmine 6-10 dataset these two competing factors almost neutralize each other, but in the synthetic dataset \(r = 3\) is the optimal latent space dimension.

### VI. Conclusion

In this paper, we propose a general framework for transductive transfer learning. The basic idea is to seek a latent space in which the data from different domains share the same classification function. We learn the latent space mapping functions and the classification function by simultaneously minimizing the feature reconstruction loss and classification loss. We demonstrate the idea by developing one specific example, namely LATTL algorithm. Different from existing transfer learning algorithms, it allows different mapping functions for different domains, and seeks them by simultaneously minimizing the squared loss with L-2 regularizer (as reconstruction loss) and transductive hinge loss (as the classification loss). We also provide theoretical analysis on the classification loss of the LATTL algorithm via Rademacher complexity as well as a solution to estimate its empirical value. Experiment results on one synthetic dataset and three application datasets demonstrate the effectiveness of our algorithm.

For future work, we are interested in investigating the theoretical bound of the general framework we proposed in the paper and efficient algorithms to solve the associated optimization problems. Acceleration of the speed of the algorithm using parallel implementation and randomized updates of Equations (6) and (7) is another direction for future work.

**ACKNOWLEDGEMENT**

The work in this paper is sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) under the
Anomaly Detection at Multiple Scales (ADAMS) program, Agreement Number W911NF-11-C-0200. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Defense Advanced Research Projects Agency or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES


[22] Neil D. Lawrence and John C. Platt. Learning to learn with the informative vector machine. In ICML, 2004


[34] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. JMLR, MAY 2007.