Understanding How Feature Structure Transfers in Transfer Learning

Tongliang Liu[†], Qiang Yang[‡], Dacheng Tao[†]
[†]UBTech Sydney AI Institute and SIT, FEIT, The University of Sydney, Australia
[‡]Hong Kong University of Science and Technology, Hong Kong
{tongliang.liu, dacheng.tao}@sydney.edu.au, qyang@cse.ust.hk

Abstract

Transfer learning transfers knowledge across domains to improve the learning performance. Since feature structures generally represent the common knowledge across different domains, they can be transferred successfully even though the labeling functions across domains differ arbitrarily. However, theoretical justification for this success has remained elusive. In this paper, motivated by self-taught learning, we regard a set of bases as a feature structure of a domain if the bases can (approximately) reconstruct any observation in this domain. We propose a general analysis scheme to theoretically justify that if the source and target domains share similar feature structures, the source domain feature structure is transferable to the target domain, regardless of the change of the labeling functions across domains. The transferred structure is interpreted to function as a regularization matrix which benefits the learning process of the target domain task. We prove that such transfer enables the corresponding learning algorithms to be uniformly stable. Specifically, we illustrate the existence of feature structure transfer in two well-known transfer learning settings: domain adaptation and learning to learn.

1 Introduction

Transfer learning is motivated by human learning. When humans encounter a continual stream of learning tasks, we do not just learn concepts, but also biases, and we are capable to transfer them to new tasks. Transfer learning is therefore referred to as extracting knowledge from source domains and applying it to improve the learning performance in a target domain.

The last decade has witnessed the success of various transfer learning algorithms [Shao *et al.*, 2014; Long *et al.*, 2015; Li *et al.*, 2015; Liu *et al.*, 2016; Shao *et al.*, 2016] in exploiting the transfer of knowledge across domains. Self-taught learning [Raina *et al.*, 2007], domain adaptation [Ben-David *et al.*, 2007] (sometimes called transfer learning [Maurer *et al.*, 2013] or lifelong learning [Pentina and Lampert, 2014]), and learning to learn [Baxter, 2000] (sometimes called multi-task learning [Ando and Zhang, 2005]) have achieved great successes in a variety of tasks [Si *et al.*, 2010; Luo *et al.*, 2014]. For additional references, the interested reader is referred to the survey [Pan and Yang, 2010].

Domain adaptation solves the problem in which the joint distributions P_{xy} over the source and target domains are different but related, where x denotes the feature and y the label. Most algorithms for domain adaptation try to (appropriately) correct the source domain distribution so that its knowledge can be transferred to the target domain. Covariate shift (P_x differs across domains) [Huang *et al.*, 2006], model shift ($P_{y|x}$ differs across domains) [Wang and Schneider, 2014], target shift (P_y differs across domains) [Zhang *et al.*, 2013] and conditional shift ($P_{x|y}$ differs across domains) [Zhang *et al.*, 2013] are the representative and notable models for domain adaptation. All these models will succeed only when the conditional distributions $P_{y|x}$ (or the labeling functions) across domains are (or can be corrected to be) identical (or at least nearly identical). Most of the existing theoretical analyses [Ben-David *et al.*, 2007; Blitzer *et al.*, 2008; Mansour *et al.*, 2009; Ben-David *et al.*, 2010; Zhang *et al.*, 2015; Gong *et al.*, 2016] for domain adaptation are based upon this condition, and cannot explain the success of the applications where the conditional distributions $P_{y|x}$ differ significantly (or even arbitrarily) across the source and target domains.

Learning to learn was introduced by [Baxter, 2000] who defined the task environment as a probability measure on a set of related tasks. Most algorithms commonly assume that parameters for modeling different tasks are partially shared. Most of the existing theoretical analyses [Baxter, 2000; Maurer, 2009; Kuzborskij and Orabona, 2013; Maurer *et al.*, 2013] for learning to learn focus on this assumption. For example, [Maurer *et al.*, 2013] assumed that the task parameters are well approximated by sparse linear combinations of the atoms in a dictionary and provided a theoretical justification for the success. The corresponding theoretical results are therefore based upon the condition that the conditional distributions $P_{y|x}$ across domains are related and limited in the environment and cannot differ arbitrarily.

Can transfer learning work when the conditional distributions $P_{y|x}$ differ arbitrarily across domains? The answer is positive. One well-known successful example is self-taught learning [Raina *et al.*, 2007]. Given a large amount of unlabeled data (e.g., randomly collected from the Internet), self-

taught learning extracts domain invariant visual structure (or feature structure defined in this paper) to significantly improve the classification accuracy in the task domain. It does not require the source data to follow the same class labels or generative distribution as the target domain data, so its conditional distributions $P_{y|x}$ of source and target domains can differ very much, or even arbitrarily. Feature structures generally represent common knowledge across different domains, so they are transferrable regardless of the change of the conditional distributions $P_{y|x}$ across domains.

In this paper, motivated by self-taught learning, we define the feature structure of a domain as a set of bases which can (approximately) reconstruct any observation in the domain, and say an algorithm is a feature structure transfer learning algorithm if it transfers feature structure from the source domain to the target domain. For example, a self-taught learning algorithm is a feature structure transfer learning algorithm, in which the transferred feature structure is a set of sparse coding bases. However, to the best of our knowledge, theoretical justification for the success of feature structure transfer learning remains elusive.

Motivated by that self-taught learning can be formulated as a Tikhonov regularized learning in the target domain, where the Tikhonov matrix contains the feature structure extracted from the source domains, we first provide a theoretical justification for feature structure transfer learning from the perspective of regularization. We show that if the feature structures of the source and target domains are similar (i.e., for any target domain observation x^t, it holds that $x^t \approx B\alpha$, where B denotes the extracted source domain feature structure and α a reconstruction coefficient), the source domain data will provide a regularization matrix and enable feature structure transfer learning algorithms to be uniformly stable [Bousquet and Elisseeff, 2002] and to learn desirable accurate predictors from small fractions of labeled examples, which provides insight as to why those algorithms achieve their empirical successes.

In contrast to the existing theoretical analyses, we focus on justifying how feature structure can be transferred to improve classification performance from the perspective of regularization. Specifically, our analyses show that if the source domain observations can be a feature structure for the target domain and the employed loss function is strongly convex, the source domain observations can be interpreted to function as a regularization matrix, which benefits the learning process of the target domain task. We prove that existing learning algorithms, such as domain adaptation and learning to learn, for learning the target predictors are feature structure transfer learning algorithms and therefore are uniformly stable, which are indispensable complements for the existing theoretical analyses.

Organization of the paper. We describe the learning setup in Section 2. In Section 3, we introduce our motivation that self-taught learning transfers feature structure and is uniformly stable. We provide theoretical justifications for the transfer of feature structure in the algorithms of domain adaptation and learning to learn in Sections 4 and 5, respectively. Finally, we draw conclusions and discuss future work in Section 6. We present the proof of our assertion in Section A.

2 Preliminaries

We start by introducing the notation that will be used throughout this paper. Let $\mathcal{H}$ denote the separable real Hilbert space with the inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$. Let $z = (x, y) \in \mathcal{H} \times \{-1, 1\}$ be a training example and $S = \{z_1, \ldots, z_n\}$ an i.i.d. training sample. We denote the i.i.d. training samples of the source and target domains by $S_s = \{z_1^s, \ldots, z_{n_s}^s\} = \{(x_1^s, y_1^s), \ldots, (x_{n_s}^s, y_{n_s}^s)\}$ and $S_t = \{z_1^t, \ldots, z_{n_t}^t\} = \{(x_1^t, y_1^t), \ldots, (x_{n_t}^t, y_{n_t}^t)\}$, respectively. Let H be a linear hypothesis class and $\ell(y, h(x))$ a loss function measuring the risk that is incurred by predicting $h(x)$ when the true label is y. We denote the expected and empirical risks by $R(h) = \mathbb{E}\ell(y, h(x))$ and $R_n(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i))$, respectively. A learning algorithm discovers a mapping that maps a training sample S to a hypothesis function $h_S \in H$. The defect $R(h_S) - R_n(h_S)$ is called the generalization error, which is usually exploited to measure the "goodness" of learning algorithms. A small generalization error implies that the algorithm could generalize well on unseen data.

Intuitively, given a larger training sample, a learning algorithm will learn a more accurate predictor. However, in practice, for a specific target domain, the training examples may be limited. Fortunately, we often have a large number of additional training examples, drawn from closely related source domains, which contain information that can improve the learning performance of the target domain task. Transfer learning is therefore defined as learning algorithms that employ the additional source domain examples to improve the learning performance in the target domain.

We define a set of bases B as a feature structure of a domain $\mathcal{X} \times \{-1, 1\}$ if for any observation $x \in \mathcal{X}$, $x \approx B\alpha$, where α is the reconstruction coefficient. In this paper, we consider the transfer learning algorithms where feature structures are transferable across domains to improve the learning performance, which we call feature structure transfer learning.

To theoretically justify feature structure transfer learning, we introduce the stability framework proposed by [Bousquet and Elisseeff, 2002].

Definition 1 (Uniform stability) *An algorithm has uniform stability β with respect to the loss function ℓ and a specific domain $\mathcal{Z} \subset \mathcal{H} \times \{-1, 1\}$ if for any independently distributed training sample $S \in \mathcal{Z}^n$, any $i \in \{1, ..., n\}$ and any $z \in \mathcal{Z}$, the following holds:*

$$|\ell(y, h_S(x)) - \ell(y, h_{S^i}(x))| \leq \beta, \tag{1}$$

where S^i denotes the training sample S with the i-th example z_i being replaced by an i.i.d. example z_i'.

[Bousquet and Elisseeff, 2002] also proved the following generalization error bound.

Theorem 1 *Let the learning algorithm be β-stable and let S be a training sample with n independent random examples. Assume that the loss function ℓ is bounded by M. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds:*

$$R(h_S) \leq R_n(h_S) + \beta + (2n\beta + M)\sqrt{\frac{\log 1/\delta}{2n}}. \tag{2}$$

In this paper, we will illustrate that the transfer of feature structure enables transfer learning algorithms to be uniformly stable, as a result of which the generalization errors will be small.

3 Self-taught Learning and Motivation

In this section, we show that feature structure will have a Tikhonov regularization property in self-taught learning. Motivated by this, we will show in the following sections that feature structure transfer exists widely in existing transfer learning algorithms.

Self-taught learning introduced by [Raina *et al.*, 2007] exploits the unlabeled data in the source domain to improve classification performance in the target domain. It first extracts a feature structure by learning a set of sparse coding bases B and then approximately reconstructs each target domain observation as a sparse linear combination of the bases B, i.e., $x_i^t = B\alpha_i + \eta_i, \|\alpha\|_1 \leq r_1, i = 1, \ldots, n_t$, where η_i is the corresponding residual satisfying that $\|\eta_i\|$ is small (because B can be over-completed), α_i encodes the reconstruction coefficient for this target domain observation, and r_1 is a constant. Lastly, self-taught learning learns a predictor by minimizing the following regularized objective function:

$$\min_v \frac{1}{n_t} \sum_{i=1}^{n_t} \ell\left(y_i^t, \langle v, \alpha_i \rangle\right) + \lambda \|v\|^2, \tag{3}$$

where λ is a regularization parameter.

We explicitly emphasize the function of the coding bases B and reformulate (3) as

$$\min_w \frac{1}{n_t} \sum_{i=1}^{n_t} \ell\left(y_i^t, \langle w, B\alpha_i \rangle\right) + \lambda \|B^T w\|^2. \tag{4}$$

Optimization problems (3) and (4) are equivalent if we write $v = B^T w$.

If the residuals $\eta_i, i = 1, \ldots, n_t$ are small, $B\alpha_i$ will be approximately equal to the target domain observations x_i^t. Thus, self-taught learning can be viewed as learning algorithms that minimize the following regularized objective function:

$$\min_w \frac{1}{n_t} \sum_{i=1}^{n_t} \ell\left(y_i^t, \langle w, x_i^t \rangle\right) + \lambda \|B^T w\|^2. \tag{5}$$

By regularizing the parameter with respect to the feature-structure-based regularization matrix B, self-taught learning algorithms that solve the optimization problems (3) and (4) are uniformly stable.

As discussed in [Bousquet and Elisseeff, 2002], to show regularized learning algorithm is uniformly stable, we need the employed loss function to be admissible (a Lipschitz-like condition, which has been widely used [Mohri *et al.*, 2012]).

Definition 2 (σ-admissible) *A loss function ℓ is σ-admissible with respect to the hypothesis class H if there exists a $\sigma > 0$ such that for any two hypotheses $h, h' \in H$ and any example $z \in \mathcal{H} \times \{-1, 1\}$, the following inequality holds:*

$$|\ell(y, h(x)) - \ell(y, h'(x))| \leq \sigma |h(x) - h'(x)|. \tag{6}$$

Theorem 2 *Let the loss function ℓ be convex and σ-admissible. Algorithms that minimize the objective functions in (3) and (4) are uniformly stable with respect to the target domain. For any (x^t, y^t), $x^t = B\alpha + \eta_1$, in the target domain, the following inequality holds for problem (4):*

$$\left|\ell(y^t, \langle w_{S_t}, B\alpha \rangle) - \ell\left(y^t, \left\langle w_{S_t^i}, B\alpha \right\rangle\right)\right| \leq \frac{\sigma^2 r_1^2}{\lambda n_t}. \tag{7}$$

The proof method of Theorem 2 is the same as that in [Bousquet and Elisseeff, 2002].

Problems (4) and (5) are the well-known Tikhonov regularized empirical risk minimization, where $\|B^T w\|^2$ is the Tikhonov regularization and B^T is the Tikhonov matrix (also known as the regularization matrix). The frequently used ℓ_2-regularization $\|w\|^2$ is a special case of Tikhonov regularization by setting $B = I$. For particular problems, it would be more reasonable to find a suitable regularization matrix B than simply setting $B = I$. Self-taught learning provides a way to find a suitable regularization matrix B and its success has been empirically proven.

Our analyses show that self-taught learning extracts a feature structure B form the source domain data. It then transfers the feature structure to improve the target task performance by regularizing the learning parameter using the extracted feature structure as a regularization matrix. Theorem 2 proves that self-taught learning algorithms are uniformly stable. It therefore has a small generalization error according to Theorem 1. Motivated by the feature structure transfer procedure of self-taught learning, we will show in the following sections that a similar feature structure transfer procedure exists widely in the traditional transfer learning algorithms.

We note here that feature structure transfer learning differs very much from the pre-processing technique that represents the target domain observations onto a subspace. For example, the feature structure B could be over-completed. Moreover, we will show in the following sections that feature structures can be transferred in a general form of a set of source domain observations for domain adaptation and learning to learn problems, where we do not need to specifically learn the coding bases B.

4 Domain Adaptation

Often, domain adaptation studies the transfer learning problem where the target domain has a small amount of data, or even no label information, while the source domain has relatively sufficient data. However, in this paper, we consider the situation where the target domain has a number of labeled examples, which often happens in practice.

We study the domain adaption algorithms proposed by [Blitzer *et al.*, 2008] which minimize a convex combination of the source and target empirical risk:

$$\min_w R_{n_s+n_t,\alpha} = \alpha R_{n_t}(w) + (1-\alpha) R_{n_s}(w), \tag{8}$$

where $\alpha \in (0, 1)$. By setting a proper value for α, i.e., $\alpha = n_t/(n_s + n_t)$, the objective function in (8) includes the objective function of the traditional domain adaptation algorithm

$$\min_w \frac{1}{n_s + n_t} \sum_{i=1}^{n_s+n_t} \ell\left(y_i, \langle w, x_i \rangle\right) \tag{9}$$

as a special case.

In the following, we will show that if the source and target domains have similar feature structures, uniform stability can be derived for the learning algorithms that minimize (8) by interpreting the function of the source data as a regularization matrix.

The source domain observations can function as a transferable feature structure for improving the target domain task if the following assumption holds:

Assumption 1 *For any target domain observation x^t, we have $x^t = \sum_{j=1}^{n_s} \gamma_j x_j^s + \eta_2$, where $\|\gamma\| \leq r_2$, $\{x_1^s, \ldots, x_{n_s}^s\}$ are the source domain observations, and η_2 is a small residual that $\|\eta_2\| \leq \varepsilon_2$.*

To interpret the function of source domain observations as a regularization, we need the employed loss function to be strongly convex.

Definition 3 (c-strongly convex) *A differentiable loss function $\ell(y, h(x))$ is called c-strongly convex with respect to $\|\cdot\|$ if the following inequality holds for any $z = (x, y)$ in its domain and any $h, h' \in H$:*

$$(\nabla \ell(y, h(x)) - \nabla \ell(y, h'(x)))^T (h(x) - h'(x)) \geq c\|h(x) - h'(x)\|^2, \quad (10)$$

where $c > 0$ and $\nabla \ell(y, h(x))$ denotes the gradient of the loss function $\ell(y, h(x))$ with respect to the predictor $h(x)$.

Strong convexity and strong smoothness are dual properties. Strongly convex programming algorithms have many benign properties, both in the speed of optimization and the quality of generalization [Kakade and Tewari, 2009].

The square loss function is 2-strongly convex and has been widely used. Many other frequently used loss functions, such as hinge loss and logistic loss, are only convex but not strongly convex. However, they may be strongly convex when $h(x)$ is restricted to a compact set. For example, the logistic loss $\ell(y, h(x)) = \log(1 + \exp(-yh(x)))$ is $\exp(-U)/4$-strongly convex when $h(x)$ is restricted to the interval $[-U, U]$, because $d^2\ell(y, h(x))/d^2h(x) = \exp(yh(x))/(\exp(yh(x)) + 1)^2 \geq \exp(-U)/4$. They can also be revised as strongly convex by adding a strongly convex regularization, e.g., the ℓ_2-regularization.

The goal of domain adaptation is to modify the source domain distribution so that the learned predictor can perform well on the target domain. We now show that the feature structure provided by the source domain data can be transferred to enable domain adaptation algorithms to be uniformly stable with respect to the target domain.

Theorem 3 *Suppose Assumption 1 holds. Let the loss function ℓ be c-strongly convex and σ-admissible. Then, algorithms that solve optimization problem (8) are uniformly stable with respect to the target domain. For any (x^t, y^t) drawn from the target domain, we have:*

$$\left|\ell(y^t, \langle w_{S_t}, x^t\rangle) - \ell\left(y^t, \left\langle w_{S_t^i}, x^t\right\rangle\right)\right| \leq \frac{2\alpha n_s \sigma^2 (r_2 + O(\varepsilon_2))^2}{c(1-\alpha)n_t}. \quad (11)$$

When $\varepsilon_2 = 0$, which means that the target observations can be perfectly reconstructed by the source observations or that the source domain observations can serve as a feature structure for the target domain, we have

$$\left|\ell(y^t, \langle w_{S_t}, x^t\rangle) - \ell\left(y^t, \left\langle w_{S_t^i}, x^t\right\rangle\right)\right| \leq \frac{2\alpha n_s \sigma^2 r_2^2}{c(1-\alpha)n_t}. \quad (12)$$

See the proof in the Appendix.

Inequalities (11) and (12) in Theorem 3 show that the upper bound of the uniform stability will decrease fast with a an order of $O(1/n_t)$, where n_t is the sample size of the target domain. This means that the target learning algorithm can generalize fast on unseen data.

The empirical study in [Blitzer *et al.*, 2008] shows that for some real-world datasets, tuning the parameter α can greatly improve transfer learning performance. They also derived upper bound and discussed that by choosing different values of α, we can effectively trade off the small amount of target data against the large amount of less relevant source data. This is in accordance with our analyses, where α has been interpreted as a regularization parameter. Such a tuning parameter is necessary for the transfer of feature structure by balancing the different feature structure scales across domains.

In the proof of Theorem 3, we have interpreted the source domain observations $X^s = [x_1^s, \ldots, x_{n_s}^s]$ as a transferable feature structure, which function as a regularization matrix for the learning algorithm that learns the target domain task; while in self-taught learning, the coding bases B is a transferable feature structure functioning as a regularization matrix for the learning algorithm that learns the target domain task. Although the source domain data are labeled, we have not exploited the label information during the proof procedure, and thus our theoretical justification for feature structure transfer depends on the marginal distribution P_x and is independent of the change of the conditional distribution $P_{y|x}$ over domains. This makes sense because we observed that feature structure can be transferred even though the conditional distribution differs arbitrarily across domains.

The obtained generalization error bound in Theorem 1 has a fast convergence rate with respect to the target domain training sample size n_t, of order $O(1/n_t)$. However, no benefit for increasing the source training sample size n_s is shown. This is because we have theoretically focused on justifying the transfer of feature structure, but not the transfer of labeling information, and we have assumed that the target domain feature structure is already given by the source domain training sample; under this circumstance, increasing the size of the source domain training sample may help little. In practice, however, more source domain training examples will provide a more completed feature structure and more labeling information. A large source domain training sample is therefore preferred.

In the rest of the paper, for simplicity, we will consider the case that the residual is zero, which occurs if the feature structure is perfectly provided by the source domain observations (i.e., the source domain observations can exactly reconstruct the target domain data). However, similar to the results shown in Theorem 3, our results can be easily extended to the case in which the residual $\varepsilon \neq 0$.

5 Learning to Learn

Learning to learn was formulated by [Baxter, 2000], where a probability measure $\mathcal{E}$ known as the environment was introduced on a set of related tasks. It aims to find learning algorithm that performs well on all of the tasks in the environment.

Let μ_l be the probability measure of the example $z_l = (x_l, y_l)$ in the context of the task l. We will use $S_l = \{(x_{l,1}, y_{l,1}), \ldots, (x_{l,n_l}, y_{l,n_l})\}$ to denote the training sample of the task l. Let h_{D,S_l} denote the predictor computed by a learning algorithm $\mathcal{A}_D$ using the training sample S_l, where D is a learning parameter for the learning algorithm $\mathcal{A}_D$. The notion of expected transfer risk [Maurer *et al.*, 2013] associated with the learning algorithm $\mathcal{A}_D$ is defined as follows:

$$R_{\mathcal{E}}(\mathcal{A}_D) = \mathbb{E}_{l\sim\mathcal{E}}\mathbb{E}_{S_l\sim\mu_l^{n_l}}\mathbb{E}_{z\sim\mu_l}\ell(y, h_{D,S_l}(x)). \tag{13}$$

The expected transfer risk can be estimated by the empirical transfer risk of a finite set of tasks. Many learning to learn algorithms therefore minimize the expected transfer risk by alternatively minimizing the empirical transfer risk of N tasks that have been drawn independently from the environment $\mathcal{E}$:

$$R_{n,N}(\mathcal{A}_D) = \frac{1}{N}\sum_{l=1}^{N}\frac{1}{n_l}\sum_{i=1}^{n_l}\ell(y_{l,i}, h_{D,S_l}(x_{l,i})). \tag{14}$$

The learning parameter D can be of many different forms, e.g., either a matrix or a vector. [Maurer *et al.*, 2013] defined D to be a matrix as a set of dictionary bases. [Evgeniou and Pontil, 2004] assumed D to be a vector as the overlap of the parameters among multiple tasks. Specifically, they assumed that all task parameters w_l can be written as $w_l = w_0 + v_l$[1]. We have proven that in this setting tasks could regularize tasks [Liu *et al.*, 2017b] and therefore the algorithms generalize fast [Liu *et al.*, 2017a].

In this paper, we consider the matrix case, where $h_{D,S_l}(x_l) = \langle D\gamma, x_l\rangle$. Then, the empirical risk minimization algorithm becomes [Maurer *et al.*, 2013]

$$\min_{D\in\mathcal{D}_k}\frac{1}{N}\sum_{l=1}^{N}\min_{\gamma\in\mathcal{C}_\gamma}\frac{1}{n_l}\sum_{i=1}^{n_l}\ell(y_{l,i}, \langle D\gamma, x_{l,i}\rangle), \tag{15}$$

where $\mathcal{D}_k$ is a set of k-dimensional dictionaries, every $D \in \mathcal{D}_k$ is a linear map from $\mathbb{R}^k$ to the hypothesis class H, and $\mathcal{C}_\gamma$ is a set of code vectors in $\mathbb{R}^k$ satisfying some properties, such as sparsity.

We apply our analysis scheme to learning to learn by focusing on the performance of one particular task. For simplicity, we assume there are two tasks. Let denote the domain of the focused task as the target domain and the other one the source domain. Learning to learn algorithms then solve the following optimization problem:

$$\min_{D}\sum_{k\in\{s,t\}}\min_{\gamma}\frac{1}{n_k}\sum_{i=1}^{n_k}\ell\left(y_i^k, \left\langle D\gamma, x_i^k\right\rangle\right). \tag{16}$$

[1]Constraints are needed to control a trade-off between w_0 and v_l. Many algorithms encourage w_0 to be large.

Let the non-zero parts of the target parameter γ^t and source parameter γ^s have an overlap γ_0, i.e., $\gamma^t = \gamma_0 + \Delta\gamma^t$. Using a proof method similar to that of Theorem 3, we show that "source" domain observations can be transferred as a feature structure to the "target" domain and function as a regularization matrix, which enables learning to learn algorithms to be uniformly stable with respect to the target domain.

Theorem 4 *Suppose Assumption 1 holds and $\varepsilon_2 = 0$. Let the loss function ℓ be c-strongly convex and σ-admissible. Algorithms that solve problem (16) are uniformly stable with respect to the target domain. For any learned $\gamma_0, \Delta\gamma^t \in \mathcal{H}$ and any (x^t, y^t) drawn from the target domain, the following holds:*

$$\left|\ell(y^t, \langle D(\gamma_{0,S_t} + \Delta\gamma^t), x^t\rangle) - \ell\left(y^t, \left\langle D(\gamma_{0,S_t^i} + \Delta\gamma^t), x^t\right\rangle\right)\right| \le \frac{2n_s\sigma^2 r_2^2}{cn_t}. \tag{17}$$

The proof method of Theorem 4 is the same as that of Theorem 3.

Theorem 4 shows that the learning of the overlapping information between source and target tasks will be benefit from feature structure transfer learning. The discussions on domain adaptation apply to learning to learn as well.

6 Conclusion

This paper studied the feature structure transfer learning problem. We theoretically justified that feature structure can be transferred, independently of the change of $P_{y|x}$ over domains, to improve the learning performance in the target domain. Motivated by self-taught learning, we discussed how feature structure can be transferred in domain adaptation and learning to learn settings from a regularization perspective. Our analysis implies that a tuning parameter is necessary to help transfer feature structure information from the source domain to the target domain. We believe that our analysis scheme applies to many other existing transfer learning settings as well.

The theoretical analyses in this paper stimulate our future work in two related directions. First, as we have interpreted the function of the transferred feature structure as regularization, we will consider introducing proper regularization parameters to improve the performance of existing transfer learning applications. Second, we plan to learn feature representation for (semi-supervised) feature structure transfer learning algorithms by designing a new criterion to match feature structure.

Acknowledgments

Liu and Tao were partially supported by Australian Research Council Projects FT-130101457, DP-140102164, LP-150100671.

A Appendix

A.1 Used Tool

We introduce Bregman divergence [Mohri *et al.*, 2012] to help upper bound the uniform stability.

Definition 4 (Bregman divergence) *Let $f : \mathcal{H} \to \mathbb{R}$ be a convex function. For all $s, t \in \mathcal{H}$, we have*

$$B_f(s\|t) = f(s) - f(t) - \langle s - t, \nabla f(t)\rangle, \tag{18}$$

where $\nabla f(t)$ denotes the gradient of function $f(t)$.

Bregman divergence has the following properties. Detailed discussions can be found in [Mohri *et al.*, 2012].

Lemma 1 *Bregman divergence is additive and non-negative. If $f = f_1 + f_2$, and both f_1 and f_2 are convex, for any $s, t \in \mathcal{H}$, we have*

$$B_f(s\|t) = B_{f_1}(s\|t) + B_{f_2}(s\|t) \quad \textit{and} \quad B_f(s\|t) \geq 0. \tag{19}$$

A.2 Proof of Theorem 3

In this subsection, we prove that algorithms solving the following optimization problem:

$$\min_w R_{n_s+n_t,\alpha} = \alpha R_{n_t}(w) + (1-\alpha) R_{n_s}(w), \tag{20}$$

are uniformly stable with respect to the target domain.

Let

$$g_{S_t}(w) = \alpha R_{n_t}(w) = \frac{\alpha}{n_t} \sum_{i=1}^{n_t} \ell\left(y_i^t, \langle w, x_i^t\rangle\right), \tag{21}$$

$$g_{S_s}(w) = (1-\alpha) R_{n_s}(w) = (1-\alpha) \sum_{i=1}^{n_s} \ell\left(y_i^s, \langle w, x_i^s\rangle\right) \tag{22}$$

and

$$g_{S_t,S_s}(w) = R_{n_s+n_t,\alpha} = g_{S_t}(w) + g_{S_s}(w). \tag{23}$$

By abusing the notation a little bit without confusion, we let w be the solution to optimization problem (20) when the input training sample is S_t and w' be the solution when the input training sample is S_t^i. Since the employed loss function is convex. According to the non-negative and additive properties of Bregman divergence, we have

$$\begin{aligned} &B_{g_{S_t,S_s}}(w'\|w) + B_{g_{S_t^i,S_s}}(w\|w') \\ &\geq B_{g_{S_s}}(w'\|w) + B_{g_{S_s}}(w\|w'). \end{aligned} \tag{24}$$

To upper bound the left-hand side of inequality (24), we have

$$\begin{aligned} &B_{g_{S_t,S_s}}(w'\|w) + B_{g_{S_t^i,S_s}}(w\|w') \\ &= g_{S_t,S_s}(w') - g_{S_t,S_s}(w) - \langle w' - w, \nabla g_{S_t,S_s}(w)\rangle \\ &+ g_{S_t^i,S_s}(w) - g_{S_t^i,S_s}(w') - \left\langle w - w', \nabla g_{S_t^i,S_s}(w')\right\rangle \\ &= \frac{\alpha}{n_t}\left(\ell\left(y_i^t, \langle w', x_i^t\rangle\right) - \ell\left(y_i^t, \langle w, x_i^t\rangle\right)\right. \\ &\left. + \ell\left(y'^t_i, \left\langle w, x'^t_i\right\rangle\right) - \ell\left(y'^t_i, \left\langle w', x'^t_i\right\rangle\right)\right) \\ &\leq \frac{\alpha\sigma}{n_t}\left(\left|\langle w - w', x_i^t\rangle\right| + \left|\left\langle w - w', x'^t_i\right\rangle\right|\right) \\ &\quad \text{(Using Assumption 2)} \\ &= \frac{\alpha\sigma}{n_t}\left(\left|\left\langle w - w', \sum_{j=1}^{n_s}\gamma_j x_j^s\right\rangle\right| + \left|\langle w - w', \eta_2\rangle\right|\right. \\ &\left. + \left|\left\langle w - w', \sum_{j=1}^{n_s}\gamma_j' x_j^s\right\rangle\right| + \left|\langle w - w', \eta_2'\rangle\right|\right) \\ &\quad \text{(Using Cauchy-Schwarz inequality)} \\ &\leq \frac{2\alpha\sigma(r_2 + O(\varepsilon_2))\sqrt{\sum_j \left\langle w - w', x_j^s\right\rangle^2}}{n_t}. \end{aligned} \tag{25}$$

To lower bound the right-hand side of inequality (24), we consider two different forms of loss function: (i) $\ell(y, \langle w, x\rangle) = \ell(y - \langle w, x\rangle)$ and (ii) $\ell(y, \langle w, x\rangle) = \ell(y\langle w, x\rangle)$. When the loss function is of form (i), we have

$$\begin{aligned} &B_{g_{S_s}}(w'\|w) + B_{g_{S_s}}(w\|w') \\ &= -\left\langle w' - w, (1-\alpha)\sum_{j=1}^{n_s}\nabla\ell\left(y_j^s, \langle w, x_j^s\rangle\right)x_j^s\right\rangle \\ &- \left\langle w - w', (1-\alpha)\sum_{j=1}^{n_s}\nabla\ell\left(y_j^s, \langle w', x_j^s\rangle\right)x_j^s\right\rangle \\ &\quad \text{(Since } \ell \text{ is } c\text{-strongly convex)} \\ &\geq c(1-\alpha)\sum_j \left\langle w - w', x_j^s\right\rangle^2. \end{aligned} \tag{26}$$

When the loss function is of form (ii), we have

$$\begin{aligned} &B_{g_{S_s}}(w'\|w) + B_{g_{S_s}}(w\|w') \\ &= -\left\langle w' - w, (1-\alpha)\sum_{j=1}^{n_s}\nabla\ell\left(y_j^s, \langle w, x_j^s\rangle\right)x_j^s y_j^s\right\rangle \\ &- \left\langle w - w', (1-\alpha)\sum_{j=1}^{n_s}\nabla\ell\left(y_j^s, \langle w', x_j^s\rangle\right)x_j^s y_j^s\right\rangle \\ &\geq c(1-\alpha)\sum_j \left\langle w - w', x_j^s y_j^s\right\rangle^2 \\ &= c(1-\alpha)\sum_j \left\langle w - w', x_j^s\right\rangle^2. \end{aligned} \tag{27}$$

Combining (24), (25), (26) and (27), we have

$$\sqrt{\sum_j \left\langle w - w', x_j^s\right\rangle^2} \leq \frac{2\alpha\sigma(r_2 + O(\varepsilon_2))}{c(1-\alpha)n_t}. \tag{28}$$

Thus, for any training sample $S \in (\mathcal{H} \times \mathbb{R})^n$, any $i \in \{1, ..., n\}$ and any $z \in \mathcal{H} \times \mathbb{R}$, $x^t = \sum_{j=1}^{n_s}\gamma_j x_j^s + \eta_2$, the following holds:

$$\begin{aligned} &\left|\ell(y^t - \langle w, x^t\rangle) - \ell\left(y^t - \langle w', x^t\rangle\right)\right| \\ &\leq \sigma\left|\left\langle w - w', \sum_{j=1}^{n_s}\gamma_j x_j^s + \eta_2\right\rangle\right| \\ &\leq \sigma\sqrt{\sum_j \left\langle w - w', x_j^s\right\rangle^2}(r_2 + O(\varepsilon_2)) \\ &\leq \frac{2\alpha\sigma^2(r_2 + O(\varepsilon_2))^2}{c(1-\alpha)n_t}. \end{aligned} \tag{29}$$

This completes the proof of Theorem 4. ■

References

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

[Baxter, 2000] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12(1):149–198, 2000.

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007.

[Ben-David *et al.*, 2010] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[Blitzer *et al.*, 2008] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *NIPS*, 2008.

[Bousquet and Elisseeff, 2002] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[Evgeniou and Pontil, 2004] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *SIGKDD*, 2004.

[Gong *et al.*, 2016] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *ICML*, pages 2839–2848, 2016.

[Huang *et al.*, 2006] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *NIPS*, 2006.

[Kakade and Tewari, 2009] Sham M Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In *NIPS*, 2009.

[Kuzborskij and Orabona, 2013] Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *ICML*, 2013.

[Li *et al.*, 2015] Ya Li, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Multi-task model and feature joint learning. In *IJCAI*, pages 3643–3649, 2015.

[Liu *et al.*, 2016] Hongfu Liu, Ming Shao, and Yun Fu. Structure-preserved multi-source domain adaptation. In *ICDM*, pages 1059–1064, 2016.

[Liu *et al.*, 2017a] Tongliang Liu, Gábor Lugosi, Gergely Neu, and Dacheng Tao. Algorithmic stability and hypothesis complexity. *arXiv preprint arXiv:1702.08712*, 2017.

[Liu *et al.*, 2017b] Tongliang Liu, Dacheng Tao, Mingli Song, and Stephen J Maybank. Algorithm-dependent generalization bounds for multi-task learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):227–241, 2017.

[Long *et al.*, 2015] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[Luo *et al.*, 2014] Yong Luo, Tongliang Liu, Dacheng Tao, and Chao Xu. Decomposition-based transfer distance metric learning for image classification. *IEEE Transactions on Image Processing*, 23(9):3789–3801, 2014.

[Mansour *et al.*, 2009] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009.

[Maurer *et al.*, 2013] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. In *ICML*, 2013.

[Maurer, 2009] Andreas Maurer. Transfer bounds for linear feature learning. *Machine Learning*, 75(3):327–350, 2009.

[Mohri *et al.*, 2012] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT Press, 2012.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Pentina and Lampert, 2014] Anastasia Pentina and Christoph H Lampert. A PAC-bayesian bound for lifelong learning. In *ICML*, 2014.

[Raina *et al.*, 2007] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, 2007.

[Shao *et al.*, 2014] Ming Shao, Dmitry Kit, and Yun Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109(1-2):74–93, 2014.

[Shao *et al.*, 2016] Ming Shao, Zhengming Ding, Handong Zhao, and Yun Fu. Spectral bisection tree guided deep adaptive exemplar autoencoder for unsupervised domain adaptation. In *AAAI*, 2016.

[Si *et al.*, 2010] Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929–942, 2010.

[Wang and Schneider, 2014] Xuezhi Wang and Jeff Schneider. Flexible transfer learning under support and model shift. In *NIPS*, 2014.

[Zhang *et al.*, 2013] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *ICML*, 2013.

[Zhang *et al.*, 2015] Kun Zhang, Mingming Gong, and Bernhard Scholkopf. Multi-source domain adaptation: a causal view. In *AAAI*, pages 3150–3157, 2015.