

Hypothesis Transfer Learning via Transformation Functions

Authored by:

Aarti Singh
Barnabas Poczos
Simon S. Du
Jayanth Koushik

Abstract

We consider the Hypothesis Transfer Learning (HTL) problem where one incorporates a hypothesis trained on the source domain into the learning procedure of the target domain. Existing theoretical analysis either only studies specific algorithms or only presents upper bounds on the generalization error but not on the excess risk. In this paper, we propose a unified algorithm-dependent framework for HTL through a novel notion of transformation functions, which characterizes the relation between the source and the target domains. We conduct a general risk analysis of this framework and in particular, we show for the first time, if two domains are related, HTL enjoys faster convergence rates of excess risks for Kernel Smoothing and Kernel Ridge Regression than those of the classical non-transfer learning settings. We accompany this framework with an analysis of cross-validation for HTL to search for the best transfer technique and gracefully reduce to non-transfer learning when HTL is not helpful. Experiments on robotics and neural imaging data demonstrate the effectiveness of our framework.

1 Paper Body

We consider the Hypothesis Transfer Learning (HTL) problem where one incorporates a hypothesis trained on the source domain into the learning procedure of the target domain. Existing theoretical analysis either only studies specific algorithms or only presents upper bounds on the generalization error but not on the excess risk. In this paper, we propose a unified algorithm-dependent framework for HTL through a novel notion of transformation function, which characterizes the relation between the source and the target domains. We conduct a general risk analysis of this framework and in particular, we show for the first time, if two domains are related, HTL enjoys faster convergence rates of excess risks for Kernel Smoothing and Kernel Ridge Regression than those

of the classical non-transfer learning settings. Experiments on real world data demonstrate the effectiveness of our framework.

1

Introduction

In a classical transfer learning setting, we have a large amount of data from a source domain and a relatively small amount of data from a target domain. These two domains are related but not necessarily identical, and the usual assumption is that the hypothesis learned from the source domain is useful in the learning task of the target domain. In this paper, we focus on the regression problem where the functions we want to estimate of the source and the target domains are different but related. Figure 1a shows a 1D toy example of this setting, where the source function is $f_{\text{so}}(x) = \sin(4\pi x)$ and the target function is $f_{\text{ta}}(x) = \sin(4\pi x) + 4\pi x$. Many real world problems can be formulated as transfer learning problems. For example, in the task of predicting the reaction time of an individual from his/her fMRI images, we have about 30 subjects but each subject has only about 100 data points. To learn the mapping from neural images to the reaction time of a specific subject, we can treat all but this subject as the source domain, and this subject as the target domain. In Section 6, we show how our proposed method helps us learn this mapping more accurately. This paradigm, hypothesis transfer learning (HTL) has been explored empirically with success in many applications [Fei-Fei et al., 2006, Yang et al., 2007, Orabona et al., 2009, Tommasi et al., 2010, Kuzborskij et al., 2013, Wang and Schneider, 2014]. Kuzborskij and Orabona [2013, 2016] pioneered the theoretical analysis of HTL for linear regression and recently Wang and Schneider [2015] analyzed Kernel Ridge Regression. However, most existing works only provide generalization bounds, i.e. the difference between the true risk and the training error or the leave-one-out error. These analyses are not complete because minimizing the generalization error does not necessarily

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

12 10

6

1

0.5

2

Offset KS Only Target KS f_{ta}

Scale KS Only Target KS f_{ta}

1 0

Y

Y

8

3

f_{so} Source data f_{ta} Target Data

Y

14

4

21

0
2
?2
0 ?2 0
0.2
0.4
0.6
0.8
1
?0.5 0
0.1
0.2 X
X

(a) A toy example of transfer learning. We have many more samples from the source domain than the target domain.

0.3
0.4
?3 0
0.1
0.2 X
0.3
0.4

(b) Transfer learning with Offset Transformation.

0.3
0.4
?3 0
0.1
0.2 X
0.3
0.4

(c) Transfer learning with Scale Transformation.

Figure 1: Experimental results on synthetic data. reduce the true risk. Further, these works often rely on a particular form of transformation from the source domain to the target domain. For example, Wang and Schneider [2015] studied the offset transformation that instead of estimating the target domain function directly, they learn the residual between the target domain function and the source domain function. It is natural to ask what if we use other transfer functions and how it affects the risk on the target domain. In this paper, we propose a general framework of HTL. Instead of analyzing a specific form of transfer, we treat it as an input of our learning algorithm. We call this input transformation function since intuitively, it captures the relevance between these two domains.¹ This framework unifies many previous works Wang and Schneider [2014], Kuzborskij and Orabona [2013], Wang et al. [2016] and naturally induces a class of new learning procedures. Theoretically, we develop excess risk analysis for this framework. The performance depends on the stability [Bousquet and Elisseeff, 2002] of the algorithm used as a subroutine that if the algorithm is stable then the estimation error in the source domain will not affect the estimation in the target domain much. To our knowledge, this connection was first established by Kuzborskij et al. [2013] in the linear regression setting but here we generalize it to a broader context. In particular, we provide explicit risk bounds for two widely used nonlinear estimators, Kernel Smoothing (KS) estimators and Kernel Ridge Regression (KRR) as subroutines. To the best of our knowledge, these are the first results showing when two domains are related,

transfer learning techniques have faster statistical convergence rate of excess risk than that of non-transfer learning of kernel based methods. Further, we accompany this framework with a theoretical analysis showing a small amount of data for crossvalidation enables us (1) avoid using HTL when it is not useful and (2) choose the best transformation function as input from a large pool. The rest of the paper is organized as follows. In Section 2 we introduce HTL and provide necessary backgrounds for KS and KRR. We formalize our transformation function based framework in Section 3. Our main theoretical results are in Section 4 and specifically in Section 4.1 and Section 4.2 we provide explicit risk bounds for KS and KRR, respectively. In Section 5 we analyze crossvalidation in HTL setting and in Section 6 we conduct experiments on real world data data. We conclude with a brief discussion of avenues for future work.

2

Preliminaries

2.1

Problem Setup

In this paper, we assume both $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}$ lie in compact subsets: $X \in \mathcal{X} \subset \mathbb{R}^d$, $Y \in \mathcal{Y} \subset \mathbb{R}$ for some $\mathcal{X} \subset \mathbb{R}^d$, $\mathcal{Y} \subset \mathbb{R}$. Throughout the paper, we use $T = \{(X_i, Y_i)\}_{i=1}^n$ to denote a set of n samples. Let (X, Y) be the sample from the source domain, and (X, Y) the sample from the target domain. In our setting, there are n_s samples drawn i.i.d from the source distribution: $T_s = \{(X_{s,i}, Y_{s,i})\}_{i=1}^{n_s}$, and n_t samples drawn i.i.d from the target distribution: $T_t = \{(X_{t,i}, Y_{t,i})\}_{i=1}^{n_t}$. In addition, we also use n_{val} samples drawn i.i.d from the target domain for cross-validation. We model the joint relation between X and Y by: $Y_s = f_s(X_s) + \epsilon_s$ and $Y_t = f_t(X_t) + \epsilon_t$ where f_s and f_t are regression functions and we assume the noise $E[\epsilon_s] = E[\epsilon_t] = 0$, i.i.d, 1

We formally define the transformation functions in Section 3.

2

and bounded. We use $A : T \rightarrow \mathbb{R}$ to denote an algorithm that takes a set of samples T and produce \hat{f} an estimator. Given an estimator \hat{f} , we define the integrated L2 risk as $R(\hat{f}) = E \int \hat{f}(X) - Y$

where the expectation is taken over the distribution of (X, Y) . Similarly, the empirical L2 risk on $T = \{(X_i, Y_i)\}_{i=1}^n$ a set of sample T is defined as $R(\hat{f}) = \frac{1}{n} \sum_{i=1}^n (\hat{f}(X_i) - Y_i)^2$. In HTL setting, we use \hat{f}_s an

i=1

estimator from the source domain to facilitate the learning procedure for \hat{f}_t .

2.2

Kernel Smoothing

We say a function f is in the (γ, β) Hölder class [Wasserman, 2006], if for any $x, x' \in \mathbb{R}^d$, f satisfies $|f(x) - f(x')| \leq C \|x - x'\|^\gamma$, for some $C \in (0, 1)$. The kernel smoothing method uses a positive kernel K on $[0, 1]$, highest at 0, decreasing on $[0, 1]$, 0 outside $[0, 1]$, and $\int_{\mathbb{R}^d} K(u) du = 1$. Using n $T = \{(X_i, Y_i)\}_{i=1}^n$, the kernel smoothing estimator is defined as follows:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n w_i(x) Y_i, \quad w_i(x) = \frac{K(x, X_i)}{\sum_{j=1}^n K(x, X_j)}$$

2.3

Kernel Ridge Regression

Another popular non-linear estimator is the kernel ridge regression (KRR) which uses the theory of reproducing kernel Hilbert space (RKHS) for regression [Vovk, 2013]. Any symmetric positive semidefinite kernel function $K : \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathbb{R}$ defines a RKHS H . For each $x \in \mathcal{R}^d$, the function $z \mapsto K(z, x)$ is contained in the Hilbert space H ; moreover, the Hilbert space is endowed with an inner product $\langle \cdot, \cdot \rangle_H$ such that $K(\cdot, x)$ acts as the kernel of the evaluation functional, meaning $\langle f, K(\cdot, x) \rangle_H = f(x)$ for $f \in H$. In this paper we assume K is bounded: $\sup_{x, y \in \mathcal{R}^d} K(x, y) = k_0 < \infty$. Given the inner product, the H norm of a function $g \in H$ is defined as $\|g\|_H = \sqrt{\langle g, g \rangle_H}$ and similarly the L_2 norm, $\|g\|_2 = \sqrt{\int_{\mathcal{R}^d} g(x)^2 dP_X}$ for a given P_X . Also, the kernel induces an integral operator $TK : L_2(P_X) \rightarrow L_2(P_X)$: $TK[f](x) = \int_{\mathcal{R}^d} K(x, y) f(y) dP_X(y)$ with countably many non-zero eigenvalues: $\{\lambda_i\}_{i=1}^\infty$. For a given function f , the approximation error is $\|f - \sum_{i=1}^n \lambda_i^{-1} \langle f, K(\cdot, X_i) \rangle_H K(\cdot, X_i)\|_H^2$ defined as: $A_f(n) = \inf_{h \in H} \|h - f\|_H^2 = \sum_{i=n+1}^\infty \lambda_i$ for $n \geq 0$. Finally the estimated function \hat{f}_n evaluated at point x can be written as $\hat{f}_n(x) = \frac{1}{n} \sum_{i=1}^n K(x, X_i) (K(X_i, X_i) + \lambda)^{-1} Y_i$ where $X \in \mathcal{R}^d$ are the inputs of training samples and $Y \in \mathbb{R}$ are the training labels Vovk [2013].

2.4 Related work

Before we present our framework, it is helpful to give a brief overview of existing literature on theoretical analysis of transfer learning. Many previous works focused on the settings when only unlabeled data from the target domain are available [Huang et al., 2006, Sugiyama et al., 2008, Yu and Szepesvári, 2012]. In particular, a line of research has been established based on distribution discrepancy, a loss induced metric for the source and target distributions [Mansour et al., 2009, Ben-David et al., 2007, Blitzer et al., 2008, Cortes and Mohri, 2011, Mohri and Medina, 2012]. For example, recently Cortes and Mohri [2014] gave generalization bounds for kernel based methods under convex loss in terms of discrepancy. In many real world applications such as yield prediction from pictures [Nuske et al., 2014], or prediction of response time from fMRI [Verstynen, 2014], some labeled data from the target domain is also available. Cortes et al. [2015] used these data to improve their discrepancy minimization algorithm. Zhang et al. [2013] focused on modeling target shift ($P(Y)$ changes), conditional shift ($P(X|Y)$ changes), and a combination of both. Recently, Wang and Schneider [2014] proposed a kernel mean embedding method to match the conditional probability in the kernel space and later derived generalization bound for this problem Wang and Schneider [2015]. Kuzborskij and Orabona [2013, 2016], Kuzborskij et al. [2016] gave excess risk bounds for target domain estimator in the form of a linear combination of estimators from multiple source domains and an additional linear function. Ben-David and Uner [2013] showed a similar bound of the same setting with different quantities capturing the relatedness. Wang et al. [2016] showed that if the features of source and target domain are $[0, 1]^d$, using orthonormal basis function estimator, transfer

learning achieves better excess risk 3

guarantee if $f_{\text{ta}} \approx f_{\text{so}}$ can be approximated by the basis functions easier than f_{ta} . Their work can be viewed as a special case of our framework using the transformation function $G(a, b) = a + b$.

3

Transformation Functions

In this section, we first define our class of models and give a meta-algorithm to learn the target regression function. Our models are based on the idea that transfer learning is helpful when one transforms the target domain regression problem into a simpler regression problem using source domain knowledge. Consider the following example. $f_{\text{ta}}(x) = f_{\text{so}}(x) + x$. Example: Offset Transfer. Let $f_{\text{so}}(x) = x(1 - x) \sin x + 0.05$ is the so called Doppler function. It requires a large number of samples to estimate well because of its lack of smoothness Wasserman [2006]. For the same reason, f_{ta} is also difficult to estimate directly. However, if we have enough data from the source domain, we can have a fairly good estimate of f_{so} . Further, notice that the offset function $w(x) = f_{\text{ta}}(x) - f_{\text{so}}(x) = x$, is just a linear function. Thus, instead of directly using T_{ta} to estimate f_{ta} , we can use the target domain samples to find an estimate of $w(x)$, denoted by $\hat{w}(x)$, and our estimator for the target domain is just: $\hat{f}_{\text{ta}}(x) = \hat{f}_{\text{so}}(x) + \hat{w}(x)$. Figure 1b shows this technique gives improved fitting for f_{ta} . The previous example exploits the fact that function $w(x) = f_{\text{ta}}(x) - f_{\text{so}}(x)$ is a simpler function than f_{ta} . Now we generalize this idea further. Formally, we define the transformation function as $G(a, b) : \mathcal{R}^2 \rightarrow \mathcal{R}$ where we assume that given $a \in \mathcal{R}$, $G(a, \cdot)$ is invertible. Here a will be the regression function of the source domain evaluated at some point and the output of G will be the regression function of the target domain evaluated at the same point. Let $G^{-1}_a(\cdot)$ denote the inverse of $G(a, \cdot)$ such that $G(a, G^{-1}_a(c)) = c$. For example if $G(a, b) = a + b$ and $G^{-1}_a(c) = c - a$. For a fixed a given G and a pair (f, \hat{f}) , they together induce a function $w_G(x) = G(f_{\text{so}}(x), \hat{f}_{\text{ta}}(x))$. In the offset transfer example, $w_G(x) = x$. By this definition, for any x , we have $G(f_{\text{so}}(x), w_G(x)) = \hat{f}_{\text{ta}}(x)$. We call w_G the auxiliary function of the transformation function G . In the HTL setting, G is a user-defined transformation that represents users' prior knowledge on the relation between the source and target domains. Now we list some other examples: Example: Scale-Transfer. Consider $G(a, b) = ab$. This transformation function is useful when f_{so} and f_{ta} satisfy a smooth scale transfer. For example, if $f_{\text{ta}} = cf_{\text{so}}$, for some constant c , then $w_G(x) = c$ because $f_{\text{ta}}(x) = G(f_{\text{so}}(x), w_G(x)) = f_{\text{so}}(x) w_G(x) = f_{\text{so}}(x) c$. See Figure 1c. Example: Non-Transfer. Consider $G(a, b) = b$. Notice that $f_{\text{ta}}(x) = w_G(x)$ and so f_{so} is irrelevant. Thus this model is equivalent to traditional regression on the target domain since data from the source domain does not help. 3.1

A Meta Algorithm

Given the transformation G and data, we provide a general procedure to estimate f_{ta} . The spirit of the algorithm is turning learning a complex function f_{ta} into an easier function w_G . First we use an algorithm A_{so} that takes T_{so} to obtain \hat{f}_{so} . Since we have sufficient data from the source domain, .

Second, we construct a new data set using f^* so should be close to the true regression function f^* so that w^*G is close to f^* so that the data points from the target domain: $T = \{X_i, Y_i\}_{i=1}^n$, where

$$Y_i = f^*(X_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

?

i

i

$i=1$

$$Y_i = f^*(X_i) + \epsilon_i$$

where ϵ_i and the expectation is taken over ϵ_i . Thus, we can use these newly constructed data to $G = \text{AWG}(T, w^*G)$. Finally, we plug trained f^* so and w^*G into learn w^*G with algorithm $\text{AWG} : w^*G(X)$. Pseudocode is transformation G to obtain an estimation for f^* so that $f^*(X) = G(f^*(X))$, w shown in Algorithm 1. Unbiased Estimator $\text{HG}(f^*(X), Y)$: In Algorithm 1, we require an unbiased estimator for $w^*G(X)$. Note that if $G(a, b)$ is linear in b or $\epsilon = 0$, we can simply set $\text{HG}(f^*(X), Y) =$

$$G(f^*(X), Y)$$

For other scenarios, $G(f^*(X), Y)$

(Y) is biased: $E[G(f^*(X), Y)] \neq f^*(X)$ and we need to design estimator using the structure of G . $f^*(x)$ (f 4

$\epsilon =$

Algorithm 1 Transformation Function based Transfer Learning so Inputs: Source domain data: $T_s = \{(X_i, Y_i)\}_{i=1}^n$, target domain data: $T_t = \{(X_i, Y_i)\}_{i=1}^n$, transformation function: G , algorithm to train f^* so : Aso , algorithm to train $w^*G : \text{AwG}$ and HG an unbiased estimator for estimating w^*G . Outputs: Regression function for the target domain: f^* so that 1: Train the source domain regression function f^* so that $\text{Aso}(T_s)$. 2: Construct new data using f^* so and T_t : $T_w = \{(X_i, W_i)\}_{i=1}^n$, where $W_i = f^*(X_i) + \epsilon_i$ so that $\text{HG}(f^*(X), Y)$.

i

i

3: Train the auxiliary function: $w^*G = \text{AwG}(T_w)$.

?

?

4: Output the estimated regression for the target domain: $f^*(X) = G(f^*(X), w^*G(X))$.

Remark 1: Many transformation functions are equivalent to a transformation function $G(a, b)$ where $G(a, b)$ is linear in b . For example, for $G(a, b) = ab^2$, i.e., $f^*(x) = f^*(x) w^*G(x)$, consider $G(a, b) = ab$ where b in G stands for b in G , i.e., $f(x) = f(x) w^*G(x)$. Therefore $w^*G = w^*G$ and we only need to estimate w^*G well instead of estimating w^*G . More generally, if $G(a, b)$ can be factorized as $G(a, b) = g_1(a) g_2(b)$, i.e., $f^*(x) = g_1(f^*(x)) g_2(w^*G(x))$, we only need to estimate $g_2(w^*G(x))$ and the convergence rate depends on the structure of $g_2(w^*G(x))$. Remark 2: When G is not linear in b and $\epsilon \neq 0$, observe that in Algorithm 1, we treat Y as

happen to know f_{so} , then the error is bounded by $O(\frac{1}{\sqrt{n_{\text{so}}}})$. However, since we are using estimated f_{so} to construct training samples for wG , the error might accumulate as n_{ta} increases. Though the third term in Theorem 1 might increase with n_{ta} , it also depends on the estimation error of f_{so} which is relatively small because of the large amount of source domain data. The stability condition (1) we used is related to the uniform stability introduced by Bousquet and Elisseeff [2002] where they consider how much will the output change if one of the training instance is removed or replaced by another whereas ours depends on two different training data sets. The connection between transfer learning and stability has been discovered by Kuzborskij and Orabona [2013], Liu et al. [2016] and Zhang [2015] in different settings, but they only showed bounds for generalization, not for excess risk. 4.1

Kernel Smoothing

We first analyze kernel smoothing method. Theorem 2 Suppose the support of X_{ta} is a subset of the support of X_{so} and the probability density of PX_{so} and PX_{ta} are uniformly bounded away from below on their supports. Further assume f_{so} is $(\frac{1}{\sqrt{n_{\text{so}}}}, \frac{1}{\sqrt{n_{\text{so}}}})$ -Hölder and wG is $(\frac{1}{\sqrt{n_{\text{so}}}}, \frac{1}{\sqrt{n_{\text{so}}}})$ -Hölder. If we use kernel smoothing estimation for f_{so} and wG with bandwidth $h_{\text{so}} = \frac{1}{\sqrt{n_{\text{so}}}}$ and $h_{wG} = \frac{1}{\sqrt{n_{\text{so}}}}$, with probability at least $1 - \frac{1}{n_{\text{ta}}}$, the risk satisfies: $E[R_f] - E[R_{f^*}] = O(\frac{1}{\sqrt{n_{\text{so}}}})$ where the expectation is taken over T_{so} and T_{ta} .

Theorem 2 suggests that the risk depends on two sources, one from estimation of f_{so} and one from estimation of wG . For the first term, since in the typical transfer learning scenarios $n_{\text{so}} \gg n_{\text{ta}}$, it is relatively small in the setting we focus on. The second term shows the power of transfer learning on transforming a possibly complex target regression function into a simpler auxiliary function. It is well known that learning f_{ta} only using target domain has risk of the order $\frac{1}{\sqrt{n_{\text{ta}}}}$. Thus, if the auxiliary function is smoother than the target regression function, i.e. $wG \in \mathcal{H}_{\text{ta}}$, we obtain better statistical rate. 4.2

Kernel Ridge Regression

Next, we give an upper bound for the excess risk using KRR: Theorem 3 Suppose $PX_{\text{so}} = PX_{\text{ta}}$ and the eigenvalues of the integral operator TK satisfy $\lambda_i \geq \frac{c}{i^p}$ for $i \geq 1$ and $p \in (0, 1)$ and there exists a constant $C > 0$ such that for $f \in \mathcal{H}$, $\|f\|_{\mathcal{H}} \leq C$ and $\|f\|_{L_2(PX)} \leq 1$. Further assume that $Af(\cdot) = c_{\text{so}} f_{\text{so}}$ and $AwG(\cdot) = c_{wG} wG$. $\frac{1}{\sqrt{n_{\text{so}} + p}}$

If we use KRR for estimating f_{so} and wG with regularization parameters $\gamma_{\text{so}} = \frac{1}{n_{\text{so}}}$

and

$$\gamma_{wG} = \frac{1}{n_{\text{so}} + p}$$

, then with probability at least $1 - \frac{1}{n_{\text{ta}}}$ the excess risk satisfies: $E[R_f] - E[R_{f^*}] = O(\frac{1}{\sqrt{n_{\text{so}} + p}})$ where the expectation is taken over T_{so} and T_{ta} .

Similar to Theorem 2, Theorem 3 suggests that the estimation error comes from two sources. For estimating the auxiliary function wG , the statistical rate depends on properties of the kernel induced RKHS, and how far the auxiliary function is from this space. For the ease of presentation, we assume so that $PX_{so} = PX_{ta}$, so the approximation errors Af and A^*f are defined on the same domain. The L^2 error of estimating f_{so} is amplified by $O(\sqrt{wG \log(n_{ta})})$, which is worse than that of nonparametric kernel smoothing. We believe this L^2 is nearly tight because and Elisseeff have shown $wG \leq Bousquet$ the uniform algorithmic stability parameter for KRR is $O(\sqrt{Bousquet}$ and Elisseeff [2002]. wG Steinwart et al. [2009] showed that for non-transfer learning, the optimal statistical L^2 rate for excess risk is $\sqrt{n_{ta}}$, so if $wG \leq \sqrt{n_{ta}}$ and n_{so} is sufficiently large then we achieve improved convergence rate through transfer learning.

Remark: Theorem 2 and 3 are not directly comparable because our assumptions on the function spaces of these two theorems are different. In general, Hilbert space is only a Banach space but not a Hilbert space. We refer readers to Theorem 1 in Zhou [2008] for details.

5

Finding the Best Transformation Function

In the previous section we showed for a specific transformation function G , if auxiliary function is smoother than the target regression function then we have smaller excess risk. In practice, we would like to try out a class of transformation functions G , which is possibly uncountable. We can construct a subset of G G_0 , which is finite and satisfies that each G in G there is a G in G_0 that is close to G . Here we give an example. Consider the transformation functions that have the form: $G = \{G(a, b) = \alpha a + b \text{ where } \alpha \in [L, -a] \text{ where } -a \in [-La, -a]\}$. We can quantize this set of transformation functions by considering a subset of G : $G_0 = \{G(a, b) = k\alpha a + b\}$ where $\alpha \in [L, -a]$, $k = \frac{1}{2K}$, $\alpha \in [L, -a]$, $0, \frac{1}{2K}, \dots, K$ and $-a \in [-La, -a]$. Here $\frac{1}{2K}$ is the quantization unit. The next theorem shows we only need to search the transformation function G in G_0 whose corresponding estimator f^*G has the lowest empirical risk on the validation dataset.

Theorem 4 Let G be a class of transformation functions and G_0 be its $\frac{1}{2K}$ -norm L^2 -cover. Suppose wG satisfies the same assumption in Theorem 1 and for any two $G_1, G_2 \in G_0$, $\|wG_1 - wG_2\|_{L^2} \leq L \|G_1 - G_2\|_{L^2}$ for some constant L . Denote $G^* = \arg\min_{G \in G} R(f^*G)$ and $G_0^* = \arg\min_{G \in G_0} R(f^*G)$. If we choose $\alpha = 0$ and $G = \arg\min_{G \in G_0} R$ and $n_{val} = \frac{1}{\log \frac{1}{\epsilon} \frac{1}{\delta}}$, then $\mathbb{E} R(f^*G) - \mathbb{E} R(f^*G_0^*) \leq O(\frac{1}{\sqrt{n_{val}}})$ with probability at least $1 - \epsilon - \delta$, $\mathbb{E} R(f^*G) - \mathbb{E} R(f^*G_0^*) = O(\frac{1}{\sqrt{n_{val}}})$ where the \mathbb{E} expectation is taken over T_{so} and T_{ta} .

Remark 1: This theorem implies that if no-transfer function ($G(a, b) = b$) is in G_0 then we will end up choosing a transformation function that has the same order of excess risk as using no-transfer learning algorithm, thus avoiding negative transfer. Remark 2: Note number of validation set is only logarithmically depending on the size of set of transformation functions. Therefore, we only need to use a very small amount of data from the target domain to do cross-validation.

Experiments

In this section we use robotics and neural imaging data to demonstrate the effectiveness of the proposed framework. We conduct experiments on real-world data sets with the following procedures. 7

Only Target KS Only Target KRR Only Source KRR Combined KS Combined KRR CDM Offset KS Offset KRR Scale KS Scale KRR

nta = 10

nta = 20

nta = 40

nta = 80

nta = 160

nta = 320

0.086 ? 0.022 0.080 ? 0.017 0.098 ? 0.017 0.092 ? 0.011 0.087 ? 0.025 0.105
? 0.023 0.080 ? 0.026 0.146 ? 0.112 0.078 ? 0.022 0.102 ? 0.033

0.076 ? 0.010 0.078 ? 0.022 0.098 ? 0.017 0.084 ? 0.008 0.077 ? 0.015 0.074
? 0.020 0.066 ? 0.023 0.066 ? 0.017 0.065 ? 0.013 0.095 ? 0.100

0.066 ? 0.008 0.063 ? 0.013 0.098 ? 0.017 0.077 ? 0.009 0.062 ? 0.009 0.064
? 0.008 0.052 ? 0.006 0.053 ? 0.007 0.056 ? 0.009 0.057 ? 0.014

0.064 ? 0.007 0.050 ? 0.007 0.098 ? 0.017 0.075 ? 0.006 0.061 ? 0.005 0.060
? 0.007 0.054 ? 0.006 0.048 ? 0.006 0.056 ? 0.005 0.052 ? 0.010

0.065 ? 0.006 0.048 ? 0.006 0.098 ? 0.017 0.074 ? 0.006 0.047 ? 0.003 0.053
? 0.009 0.050 ? 0.003 0.043 ? 0.004 0.054 ? 0.008 0.044 ? 0.004

0.063 ? 0.005 0.040 ? 0.005 0.098 ? 0.017 0.067 ? 0.006 0.041 ? 0.004 0.056
? 0.004 0.052 ? 0.004 0.041 ? 0.003 0.055 ? 0.004 0.042 ? 0.002

Table 1: 1 standard deviation intervals for the mean squared errors of various algorithms when transferring from kin-8fm to kin-8nh. The values in bold are the smallest errors for each nta . Only Source KS has much worse performance than other algorithms so we do not show its result here. Directly training on the target data T ta (Only Target KS, Only Target KRR). Only training on the source data T so (Only Source KS, Only Source KRR). Training on the combined source and target data (Combined KS, Combined KRR). The CDM algorithm proposed by Wang and Schneider [2014] with KRR (CDM). The algorithm described in this paper with $G(a, b) = (a + ?)b$ where ? is a hyper-parameter (Scale KS, Scale KRR). ? The algorithm described in this paper with $G(a, b) = ?a + b$ where ? is a hyper-parameter (Offset KS, Offset KRR). ?

? ? ? ? ?

For the ?rst experiment, we vary the size of the target domain to study the effect of nta relative to nso . We use two datasets from the ?kin? family in Delve [Rasmussen et al., 1996]. The two datasets we use are ?kin-8fm? and ?kin-8nh?, both with 8 dimensional inputs. kin-8fm has fairly linear output, and low noise. kin-8nh on the other hand has non-linear output, and high noise. We consider the task of transfer learning from kin-8fm to kin-8nh. In this experiment, We set nso to 320, and vary nta in {10, 20, 40, 80, 160, 320}. Hyper-parameters were picked using grid search with 10-fold cross-validation on the target data

?? ?
 ?? ?
 ?
 ?
 ?

Figure 2: Box plots of experimental results on real datasets. Each box extends from the first to third quartile, and the horizontal lines in the middle are medians. For the robotics data, we report mean squared error (lower the better) and for the fMRI data, we report R-squared (the higher the better). For the ease of presentation, we only show results of algorithms with good performances.

	Only Target	KS Only Target	KRR Only Source	KS Only Source	KRR Combined	KS Combined	KRR CDM	Offset KS	Offset KRR	Scale KS	Scale KRR
Mean	-0.0096	0.1041	-0.4932	-0.8763	-0.7540	-0.5868	-3.1183	0.1190	0.1080	0.0017	0.0897
Median	0.0444	0.1186	-0.5366	-0.9363	-0.2023	-0.0691	-3.4510	0.1081	0.1221	-0.0321	0.1107
Standard Deviation	0.1041	0.2361	0.4555	0.6265	1.5109	1.3223	2.6473	0.0612	0.0682	0.0632	0.1104

Table 2: Mean, median, and standard deviation for the coefficient of determination (R-squared) of various algorithms on the fMRI dataset.

Table 2 shows the full table of results for the fMRI task. Using only the source data produces large negative R-squared, and while Only Target KRR does produce a positive mean R-squared, it comes with a high variance. On the other hand, both Offset methods have low variance, showing consistent performance. For this particular case, the Scale methods do not perform as well as the Offset methods, and as has been noted earlier, in general we would use cross validation to select an appropriate transfer function.

7

Conclusion and Future Works

In this paper, we proposed a general transfer learning framework for the HTL regression problem when there is some data available from the target domain. Theoretical analysis shows it is possible to achieve better statistical rate using transfer learning than standard supervised learning. Now we list two future directions and how our results could be further improved. First, in many real world applications, there is also a large amount of unlabeled data from the target domain available. Combining our proposed framework with previous works for this scenario [Cortes and Mohri, 2014, Huang et al., 2006] is a promising direction to pursue. Second, we only present upper bounds in this paper. It is an interesting direction to obtain lower bounds for HTL and other transfer learning scenarios.

8

Acknowledgements

S.S.D. and B.P. were supported by NSF grant IIS1563887 and ARPA-E Terra program. A.S. was supported by AFRL grant FA8750-17-2-0212. 9

2 References

- Shai Ben-David and Ruth Urner. Domain adaptation as learning with auxiliary information. In *New Directions in Transfer and Multi-Task-Workshop@ NIPS*, 2013.
- Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.
- John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129?136, 2008.
- Olivier Bousquet and Andr  Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499?526, 2002.
- Raymond J Carroll, David Ruppert, Leonard A Stefanski, and Ciprian M Crainiceanu. *Measurement error in nonlinear models: a modern perspective*. CRC press, 2006.
- Corinna Cortes and Mehryar Mohri. Domain adaptation in regression. In *Algorithmic Learning Theory*, pages 308?323. Springer, 2011.
- Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103?126, 2014.
- Corinna Cortes, Mehryar Mohri, and Andr s Mu oz Medina. Adaptation algorithm and theory based on generalized discrepancy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 169?178. ACM, 2015.
- Cecil C Craig. On the tchebychev inequality of bernstein. *The Annals of Mathematical Statistics*, 4(2):94?102, 1933.
- Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594?611, 2006.
- Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Sch lkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601?608, 2006.
- Samory Kpotufe and Vikas Garg. Adaptivity to local smoothness and dimension in kernel regression. In *Advances in Neural Information Processing Systems*, pages 3075?3083, 2013.
- Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *ICML (3)*, pages 942?950, 2013.
- Ilja Kuzborskij and Francesco Orabona. Fast rates by transferring from auxiliary hypotheses. *Machine Learning*, pages 1?25, 2016.
- Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From n to $n+1$: Multiclass transfer incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3358?3365, 2013.
- Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. Scalable greedy algorithms for transfer learning. *Computer Vision and Image Understanding*, 2016.
- Tongliang Liu, Dacheng Tao, Mingli Song, and Stephen Maybank. Algorithm-dependent generalization bounds for multi-task learning. *IEEE transactions on pattern analysis and machine intelligence*, 2016.
- Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.
- Mehryar Mohri and Andres Munoz Medina. New analysis and algorithm for learning with drifting distributions. In *Algorithmic Learning Theory*, pages 124?138. Springer, 2012.
- 10 Stephen Nuske, Kamal Gupta, Srinivasa Narasimhan, and Sanjiv Singh.

Modeling and calibrating visual yield estimates in vineyards. In *Field and Service Robotics*, pages 343–356. Springer, 2014. Francesco Orabona, Claudio Castellini, Barbara Caputo, Angelo Emanuele Fiorilla, and Giulio Sandini. Model adaptation with least-squares svm for adaptive hand prosthetics. In *Robotics and Automation*, 2009. ICRA’09. IEEE International Conference on, pages 2897–2903. IEEE, 2009. Carl Edward Rasmussen, Radford M Neal, Georey Hinton, Drew van Camp, Michael Revow, Zoubin Ghahramani, Rafal Kustra, and Rob Tibshirani. Delve data for evaluating learning in valid experiments. URL <http://www.cs.toronto.edu/delve>, 1996. Ingo Steinwart, Don R Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *COLT*, 2009. J Ridley Stroop. Studies of interference in serial verbal reactions. *Journal of experimental psychology*, 18(6):643, 1935. Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008. Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, pages 3081–3088. IEEE, 2010. Timothy D Verstynen. The organization and dynamics of corticostriatal pathways link the medial orbitofrontal cortex to future behavioral responses. *Journal of neurophysiology*, 112(10):2457–2469, 2014. Vladimir Vovk. Kernel ridge regression. In *Empirical Inference*, pages 105–116. Springer, 2013. Xuezhi Wang and Jeff Schneider. Flexible transfer learning under support and model shift. In *Advances in Neural Information Processing Systems*, pages 1898–1906, 2014. Xuezhi Wang and Jeff Schneider. Generalization bounds for transfer learning under model shift. 2015. Xuezhi Wang, Junier B Oliva, Jeff Schneider, and Barnab’s P’czos. Nonparametric risk and stability analysis for multi-task learning problems. In *25th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 1, page 2, 2016. Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006. Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 188–197. ACM, 2007. Yaoliang Yu and Csaba Szepesv’ri. Analysis of kernel mean matching under covariate shift. *arXiv preprint arXiv:1206.4650*, 2012. Kun Zhang, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 819–827, 2013. Yu Zhang. Multi-task learning and algorithmic stability. In *AAAI*, volume 2, pages 6–2, 2015. Ding-Xuan Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of computational and Applied Mathematics*, 220(1):456–463, 2008.