Nonparametric Risk and Stability Analysis for Multi-Task Learning Problems

Xuezhi Wang, Junier B. Oliva, Jeff Schneider, Barnabás Póczos
Carnegie Mellon University, Pittsburgh, PA, USA
{xuezhiw, joliva, schneide, bapoczos}@cs.cmu.edu

Abstract
Multi-task learning attempts to simultaneously leverage data from multiple domains in order to estimate related functions on each domain. For example, a special case of multi-task learning, transfer learning, is often employed when one has a good estimate of a function on a source domain, but is unable to estimate a related function well on a target domain using only target data. Multi-task/transfer learning problems are usually solved by imposing some kind of “smooth” relationship among/between tasks. In this paper, we study how different smoothness assumptions on task relations affect the upper bounds of algorithms proposed for these problems under different settings. For general multi-task learning, we study a family of algorithms which utilize a reweighting matrix on task weights to capture the smooth relationship among tasks, which has many instantiations in existing literature. Furthermore, for multi-task learning in a transfer learning framework, we study the recently proposed algorithms for the “model shift”, where the conditional distribution \( P(Y|X) \) is allowed to change across tasks but the change is assumed to be smooth. In addition, we illustrate our results with experiments on both simulated and real data.

1 Introduction
As machine learning is applied to a growing number of domains, many researchers have looked to exploit similarities in machine learning tasks in order to increase performance. For example, one may suspect that data for the classification of one commodity as profitable or not may help in classifying a different commodity. Similarly, it is likely that data for spam classification in one language can help spam classification in another language. A common technique for leveraging data from different domains for machine learning tasks is multi-task learning. Multi-task learning pools multiple domains together and couples the learning of several tasks by regularizing separate estimators jointly and dependently. For instance, in transfer learning, a special case of multi-task learning, one uses data (or an estimator) from a well understood source domain with plentiful data to aid the learning of a target domain with scarce data. Although multi-task learning algorithms are becoming prevalent in machine learning, there are gaps in our understanding of their properties, especially in nonparametric settings. This paper looks to increase our understanding of fundamental questions such as: What can one say about the true risk of a multi-task estimator given its empirical risk? How do relative sample sizes affect learning among different domains? How does the similarity between two functions affect one’s ability to transfer learning between them?

![Figure 1: Toy example illustrating general multi-task learning (left) and transfer learning (right).](image-url)
over typical one data-set learning. In this paper, we provide analysis that connects the smoothness of the offset function to the learning bounds for this kind of transfer. We obtain tighter learning bounds for transfer learning when we assume a smooth change across domains, given that the data from the source domain is sufficiently large.

**Contribution** We provide a stability analysis for multi-task learning that allows one to understand the gap between the true risk and the empirical risk for many popular estimators. Also, we analyze the risk of multi-task learning under a nonparametric function transfer learning framework. In our analysis we derive an upperbound for the $L_2$ risk that elucidates previously unknown question such as the relationship between sample sizes and loss, as well as conditions for outperforming one data-set estimation with transfer learning.

## 2 Related Work

A large part of multi-task learning work is formulated using kernel ridge regression (KRR) with various regularizations on task relations. The $L_2$ penalty is used on a shared mean function and on the variations specific to each task [Evgeniou and Pontil, 2004; Evgeniou et al., 2005]. In [Solomom et al., 2013] a pairwise $L_2$ penalty is placed between each pair of tasks. In [Zhou et al., 2011] an $L_2$ penalty is proposed on each pair of consecutive tasks that controls the temporal smoothness. By regularizing the shared clustering structure among task parameters, task clusters are constructed for different features [Zhong and Kwok, 2012]. A multi-linear multi-task learning algorithm is proposed in [Romera-Paredes et al., 2013] by placing a trace norm penalty. However, there are very few literature dealing with the stability of multi-task KRR algorithms. The stability bounds for transductive regression algorithms are analyzed in [Cortes et al., 2008]. In [Audiffren and Kadri, 2013], the authors study the stability properties of multi-task KRR by considering each single task separately, thus failing to reveal the advantage of regularizing task relations. In [Maurer and Pontil, 2013], a regularized trace-norm multi-task learning algorithm is studied where the task relations are modeled implicitly, while we study a general family of multi-task learning algorithms where the task relations are modeled explicitly, reflected by a reweighting matrix $\Lambda$ on task weights. More recently, the algorithmic stability for multi-task algorithms with linear models $f(x) = w^T x$ is studied in [Zhang, 2015]. While in this paper we consider the more challenging nonlinear models with feature map $\phi(x)$, where the new multi-task kernel between $x_i, x_j$ is defined as $\phi(x_i) \Lambda^{-1} \phi^T(x_j)$ by absorbing the reweighting matrix $\Lambda$ on task weights. Our theory is also developed on the more general nonlinear models.

Most work on transfer learning assumes that specific parts of the model can be carried over between tasks. Recent work on covariate shift [Shimodaiera, 2000; Huang et al., 2007; Gretton et al., 2007; Yu and Szepesvri, 2012; Wen et al., 2014; Reddi et al., 2015] considers the case where only $P(X)$ differs across domains, while $P(Y|X)$ stays the same (here $X$ denotes the input feature space and $Y$ denotes the output label space). In [Zhang et al., 2013], target and conditional shift are modeled by matching the marginal distributions on $X$. For transfer learning under model shift, there could be a difference in $P(Y|X)$ that can not simply be captured by the differences in distribution $P(X)$, hence neither covariate shift or target/conditional shift will work well under the model shift assumption. This problem is also demonstrated in [Wang et al., 2014]. In the same paper, the authors propose a transfer learning algorithm to handle the general case where $P(Y|X)$ changes smoothly across domains.

We focus our analysis to the nonparametric setting. In particular, we consider orthogonal series regression, where one attempts to model functions using a finite collection of orthonormal basis functions [Tsymbakov, 2009; Wasserman, 2006]. Moreover, we also consider kernel ridge regression, a natural generalization of ridge regression [Hoel and Kendall, 1970] to the nonparametric setting [Györfi, 2002].

## 3 Stability Analysis on Multi-Task Kernel Ridge Regression

In this section, we analyze the stability bounds for multi-task kernel ridge regression (MT-KRR). Our analysis shows that, MT-KRR achieves tighter stability bounds than independent task learning by regularizing task relations. In addition, different regularization techniques yield different stability bounds that are closely related to the diagonal blocks of the inverted reweighting matrix. Due to space constraints please refer to the appendix for all the proofs.

### 3.1 Multi-task KRR Algorithm: Formulation and Objective

Assume we have $T$ tasks, each task $t$ has data matrix $X_t \in \mathcal{R}^{n_t \times d}, Y_t \in \mathcal{R}^{n_t}$, where $x_{t,i} \in \mathcal{X}$ is the $i$-th row of $X_t$, and $y_{t,i} \in \mathcal{Y}$ is the $i$-th scalar of $Y_t$. $n_t$ is the number of data points for each task, and $d$ is the dimension of features. Denote the total number of data points as $m = \sum_{t=1}^{T} n_t$.

Let $\Phi$ be the feature mapping on $x$ associated to kernel $k$ with dimension $q$, and $\Phi(X_t)$ denote the matrix in $\mathcal{R}^{n_t \times q}$ whose rows are the vectors $\phi(x_{t,i})$. Let $\Phi(X) \in \mathcal{R}^{m \times q}$ represent the diagonalized data matrix $\Phi(X) = \text{diag}(\Phi(X_1), \Phi(X_2), \ldots, \Phi(X_T))$ for all tasks, $Y \in \mathcal{R}^{m \times 1}$ be the stacked label vector $Y = [Y_1 \ Y_2 \ldots \ Y_T]^T$, and $w \in \mathcal{R}^{q \times 1}$ be the stacked weight vector $w = [w_1 \ w_2 \ldots \ w_T]^T$.

Throughout the paper we use $\ell_2$ loss as the loss function for a hypothesis $h$, i.e., $l(h(x), y) = (h(x) - y)^2$. Note that $l(h(x), y)$ is a $\sigma$-admissible loss function, i.e., $\forall x,y, \forall h, h', |l(h(x), y) - l(h'(x), y)| \leq \sigma |h(x) - h'(x)|$. For $\ell_2$ loss $\sigma = 4B$, assuming $|h(x)| \leq \tilde{B}, |y| \leq B$ for some $B > 0$. Define the MT-KRR objective as:

$$
\min_w \frac{1}{m} \|Y - \Phi(X)w\|_F^2 + w^T \Lambda w,
$$

where $\Lambda$ is a $Tq \times Tq$ reweighting matrix on task weights $w$. Let $\phi(x_{t,i}) = [0 \ldots 0 \phi(x_{t,i}) \ldots 0]$ be a row of $\Phi(X)$ for task $t$. Let $H$ be a reproducing kernel Hilbert space with kernel $k_{\Lambda^{-1}}(x_{s,i}, x_{t,i}) = \phi(x_{s,i})\Lambda^{-1}\phi^T(x_{t,i})$ ($s,t$ are

---

1 Available at [http://www.autonlab.org/autonweb/24058.html](http://www.autonlab.org/autonweb/24058.html)
Theorem 3.3. For the MT-KRR algorithm, we have the following theorem.

\[
\min_{y \in \mathcal{H}} \frac{1}{n} \sum_{t=1}^{T} \sum_{i=1}^{n_t} (y_{t,i} - g(x_{t,i}))^2 + \|g\|^2_{k_{\lambda^{-1}}},
\]

where \( g(x) = (g(x_{k_{\lambda^{-1}}}(x)), y_{\ell}) \), and \( \|\cdot\|_{k_{\lambda^{-1}}} \) is the norm in \( \mathcal{H} \). This generalizes to the case where \( q = \infty \).

The solution to MT-KRR is (assuming nonsingular \( \Lambda \)): \( w = \Lambda^{-1} \Phi^T(X)\Phi(X)\Lambda^{-1} + mI)^{-1}Y \). Note in multi-task learning setting, we have \( \Lambda = \Omega \otimes I_q \) for some \( \Omega \in \mathbb{R}^{T \times T} \), where \( I_q \) is the \( q \times q \) identity matrix and \( \otimes \) is the Kronecker product. By the property of the inverse of a Kronecker product, \( \Lambda^{-1} = M \otimes I_q \) where \( M = \Omega^{-1} \), and it can be easily shown that \( k_{\Lambda^{-1}}(x_{a,i}, x_{t,j}) = M_{a,i}k(x_{a,i}, x_{t,j}) \).

Most existing multi-task algorithms can be cast into the above framework, see Table 1 for a few examples.

Remark. Eq. 1 assumes same weight \( 1/m \) on the loss for \( (x_{t,i}, y_{t,i}) \) for all tasks. Alternatively, we can put different weights on the loss for different tasks, i.e., \( \min_{\omega} \sum_{t=1}^{T} \sum_{i=1}^{n_t} (\phi(x_{t,i}) - y_{t,i})^2 + w^t \Lambda w \). The solution becomes \( w = \Lambda^{-1} \Phi^T(X)\Phi(X)\Lambda^{-1}(\Phi(X) + C^{-1}I)^{-1}Y \), where \( C \) is the -reweight matrix with \( 1/m \) as the diagonal elements. As \( C \) is the same under different \( \Lambda \)’s, it is not the focus of this paper. A study on the effect of \( C \) can be found in [Cortes et al., 2008].

3.2 Uniform Stability for MT-KRR

We study the uniform stability [Bousquet and Elisseeff, 2002], which is usually used to bound true risk in terms of empirical risk, for the MT-KRR algorithm.

Definition 3.1. ([Bousquet and Elisseeff, 2002]). The uniform stability \( \beta \) for an algorithm \( A \) w.r.t. the loss function \( l \) is defined as: \( \forall S \subseteq \mathbb{Z}^m, \forall i \in \{1, ..., m\}, \|l(A_{S \setminus i}) - l(A_S)\|_{\infty} \leq \beta \), where \( Z = \mathcal{X} \times \mathcal{Y} \) drawn i.i.d from an unknown distribution \( D \), and \( S \setminus i \) is formed by removing the \( i \)-th element from \( S \).

Definition 3.2. (Uniform stability w.r.t a task \( t \)). Let \( i \) be a data index for task \( t \). The uniform stability \( \beta_t \) of a learning algorithm \( A \) w.r.t a task \( t \), w.r.t. loss \( l \) is: \( \forall S \subseteq \mathbb{Z}^m, \forall i \in \{1, ..., n_t\}, \|l(A_{S \setminus i}) - l(A_S)\|_{\infty} \leq \beta_t \).

Let the random or generalization error be defined as \( R(A, S) = \mathbb{E}_z[l(A_S, z)], z \in Z \), and the empirical error be defined as \( R_{emp} = \frac{1}{m} \sum_{z_i \in Z} l(A_{S}, z_i), z_i \in Z^m \). Then we have the following generalization error bound (Theorem 12, [Bousquet and Elisseeff, 2002]) with probability at least \( 1 - \delta \): \( R \leq R_{emp} + 2\beta + (4m\beta + 4B^2)\sqrt{\frac{\ln 1/\delta}{2m}} \). This theorem gives tight bounds when the stability \( \beta \) scales as \( 1/m \). For the MT-KRR algorithm, we have the following theorem hold with respect to the uniform stability:

Theorem 3.3. Denote \( \Lambda^{-1} = M \otimes I_q \), and \( M_1, ..., M_T \) are the diagonal elements of \( M \). Assuming the kernel values are bounded: \( \forall x \in \mathcal{X}, k(x, x) \leq \kappa^2 < \infty \). The learning algorithm defined by the minimizer of Eq. 1 has uniform stability \( \beta \) w.r.t. \( \sigma \)-admissible loss \( l \) with:

\[
\beta \leq \frac{\sigma^2\kappa^2}{2m} \max_t M_t.
\]

The proof is similar to the proof of Thm. 22 in [Bousquet and Elisseeff, 2002], except that in the multi-task learning setting, for the \( t \)-th task \( \phi(x_{t,i}) = [0 \cdots 0 \phi(x_{t,i}) 0 \cdots 0] \), by the standard bounds for Rayleigh quotient, we have

\[
\phi(x_{t,i})^T \Lambda^{-1} \phi(x_{t,i}) \leq \kappa^2 \max_t (M_t I_q) = \kappa^2 M_t.
\]
Combining Lemma 3.5 and Lemma 3.4 we can see that, with temporal penalty we have tightest stability bounds $\beta_t$ for $t = \text{mid}$. Also, we achieve tighter stability bounds $\beta_t$ for the $t$th task than the $t-1$th task, if $t < \text{mid}$; and tighter $\beta_t$ for the $t$th task than the $t + 1$ task, if $t > \text{mid}$. However, since $M_{\text{mid}} \geq (\lambda_p + \lambda_s)/\lambda_s$, we achieve a looser bound with temporal penalty than Eq. 3 or Eq. 4. It indicates that we might lose some algorithmic stability due to the relatively restricted temporal smoothness assumption, compared to assuming pairwise smoothness. Nonetheless, the stability bound with temporal penalty is tighter than learning each task independently: $\max_t M_t < 1/\lambda_s$, for $T \geq 2$.

4 Upper Bounds on Transfer Learning

A special case of multi-task learning, transfer learning, also assumes that one can benefit from task relations, but focuses mainly on two tasks. While general multi-task learning assumes a comparable number of samples for each task, transfer learning usually assumes a sufficiently labeled source task and a very limited labeled target task. In this section, we analyze the $L_2$ risk for transfer learning with respect to the source and target sample size, and smoothness assumptions made between the tasks.

4.1 Model

We consider a densely sampled function $f_0$, which one uses to aid in the regression of a sparsely sampled function $f_1$. The relationship between functions is defined through a smoothness assumption on the difference of the two functions: $g(x) \equiv f_1(x) - f_0(x)$.

Our estimator works as follows: first, we use a sample of noisy $f_0$ values to produce an estimate $\hat{f}_0$; second, we use $\hat{f}_0$ to generate noisy samples of $g$ by subtracting $\hat{f}_0$ from noisy samples of $f_1$, and we produce an estimate $\hat{g}$; lastly, we define our estimator of $f_1$ as $\hat{f}_1(x) \equiv \hat{f}_0(x) + \hat{g}(x)$. Specifically we consider the following data:

\begin{align*}
&\{u_{i1}^{n_0}\}^{n_0}_{i=1}, \{u_{i1}^{n_1}\}^{n_1}_{i=1} \sim \text{Unif}([0, 1]^d), \quad (5) \\
&Y_0 \equiv \{y_{i0} = f_0(u_{i0}) + e_{i0}\}^{n_0}_{i=1}, \quad (6) \\
&Y_1 \equiv \{y_{i1} = f_1(u_{i1}) + e_{i1}\}^{n_1}_{i=1}, \quad (7) \\
&\epsilon_{ij} \sim \Xi, \quad \mathbb{E}[\epsilon_{ij}] = 0, \quad \mathbb{V}[\epsilon_{ij}] \leq \sigma^2 < \infty. \quad (8)
\end{align*}

Note, $\Xi$ is an error distribution with moment constraints. Furthermore, we shall take $n_1 = O(n_0)$, although this is not necessary for the bounds derived below.

4.2 Basis Functions and Projections

We describe the estimation of functions using orthonormal basis functions. Let $\{\varphi_i\}_{i \in \mathbb{Z}}$ be an orthonormal basis for $L_2([0, 1])$, where $L_2(\Omega) = \{f : \Omega \mapsto \mathbb{R} : \int_{\Omega} f^2 < \infty\}$. Then, the tensor product of $\{\varphi_i\}_{i \in \mathbb{Z}}$ serves as an orthonormal basis for $L_2([0, 1]^d)$; that is, the following is an orthonormal basis for $L_2([0, 1]^d)$: $\{\varphi_{\alpha}\}_{\alpha \in \mathbb{Z}^d}$ where $\varphi_{\alpha}(x) = \prod_{i=1}^d \varphi_{\alpha_i}(x_i), \ x \in [0, 1]^d$. So we have that $\forall \alpha, \zeta \in \mathbb{Z}^d, \{\varphi_\alpha, \varphi_\zeta\} = I\{\alpha = \zeta\}$. Let $f \in L_2([0, 1]^d)$, then $f(x) = \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f) \varphi_\alpha(x)$ where $a_\alpha(f) = \langle \varphi_\alpha, f \rangle = \int_{[0, 1]^d} \varphi_\alpha(z) f(z) dz \in \mathbb{R}$.

Suppose function $f$ has a corresponding set of evaluation points $Y = \{y_j = f(u_j) + \epsilon_j\}_{j=1}^n$ where $u_j \sim \text{Unif}([0, 1]^d)$ and $\mathbb{E}[\epsilon_j] = 0, \mathbb{E}[\epsilon_j^2] < \infty$. Then, $\hat{f}$, the estimate of $f$, will be:

$$
\hat{f}(x) = \sum_{\alpha \in M} a_{\alpha}(Y) \varphi_\alpha(x) \quad \text{where} \quad a_{\alpha}(Y) = \frac{1}{n} \sum_{j=1}^n y_j \varphi_\alpha(u_j),
$$

and $M$ is a finite set of indices for basis functions.

4.3 Theory

We bound the $L_2$ risk of a transfer learning based estimate of $f_1$: $\mathbb{E}\left[\|f_1 - \hat{f}_1\|^2\right]$. First, we state our assumptions on functions $f_0$, and $f_1$:

(a) Sobolev Ellipsoid Function Class Assumptions. We shall make a Sobolev ellipsoid function class assumption for $f_0, f_1 \in \mathcal{F}$. Let $a(f) \equiv \{a_\alpha(f)\}_{\alpha \in \mathbb{Z}^d}$. Suppose that $\mathcal{F}_{\gamma,A} = \{f : a(f) \in \Theta_{\gamma,A}, \|f\|_{\infty} \leq f_{\text{max}}\}$, where $\Theta_{\gamma,A} = \{\{\theta_{\alpha}\}_{\alpha \in \mathbb{Z}^d} : \sum_{\alpha \in \mathbb{Z}^d} \theta_{\alpha}^2 \kappa_2(\alpha) < A^2\}$, and $\kappa_2(\alpha) = \sum_{i=1}^d |\alpha_i|^2 \gamma_i$ for $\gamma \in \mathbb{R}_{++} \cup \mathbb{R}_{++} \cup \mathbb{R}_{++} = (0, \infty)$. This assumption will control the tail-behavior of projection coefficients and allow one to effectively estimate $f \in \mathcal{F}$ using a finite number of projection coefficients on the empirical functional observation.

(b) Smooth Difference Assumption. We shall make an additional assumption on the difference between $f_1$ and $f_0$, $g(x) \equiv f_1(x) - f_0(x)$: $g = f_1 - f_0 \in \mathcal{F}_{p,B}$. Namely, we are imposing a smoothness constraint on the difference between our functions $f_0$ and $f_1$, which we will show controls the effectiveness of transfer learning.
Estimator: Before writing our estimator for \( f_1 \), we define some terms. First, let \( \hat{f}_0 \) be the standard estimator for \( f_0 \) based on \( Y_0 \), let \( M_t(\alpha) \equiv \{ \alpha \in \mathbb{Z}^d : \kappa_\alpha(\alpha) \leq t \} \):

\[
\hat{f}_0(x) = \sum_{\alpha \in M_t(\alpha)} a_\alpha(Y_0) \varphi_\alpha(x) \quad \text{where} \quad (10)
\]

\[
a_\alpha(Y_0) = \frac{1}{n} \sum_{j=1}^{n} y_{0j} \varphi_\alpha(u_{0j}) . \quad (11)
\]

We will take \( \hat{g} \) to be the estimate of \( g \) based on \( Z \), where

\[
Z \equiv \{ z_j = y_{1j} - \hat{f}_0(u_{1j}) \}_{j=1}^{n_1} , \quad (12)
\]

\[
z_j = f_1(u_{1j}) - \hat{f}_0(u_{1j}) + \epsilon_{1j} = g(u_{1j}) + r(u_{1j}) + \epsilon_{1j}, \quad (13)
\]

and \( g(x) = f_1(x) - f_0(x) , r(x) = f_0(x) - \hat{f}_0(x) \). Our estimator for \( f_1 \) will then be: \( \hat{f}_1(x) = \hat{f}_0(x) + \hat{g}(x) \), where \( \hat{g} \) is the estimate of \( g \) based on \( Z, \hat{g}(x) = \sum_{\alpha \in M_t(\alpha)} a_\alpha(Z) \varphi_\alpha(x) \).

Risk Analysis: We analyze the \( L_2 \) risk of our estimator below. Note that:

\[
\mathbb{E} \left[ \| f_1 - \hat{f}_1 \|_2 \right] = \mathbb{E} \left[ \| f_0 + g - (\hat{f}_0 + \hat{g}) \|_2 \right] \leq \sqrt{\mathbb{E} \left[ \| f_0 - \hat{f}_0 \|_2^2 \right] + \mathbb{E} \left[ \| g - \hat{g} \|_2^2 \right]}, \quad \text{thus we first upper-bound the risk for typical function estimation} \quad \mathbb{E} \left[ \| f_0 - \hat{f}_0 \|_2^2 \right]
\]

then that for the smooth transfer \( \mathbb{E} \left[ \| g - \hat{g} \|_2^2 \right] \). First we analyze the risk of standard functions regression one a single data-set for the estimation of the source function.

Lemma 4.1. Let \( f_0 \in \mathcal{F}_{\alpha,a} \), then \( \mathbb{E} \left[ \| f_0 - \hat{f}_0 \|_2^2 \right] = O \left( n_0^{-\frac{2}{2+\rho^{-1}}} \right) \), where \( \gamma^{-1} = \sum_{i=1}^{d} \gamma_i^{-1} \).

Next we analyze the risk of estimating \( g \) from \( Z \) (13). Note that \( Z \) is not a set of noisy observations from \( g \) as \( Y_0 \) is to \( f_0 \); we, instead have biased observations (from using \( \hat{f}_0 \)), thus the rate will vary a bit.

Lemma 4.2. Let \( g \in \mathcal{F}_{\rho,B} \), then \( \mathbb{E} \left[ \| g - \hat{g} \|_2^2 \right] = O \left( n_1^{-\frac{2}{2+\rho^{-1}}} + \frac{n_1}{n_0} \frac{1}{2+\rho^{-1}} \right) \).

One can see that we pay a penalty of \((1 + n_1/n_0)^{2(2+\rho^{-1})}\) for using a biased sample to approximate \( g \). As one would expect the penalty diminishes as \( n_0 \rightarrow \infty \). Note furthermore that if \( n_0 \geq n_1 \) then this penalty is no more than \( 2^{2+\rho^{-1}} = O(1) \). Hence, the risk of \( \hat{g} \) is asymptotically upper-bounded with the same rate as that of the unbiased sample estimator \( \hat{g} \).

Transfer Estimator Risk: Below we state this section’s main theorem and discuss some insights gained from it.

Theorem 4.3. Let \( f_1 \in \mathcal{F} \) and \( \hat{f}_1(x) \equiv \hat{f}_0(x) + \hat{g}(x) \), then: \( \mathbb{E} \left[ \| f_1 - \hat{f}_1 \|_2^2 \right] = O \left( n_0^{-1/2+\gamma^{-1}} + n_1^{-1/2+\rho^{-1}} + \frac{n_1}{n_0} \frac{1}{1/2+\rho^{-1}} \right) \).

For simplification, consider the case where smoothness parameters are \( \gamma = (\tau, \ldots, \tau)^t \) and \( \rho = (\nu, \ldots, \nu)^t \), and \( n_0 = n_1^\gamma \) for \( \lambda \geq 1 \). One then has that: \( \mathbb{E} \left[ \| f_1 - \hat{f}_1 \|_2^2 \right] = O \left( n_1^{-\frac{2}{2+\gamma^{-1}}} + n_1^{-\frac{2}{2+\rho^{-1}}} + \frac{n_1}{n_0} \frac{1}{1/2+\rho^{-1}} \right) \).

In other words, transfer learning has an asymptotic risk of regressing the smooth difference function \( g \) with the target sample of size \( n_1 \): \( \mathbb{E} \left[ \| f_1 - \hat{f}_1 \|_2^2 \right] = O \left( \mathbb{E} \left[ \| g - \hat{g} \|_2^2 \right] \right) \), where \( \hat{g} \) is defined analogously to (9) with a sample size of \( n_1 \). Since functions \( f_1 \) and \( f_0 \) are similar the asymptotic reduction to the rate of estimation for \( g \) proves very beneficial.

Remark. We see that the upper bounds derived for MT-KRR and transfer learning are both affected by the smoothness assumptions we make between/among tasks (the \( \gamma, \rho \) parameter in the above analysis, and the \( \lambda_p, \lambda_s \) parameter in Sec. 3.3).

5 Experiments

5.1 Synthetic Data

Multi-Task Learning Stability. To show the stability bounds under different penalties, we simulate data with \( T \) tasks. Each task \( t \) has \( \{ X_t, Y_t \} : Y_t = f_c + f_o + 0.1e, \) where \( f_c = \sin(20x) + \sin(10x) \) is the central function, and \( f_o = \sin(5(1 + t_i)x) \) is a smoother additive function, with \( t_i \sim \text{Unif}(0,1) \), plus \( e \in \mathcal{N}(0,1) \). Fig.2 (left) shows an example of the data with \( T = 3 \) and \( n_1 = 20 \) per task.

In Fig.2, we also plot the risk difference \( R - R_{\text{emp}} \) (Sec. 3.2) w.r.t different number of tasks (fixed 10 points per task), and different number of points per task (with fixed 5 tasks), averaged over 50 experiments. We also plot the theoretical bounds fitted to the actual curve using regression) for each case. We see that the results are consistent with our analysis. Using central+offset (Eq. 3), pairwise-penalty (Eq. 4), or temporal-penalty (Lemma 3.5) we achieve tighter bounds than learning each task independently (denoted as Separate). In addition, central+offset and pairwise-penalty result in the same curve (red and blue) when we set \( \lambda_p/T \) in central+offset equal to \( \lambda_p \) in pairwise-penalty, which shows the equivalence of these two methods. Further we observe that temporal-penalty gives slightly larger \( R - R_{\text{emp}} \) than central+offset and pairwise-penalty, which coincides with our analysis.

Transfer Learning Risk. We illustrate the risk of function transfer learning through an experiment with synthetic data. We randomly generate \( f_0, g, \) and \( f_1 \) and we draw data-sets \( Y_0, Y_1 \) with various configurations of \( n_1 \) and \( n_0 \). We consider the cosine basis: \( \varphi_0(x) = 1, \varphi_k(x) = \sqrt{2} \cos(\pi kx), \forall k \geq 1 \).
5.2 Real Data

The real dataset is the Air Quality Index (AQI) dataset [Mei et al., 2014]. We extract bag-of-words vectors (feature $X$ with dimension $d = 100$, 395) from social media posts to predict the AQI (label $Y$) across cities. The results are averaged over 20 experiments. In Fig. 4 (left), we show the prediction error of MT-KRR using pairwise penalty (or equivalently the central+offset penalty) with 4 cities as 4 different tasks. We see that the MT-KRR algorithm (mtl) outperforms independent-task-learning (ind). In addition, we plot the leave-one-out error for each task (loo-1 through 4), and the prediction error by MT-KRR for the best task (mtl-min), which outperforms learning that task by itself (loo-3). Fig. 4 (right) shows the prediction error using the transfer method analyzed in this paper, compared with state-of-the-art baselines. The transfer method benefits from modeling a smoother offset across domains compared to optDA [Chattopadhyay et al., 2011] with single-source, and it also outperforms KMM [Huang et al., 2007] by allowing changes in $P(Y|X)$.

6 Conclusion

In this paper we provide theory that connects the risk bounds for both transfer and multi-task learning to the relation of tasks. We show that, by imposing a smooth relationship between/among tasks, we obtain favorable learning rates for both algorithms, compared to learning tasks independently.

Acknowledgments

This work is funded in part by DARPA grants FA87501220324, FA87501420244, NSF IIS-1247658 and DOE DE-SC0011114 grants.
References


