Submitted to the Annals of Statistics
arXiv: 0000.0000

TRANSFER LEARNING FOR NONPARAMETRIC REGRESSION: NON-ASYMPTOTIC MINIMAX ANALYSIS AND ADAPTIVE PROCEDURE[*]

By T. Tony Cai and Hongming Pu

University of Pennsylvania

Transfer learning for nonparametric regression is considered. We first study the non-asymptotic minimax risk for this problem and develop a novel estimator called the confidence thresholding estimator, which is shown to achieve the minimax optimal risk up to a logarithmic factor. Our results demonstrate two unique phenomena in transfer learning: auto-smoothing and super-acceleration, which differentiate it from nonparametric regression in a traditional setting. We then propose a data-driven algorithm that adaptively achieves the minimax risk up to a logarithmic factor across a wide range of parameter spaces. Simulation studies are conducted to evaluate the numerical performance of the adaptive transfer learning algorithm, and a real-world example is provided to demonstrate the benefits of the proposed method.

1. Introduction. Transfer learning, a technique that utilizes knowledge gained from related source domains to improve performance in a target domain, has gained widespread popularity in machine learning due to its successes across a range of applications, including natural language processing (Daumé III, 2009), computer vision (Tzeng et al., 2017), and epidemiology (Apostolopoulos and Mpesiana, 2020). See the recent survey papers on transfer learning (Weiss et al., 2016; Zhuang et al., 2020) for more examples and in-depth discussions.

Transfer learning has received significant recent attention in statistics, due to its empirical successes. It has been studied in a decision-theoretical framework for a variety of supervised learning problems, such as classification (Cai and Wei, 2021; Reeve et al., 2021), high-dimensional linear regression (Li et al., 2022a), and generalized linear models (Li et al., 2021), as well as unsupervised learning problems, such as Gaussian graphical models (Li et al., 2022b). Minimax optimal rates of convergence have been established

[*] The research was supported in part by NSF Grant DMS-2015259 and NIH grant R01-GM129781.

MSC 2010 subject classifications: Primary 62G08; secondary 62L12

Keywords and phrases: Adaptivity, nonparametric regression, optimal rate of convergence, transfer learning

and data-driven adaptive algorithms have been developed.

In this paper we consider transfer learning for nonparametric regression. Formally, in the target domain one observes independent and identically distributed (i.i.d.) samples $(X_i, Y_i) \overset{iid}{\sim} Q$, $i = 1, \ldots, n_Q$, with

$$Y_i = f(X_i) + z_i,$$

where $f(\cdot)$ is an unknown function of interest and $\{z_1, \ldots, z_{n_Q}\}$ are random noises satisfying $\mathbb{E}(z_i|X_i) = 0$. Different from the conventional setting, in transfer learning one also has auxiliary data from the source domains. For ease of presentation, we first focus on the case of a single source domain and discuss the case of multiple source domains later. In the single source domain setting, in addition to the samples from the target domain, one observes i.i.d. samples $(X'_i, Y'_i) \overset{iid}{\sim} P$, $i = 1, \ldots, n_P$, from the source domain with

$$Y'_i = g(X'_i) + z'_i,$$

where $\{z'_i\}$ are i.i.d. random noise satisfying $\mathbb{E}(z'_i|X'_i) = 0$ and g is an unknown function.

In the context of transfer learning for nonparametric regression, the joint distribution P of (X'_i, Y'_i) from the source domain and the joint distribution Q of (X_i, Y_i) from the target domain are different but related. Two popular settings that have been considered in the literature are covariate shift and posterior drift. In the case of covariate shift, the conditional mean functions from the target and source domains, f and g, are the same, but the marginal distributions of the covariates, X_i and X'_i, are different (Shimodaira, 2000; Huang et al., 2006; Wen et al., 2014). On the other hand, the posterior drift model assumes that the mean functions, f and g, may be different, but the marginal distributions of the covariates, X_i and X'_i, are the same. The posterior drift model is a general framework that can be applied to many practical problems, including robotics control (Vijayakumar et al., 2002; Nguyen-Tuong et al., 2008a,b; Yeung and Zhang, 2009; Cao et al., 2010) and air quality prediction (Mei et al., 2014; Wang et al., 2016).

In this paper, we focus on the posterior drift model, where we assume that the difference between the mean functions, f and g, can be well approximated by a polynomial function of a given order in L_1 distance. This distance is referred to as the *bias strength*, denoted by ϵ. It controls the similarity between f and g up to a polynomial of a given order. The smaller the bias strength, the more similar f and g are, and vice versa. When the bias strength is zero, f and g differ by a polynomial. In this special case, transfer learning can be highly beneficial. We refer to g as a "perfect reference" of f when the bias strength is zero, and an "imperfect reference" otherwise.

There are two natural and important goals for transfer learning: to accurately quantify the contribution of observations from the source domain to the regression task in the target domain, and to develop an optimal transfer learning algorithm. The answer to the first objective depends on various factors, including sample sizes n_P and n_Q, the smoothness of f and g, and the bias strength ϵ. In this paper, we investigate the *non-asymptotic* minimax risk of this problem and propose a data-driven, adaptive transfer learning algorithm to achieve optimal results.

1.1. *Main Results and Our Contribution.* We first establish the minimax optimal rate of convergence for transfer learning for nonparametric regression in the posterior drift setting. Suppose f is β_Q-smooth and g is β_P-smooth (which will be defined precisely later). Let F denote the set of distribution pairs (Q, P) defined in (1) in Section 2. It is shown that the minimax risk of transfer learning satisfies

$$C_L \cdot \left(n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} + (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{1}{n_Q} \right) \leq \inf_{\hat{f}} \sup_{(Q,P)\in F} \mathbb{E}||\hat{f} - f||_2^2$$

$$\leq C_U \cdot \left(n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} \ln^4(n_{\max}) + \ln^8(n_Q) \cdot (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{\ln^4(n_Q)}{n_Q} \right),$$

for some positive constants C_L and C_U not depending on ϵ, n_P or n_Q, where $n_{\max} = \max(n_P, n_Q)$ and $\beta_{\max} = \max(\beta_Q, \beta_P)$. In the special case of observing the data from the target domain only, i.e., $n_P = 0$, then the minimax risk for estimating f is of order $n_Q^{-\frac{2\beta_Q}{2\beta_Q+d}}$. Comparing it with the transfer learning risk, we can conclude when and how much transfer learning is helpful for the target task. The necessary and sufficient condition for transfer learning to improve the estimation performance is that the bias strength is smaller than $n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$ and either the mean function from the source domain is smoother or the sample size of the source domain is larger. If we fix f and g and thus the bias strength ϵ and let n_Q go to infinity then this condition fails when n_Q is sufficiently large unless $\epsilon = 0$. This means in order to make transfer learning work *asymptotically*, g has to be a perfect reference. However, with any fixed and finite sample size, the non-asymptotic analysis above shows that transfer learning can help with an imperfect reference function g as long as the bias strength ϵ is smaller than the phase transition threshold $n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$.

There are some interesting phenomena from the minimax analysis in the case where bias strength ϵ is sufficiently small.

1. If the function g is rougher than f, i.e., $\beta_P < \beta_Q$, then the minimax risk does not depend on β_P and is the same as the minimax risk when $\beta_P = \beta_Q$. This means that transfer learning can be effective even if g is less smooth than f.
2. If estimation of f and g is considered separately using the data from the target domain and from the source domain alone, the usual minimax risks for estimating f and g are proportional to $n_Q^{-\frac{2\beta_Q}{2\beta_Q+d}}$ and $n_P^{-\frac{2\beta_P}{2\beta_P+d}}$ respectively. Then if $\beta_P < \beta_Q$ and $n_P \gg n_Q$ or $\beta_P > \beta_Q$ and $n_P \ll n_Q$, the minimax risk for the transfer learning can be much smaller than either of these two minimax risks, provided ϵ is sufficiently small. This phenomenon sheds new light on the understanding of transfer learning in that even the task from the source domain is harder, i.e., $n_Q^{-\frac{2\beta_Q}{2\beta_Q+d}} \ll n_P^{-\frac{2\beta_P}{2\beta_P+d}}$, it may still help the task in the target domain.

A novel transfer learning algorithm is developed and shown to attain the minimax optimal risk, possibly up to a logarithmic factor. However, the algorithm relies on knowledge of the smoothness parameters β_Q and β_P. To address this, we propose a data-driven algorithm that adaptively achieves the minimax risk, up to a logarithmic factor, over a wide range of parameter spaces. Simulation studies are conducted to further demonstrate the performance of the adaptive transfer learning algorithm and validate the phenomena discussed.

Simulation studies are conducted to evaluate the performance of the adaptive transfer learning algorithm. The numerical results further support our theoretical analysis. The proposed method is then applied to a wine quality dataset (Cortez et al., 2009) to compare the performance of direct local polynomial regression on red wine data to using the transfer learning algorithm on both red and white wine data with varying numbers of observations. The results show that the transfer learning method improves performance.

The results and algorithms can also be extended to the setting of multiple source distributions. Suppose there are K source distributions $(P_1, \dots, P_K)$ and one target distribution Q. Each source distribution P_j corresponds to a mean function g_j and the difference between f and each g_j can be well approximated by a polynomial function of a given order. We establish the non-asymptotic minimax risk and construct an adaptive procedure that simultaneously attains the optimal risk, up to a logarithmic factor, over a large collection of parameter spaces.

1.2. *Related Literature.* The problem of transfer learning for nonparametric regression in the posterior drift setting has been studied in Wang et al. (2016), under the assumption that the mean functions from both domains have the same Sobolev smoothness and the difference belongs to a smoother Sobolev class. An upper bound for the performance of transfer learning was obtained, however, no lower bound was provided, leaving it unclear if the upper bound is sharp. Additionally, their upper bound can only be achieved when the smoothness is known, which is not typically the case in practice and no adaptation to smoothness was considered. In contrast, in the present paper, we allow the mean functions to have different Hölder smoothness and assume that the difference function can be approximated in L_1 distance by a polynomial function. We prove matching upper and lower bounds, up to a logarithmic factor, to quantify what transfer learning can achieve. In the covariate shift setting, transfer learning for nonparametric regression has also been considered in Huang et al. (2006); Wen et al. (2014).

For nonparametric transfer learning, much attention has been given to classification, with the general problem being studied in Ben-David et al. (2007), Blitzer et al. (2008), Mansour et al. (2009). Theoretical results and adaptive procedures have been developed in both the posterior drift setting (Cai and Wei, 2021; Reeve et al., 2021) and the covariate shift setting (Shimodaira, 2000; Sugiyama et al., 2007).

The transfer learning problem we consider here is related to the classical nonparametric regression literature, where only observations from the target domain are available. In particular, our algorithm uses local polynomial regression as a basic tool, which has been well-studied in the conventional setting (see, for example, Stone (1977), Cleveland (1979), Tsybakov (1986), Fan and Gijbels (1992), Fan (1993), and Xiao et al. (2003)).

1.3. *Organization and Notation.* The rest of the paper is organized as follows. We finish this section with the notation. Section 2 presents the precise formulation of the transfer learning problem studied in the paper. We then construct a novel algorithm in Section 3 and derive an upper bound for its risk. A matching lower bound is then established. The problem of adaptation to smoothness is considered in Section 4. A data-driven procedure is proposed and shown to be adaptively rate-optimal. Simulation studies and a real application are carried out in Section 5 and 6 to investigate the numerical performance of the adaptive algorithm. Section 7 generalizes the methods and theoretical analysis to the multiple source distributions setting and Section 8 discusses possible future directions. For reasons of space, the proofs are given in the Supplementary Material (Cai and Pu, 2022).

The following notation will be used in the paper. For any function $h : [0,1]^d \to \mathbb{R}$, let $||h||_1 = \int_{[0,1]^d} |h(x)|dx$, $||h|| = ||h||_2 = \sqrt{\int_{[0,1]^d} h^2(x)dx}$ and

$||h||_\infty = \sup_{x\in[0,1]^d}|h(x)|$. For any $a, b \in \mathbb{R}$, we define $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$. For any $\beta > 0$, let $w(\beta)$ denote the largest integer that is strictly smaller than β. Let $\mathrm{Poly}(M, T)$ denote the polynomial functions whose degree is smaller than or equal to T and coefficients have absolute values upper bounded by M. For any multi-index $\mathbf{t} = (t_1, \ldots, t_d)$ and vector $x = (x_1, \ldots, x_d)$, let $x^{\mathbf{t}} = \prod_{i=1}^d x_i^{t_i}$. For any multi-index $\mathbf{t} = (t_1, \ldots, t_d)$, define $|\mathbf{t}| = \sum_{j=1}^d t_j$. Define the multi-index class $\Lambda(k) = \{\mathbf{t} : |\mathbf{t}| \leq k\}$. For two functions $h_1(n), h_2(n) > 0$, we write $h_1(n) = O\big(h_2(n)\big)$ if $\limsup_{n\to\infty} \frac{h_1(n)}{h_2(n)} < \infty$; $h_1(n) = \tilde{O}\big(h_2(n)\big)$ if there exists a constant $C > 0$ such that $h_1(n) = O\big(h_2(n)\cdot\ln^C(n)\big)$; $h_1(n) = \Omega\big(h_2(n)\big)$ if $\liminf_{n\to\infty} \frac{h_1(n)}{h_2(n)} > 0$; $h_1(n) = \Theta\big(h_2(n)\big)$ if $h_1(n) = O\big(h_2(n)\big)$ and $h_2(n) = O\big(h_1(n)\big)$; and $h_1(n) = \tilde{\Theta}\big(h_2(n)\big)$ if $h_1(n) = \tilde{O}\big(h_2(n)\big)$ and $h_2(n) = \tilde{O}\big(h_1(n)\big)$.

2. Problem Formulation.

We begin by formally defining the Hölder smoothness as follows.

Definition 1. *(Hölder class) For any positive numbers β, L, the Hölder function class $\mathcal{H}(\beta, L)$ is defined to be the set of all functions h with continuous partial derivatives of order $w(\beta)$ that satisfy*

$$\max_{|\boldsymbol{t}|\leq w(\beta)} \sup_{x\in[0,1]^d} |D^{\boldsymbol{t}}h(x)| + \max_{|\boldsymbol{t}|=w(\beta)} \sup_{x\neq y\in[0,1]^d} \frac{|D^{\boldsymbol{t}}h(x) - D^{\boldsymbol{t}}h(y)|}{||x-y||^{\beta-|\boldsymbol{t}|}} \leq L.$$

We model the target mean function f and the source mean function g as Hölder smooth functions and assume for some constants $\beta_Q, \beta_P, L_Q, L_P > 0$,

$$f \in \mathcal{H}(\beta_Q, L_Q) \quad \text{and} \quad g \in \mathcal{H}(\beta_P, L_P).$$

In this case, we call f β_Q-smooth and g β_P-smooth.

Definition 2. *For any $\epsilon > 0, M > 0$, positive integer T, the class of functions $\Psi(\epsilon, M, T)$ is defined as all the functions h such that there exists a polynomial function $\psi(x) \in Poly(M, T)$ such that*

$$||\psi - h||_1 = \int_{[0,1]^d} |\psi(x) - h(x)|dx \leq \epsilon.$$

We assume $f - g \in \Psi(\epsilon, M, T)$. This assumption requires that g is close in L_1 distance to f plus a polynomial of order T. In this paper we only consider polynomials with coefficients bounded in absolute value by a constant M. It is possible to generalize ψ to be an arbitrary polynomial function. The discussion on this generalization is deferred to Section 8.

Definition 3. *For any $u_1, u_2 > 0$, the class of sub-Gaussian random variables $G(u_1, u_2)$ with constants $u_1, u_2 > 0$ are defined as all random variables Z such that for any $t > 0$,*

$$\mathbb{P}\Big(|Z| \geq t\Big) \leq u_1 \cdot e^{-u_2 t^2}.$$

For any $x \in [0,1]^d$ we assume the random noises $z_1 | X_1 = x$ and $z_1' | X_1' = x$ to be sub-Gaussian with some constants u_1, u_2. This assumption ensures the outcome to be not heavy-tailed. We assume the marginal distributions of X_1 and X_1' have density functions f_{den}^Q and f_{den}^P respectively and they are lower and upper bounded by constants C^L, C^U.

Given these definitions, the parameter space is defined by

$$F(\beta_Q, \beta_P, \epsilon, u_1, u_2, M, T, L_P, L_Q, C^L, C^U) = \Big\{(Q, P) : f \in \mathcal{H}(\beta_Q, L_Q),$$

(1)

$$g \in \mathcal{H}(\beta_P, L_P), f - g \in \Psi(\epsilon, M, T), z_1 | X_1, z_1' | X_1' \in G(u_1, u_2), C^L \leq f_{den}^Q, f_{den}^P \leq C^U\Big\}.$$

This space will be denoted by $F(\beta_Q, \beta_P, \epsilon)$ when there is no confusion. The minimax estimation risk over this parameter space is then defined as

$$R_{\beta_Q, \beta_P, \epsilon, u_1, u_2, M, T, L_P, L_Q, C^L, C^U}(n_Q, n_P) = \inf_{\hat{f}} \sup_{(Q,P) \in F(\beta_Q, \beta_P, \epsilon)} \mathbb{E}\,(\hat{f}(X) - f(X))^2.$$

For simplicity we may write $R_{\beta_Q, \beta_P, \epsilon, u_1, u_2, M, T, L_P, L_Q, C^L, C^U}(n_Q, n_P)$ as $R_{\beta_Q, \beta_P, \epsilon}$ or $R(n_Q, n_P)$ if there is no confusion. It is interesting to understand when and how much transfer learning can improve the estimation accuracy in the target domain. This question can be answered by comparing the transfer learning minimax risk $R_{\beta_Q, \beta_P, \epsilon}$ with the minimax risk using only data from the target domain, which is well known to be of order $n_Q^{-\frac{2\beta_Q}{2\beta_Q + d}}$.

3. Minimax Risk.

We consider in this section the minimax risk for the transfer learning problem. We begin with a brief review of local polynomial regression in Section 3.1 that serves as a basic tool for nonparametric regression in our algorithm. We then formally present in Section 3.2 the algorithm. The minimax risk is established in Section 3.3 and a discussion on interesting phenomena is given in Section 3.4.

3.1. *Local Polynomial Regression.*

Local polynomial regression has been widely recognized for its empirical success and desirable theoretical properties (Cleveland and Devlin, 1988; Fan and Gijbels, 1992; Fan, 1993). In particular, local polynomial regression achieves the minimax optimal rate over a Hölder ball with properly tuned parameters (Györfi et al., 2002).

For observations $\mathcal{D} = \{(X^{(i)}, Y^{(i)}), i = 1, \dots, n\}$, degree l and bandwidth b (we assume $1/b$ is an integer) the local polynomial regression estimate is defined as follows. Divide $[0,1)^d$ into $\frac{1}{b^d}$ hypercubes $\{\prod_{i=1}^d [b \cdot a_i, b \cdot a_i + b) : a_i = 0, \dots, 1/b - 1\}$. For each hypercube $B(a_1, \dots, a_d) = \prod_{i=1}^d [b \cdot a_i, b \cdot a_i + b)$, let all the observations whose covariates falling into this hypercube be $\{(X_{i_1}, Y_{i_1}), \dots, (X_{i_r}, Y_{i_r})\}$. Let $a_{\text{mid}} = (a_1 + b/2, \dots, a_d + b/2)$ be the center of this hypercube. The local polynomial regression estimate $\hat{f}_{\text{lpr}}$ on $B(a_1, \dots, a_d)$ is given by

$$\hat{f}_{\text{lpr}}(x; \mathcal{D}, l, b) = \sum_{\mathbf{t} \in \Lambda(l)} c_{\mathbf{t}}^{\text{lpr}} (\frac{x - a_{\text{mid}}}{b/2})^{\mathbf{t}}, \forall x \in B(a_1, \dots, a_d),$$

where $\{c_{\mathbf{t}}^{\text{lpr}} : \mathbf{t} \in \Lambda(l)\}$ are given by

$$\{c_{\mathbf{t}}^{\text{lpr}} : \mathbf{t} \in \Lambda(l)\} = \underset{\{c_{\mathbf{t}} : \mathbf{t} \in \Lambda(l)\}}{\arg\min} \sum_{m=1}^{r} (Y_{i_m} - \sum_{\mathbf{t} \in \Lambda(l)} c_{\mathbf{t}} (\frac{X_{i_m} - a_{\text{mid}}}{b/2})^{\mathbf{t}})^2.$$

The confidence upper and lower limits are constructed as follows:

$$\hat{f}_{\text{lpr}}^{ub}(x; \mathcal{D}, l, b, \beta) = \hat{f}_{\text{lpr}}(x : \mathcal{D}, l, b) + \sqrt{\ln(|\mathcal{D}|)} \cdot (b^{-\beta} + \frac{\ln^2(|\mathcal{D}|)}{\sqrt{|\mathcal{D}| \cdot b^d}}),$$

$$\hat{f}_{\text{lpr}}^{lb}(x; \mathcal{D}, l, b, \beta) = \hat{f}_{\text{lpr}}(x : \mathcal{D}, l, b) - \sqrt{\ln(|\mathcal{D}|)} \cdot (b^{-\beta} + \frac{\ln^2(|\mathcal{D}|)}{\sqrt{|\mathcal{D}| \cdot b^d}}).$$

The length of confidence interval is then

$$L_{\text{CI,lpr}}(\mathcal{D}, l, b, \beta) = 2 \cdot \sqrt{\ln(|\mathcal{D}|)} \cdot (b^{-\beta} + \frac{\ln^2(|\mathcal{D}|)}{\sqrt{|\mathcal{D}| \cdot b^d}}).$$

3.2. *The Confidence Thresholding(CT) Algorithm.* We now present our main transfer learning algorithm. Since we have data from both the source and target domains, the most important and challenging step of this algorithm is integrating the information from both domains. This step in our algorithm is called confidence thresholding. We shall first present this confidence thresholding procedure and then show the details of our algorithm.

3.2.1. *The Confidence Thresholding Estimator.* We first introduce the confidence thresholding estimator that is designed to estimate a function when one has access to two different estimates. Suppose we have two different estimators $\hat{h}_1$ and $\hat{h}_2$ for some unknown function h. $\hat{h}_1$ converges slower and $\hat{h}_2$ converges faster but is slightly biased, which means $\hat{h}_2$ converges to a function that is different from but close to h in L_1 distance. The confidence thresholding estimator is constructed based on $\hat{h}_1$ and $\hat{h}_2$ as follows. Let e_1 be an upper bound of the L_∞ norm of $\hat{h}_1 - h$. Then a "confidence interval" for $h(x)$ is $[\hat{h}_1(x) - e_1, \hat{h}_1(x) + e_1]$ for all $x \in [0,1)^d$. There are three different possible cases of the relationship between this confidence interval and $\hat{h}_2(x)$. If $\hat{h}_2(x)$ is greater than the upper bound of the confidence interval, then this confidence interval upper bound is better than $\hat{h}_2(x)$ in estimating $h(x)$ and in this case the confidence interval upper bound is used as the estimate. If $\hat{h}_2(x)$ is in the confidence interval, then $\hat{h}_2(x)$ is acceptable and we use $\hat{h}_2(x)$. If $\hat{h}_2(x)$ is smaller than the lower bound of the confidence interval, then the confidence interval lower bound is better than $\hat{h}_2(x)$ in estimating $h(x)$ and the confidence interval lower bound is used as the estimate. We call this estimator the *confidence thresholding estimator*:

$$\hat{\mu}_{\text{ct}}(\hat{h}_1(x), \hat{h}_2(x), e_1) = \hat{h}_1(x) + \operatorname{sgn}(\hat{h}_2(x) - \hat{h}_1(x)) \cdot (|\hat{h}_2(x) - \hat{h}_1(x)| \wedge e_1). \tag{2}$$

See Figure 1 for an illustration of the confidence thresholding estimator.

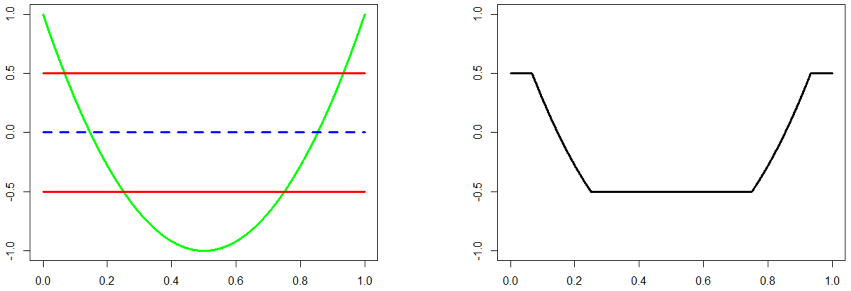


Fig 1: An illustration of the confidence thresholding estimator. On the left panel, the blue dashed line is $\hat{h}_1$, the green line is $\hat{h}_2$ and two red lines are the confidence upper bound $\hat{h}_1 + e_1$ and lower bound $\hat{h}_1 - e_1$. On the right panel, the black line is the confidence thresholding estimator $\hat{\mu}_{\text{ct}}(\hat{h}_1(x), \hat{h}_2(x), e_1)$.

The following lemma justifies the convergence of $\hat{\mu}_{\text{ct}}$.

LEMMA 1. *Suppose for a function $h : [0,1]^d \to \mathbb{R}$, we have two estimates $\hat{h}_1$ and $\hat{h}_2$. Suppose for some $e_1, e_2, e_2' > 0$, $||h - \hat{h}_1||_\infty \leq e_1$ and $||h + \tilde{h} - \hat{h}_2||_\infty \leq e_2 \leq e_1$ where function $\tilde{h} : [0,1]^d \to \mathbb{R}$ satisfies $||\tilde{h}||_1 \leq e_2'$. Let $\hat{h}(x) = \hat{\mu}_{\text{ct}}(\hat{h}_1(x), \hat{h}_2(x), e_1)$ for all $x \in [0,1]^d$. Then*

$$||\hat{h} - h||_2 \leq (e_2 + \sqrt{4e_1 \cdot e_2'}) \wedge 2e_1.$$

With this lemma we can compare the confidence thresholding estimator with the two original estimators. In the setting of this lemma, the L_2 error of $\hat{h}_1$ is upper bounded by e_1 and the L_2 error of $\hat{h}_2$ can be arbitrarily large since $||\tilde{h}_2||_2$ can be arbitrarily large. The L_2 error of the confidence thresholding estimator is upper bounded by $2e_1$, so it is at least as good as the $\hat{h}_1$ up to a constant. Besides, in the case where $e_2' \ll e_1$ and $e_2 \ll e_1$, the confidence thresholding estimator outperforms both of two original estimators.

3.2.2. *The Confidence Thresholding(CT) Algorithm.* We now present in detail the CT algorithm, which utilizes the confidence thresholding estimator and involves fitting local polynomial regression twice to produce two preliminary estimators. These estimators are used to mimic $\hat{h}_1$ and $\hat{h}_2$ in the confidence thresholding estimator.

We begin with sample splitting by randomly dividing $\mathcal{D}_Q$ into two equal-sized subsets $\mathcal{D}_{Q,1}$ and $\mathcal{D}_{Q,2}$. We first fit local polynomial regression on $\mathcal{D}_{Q,1}$ with some bandwidth $\tilde{\beta}_Q$ and obtain an estimate $\hat{f}_{\text{ref}}$. We also construct a confidence interval and compute its length, which is denoted by L_{CI}. In the confidence thresholding estimator, $\hat{f}_{\text{ref}}$ will serve as $\hat{h}_1$ and $L_{CI}/2$ will serve as e_1. We then fit another local polynomial regression to mimic $\hat{h}_2$. We can fit it either on $\mathcal{D}_{Q,1}$ or $\mathcal{D}_P$ because $\hat{h}_2$ is allowed to be biased. To get a faster convergence, we fit this local polynomial regression on the dataset with some larger bandwidth $\tilde{b}$. Let this estimate be $\hat{f}_{\text{raw}}$.

Note $f - g$ is close to a polynomial function in L_1 distance. Then $\hat{f}_{\text{raw}}$ plus some polynomial function ψ should be close to f. If ψ were known, the confidence thresholding estimator $\hat{f}_{ct}(\cdot;\psi) = \hat{\mu}_{\text{ct}}(\hat{f}_{\text{ref}}, \hat{f}_{\text{raw}} + \psi, L_{CI}/2)$ could be used to estimate $f(x)$. However, since ψ is unknown, we shall first estimate it and then plug the estimate into the confidence thresholding estimator. The estimator $\hat{\psi}$ is obtained by minimizing the empirical mean squared error on the validation set $\mathcal{D}_{Q,2}$. Formally,

$$\hat{\psi} = \underset{\psi \in \text{Poly}(\sqrt{\ln(n_Q)}, l)}{\arg\min} \sum_{i=\lfloor \frac{n_Q}{2} \rfloor + 1}^{n_Q} [Y_i - \hat{f}_{\text{ct}}(X_i;\psi)]^2.$$

Finally, we truncate the estimate $\hat{\mu}_{\mathrm{ct}}(\hat{f}_{\mathrm{ref}}(x), \hat{f}_{\mathrm{raw}} + \hat{\psi}, L_{CI}/2)$ since f is bounded. The CT algorithmx is summarized in Algorithm 1.

Algorithm 1: $\mathcal{A}^{1}(\tilde{\beta}_Q, \tilde{\beta}_P, l)$

```
Input: Hölder smoothness $\tilde{\beta}_Q$, $\tilde{\beta}_P$ and polynomial degree $l$.
Split $\mathcal{D}_Q$ into $\mathcal{D}_{Q,1} = \{(X_1, Y_1), \ldots, (X_{\lfloor \frac{n_Q}{2} \rfloor}, Y_{\lfloor \frac{n_Q}{2} \rfloor})\}$ and
  $\mathcal{D}_{Q,2} = \{(X_{\lfloor \frac{n_Q}{2} \rfloor + 1}, Y_{\lfloor \frac{n_Q}{2} \rfloor + 1}), \ldots, (X_{n_Q}, Y_{n_Q})\}$.
Calculate $\tilde{\beta}_{\max} = \max(\tilde{\beta}_Q, \tilde{\beta}_P)$, $n_{Q,1} = |\mathcal{D}_{Q,1}|$, $n_P = |\mathcal{D}_P|$, $\tilde{n}_{\max} = \max(n_{Q,1}, n_P)$,
  $\tilde{b} = \frac{1}{\lfloor \tilde{n}_{\max}^{\frac{1}{2\tilde{\beta}_{\max}+d}} \rfloor}$, $\tilde{b}_Q = \frac{1}{\lfloor n_{Q,1}^{\frac{1}{2\tilde{\beta}_Q+d}} \rfloor}$.
Fit local polynomial regression on $\mathcal{D}_{Q,1}$ with bandwidth $\tilde{b}_Q$, and calculate
  estimates and confidence interval length.
      $\hat{f}_{\mathrm{ref}}(\cdot) = \hat{f}_{\mathrm{lpr}}(\cdot; \mathcal{D}_{Q,1}, l, \tilde{b}_Q)$,
      $L_{CI} = L_{\mathrm{CI,lpr}}(\mathcal{D}_{Q,1}, l, \tilde{b}_Q, \tilde{\beta}_Q)$.
if $n_{Q,1} > n_P$ then
    fit local polynomial regression on $\mathcal{D}_{Q,1}$ with bandwidth $\tilde{b}$,
      $\hat{f}_{\mathrm{raw}}(\cdot) = \hat{f}_{\mathrm{lpr}}(\cdot; \mathcal{D}_{Q,1}, l, \tilde{b})$,
else
    fit local polynomial regression on $\mathcal{D}_P$ with bandwidth $\tilde{b}$,
      $\hat{f}_{\mathrm{raw}}(\cdot) = \hat{f}_{\mathrm{lpr}}(\cdot; \mathcal{D}_P, l, \tilde{b})$.
Estimate $\hat{\psi}$ by
      $\hat{\psi} = \mathop{\arg\min}_{\psi \in \mathrm{Poly}(\sqrt{\ln(n_Q)}, l)} \sum_{i=n_{Q,1}+1}^{n_Q} [Y_i - \hat{f}_{\mathrm{ct}}(X_i; \psi)]^2$,
  where $\hat{f}_{\mathrm{ct}}(x; \psi) = \tilde{\mu}_{\mathrm{ct}}(\hat{f}_{\mathrm{ref}}(x), \hat{f}_{\mathrm{raw}}(x) + \psi(x), L_{CI}/2)$.
Truncate the estimate at $n_Q$,
      $\hat{f}(x) = \mathrm{sgn}(\hat{f}_{\mathrm{ct}}(x; \hat{\psi})) \cdot (|\hat{f}_{\mathrm{ct}}(x; \hat{\psi})| \wedge n_Q)$
```

Remark 1. The bandwidths $\tilde{b}_Q$ and $\tilde{b}$ are chosen such that $\tilde{b}_Q \propto n_Q^{-\frac{1}{2\tilde{\beta}_Q+d}}$ and $\tilde{b} \propto (n_Q \vee n_P)^{-\frac{1}{2\tilde{\beta}_{\max}+d}}$, due to the fact that $n^{-\frac{1}{2\beta+d}}$ is the optimal bandwidth for estimating a β-smooth function based on n observations (Györfi et al., 2002).

3.3. *Minimax Risk.* The following theorem gives an upper bound for the risk of the CT algorithm. Recall $\beta_{\max} = \beta_Q \vee \beta_P$ and $n_{\max} = n_P \vee n_Q$.

Theorem 1 (Minimax upper bound). *Suppose in Algorithm 1, $\tilde{\beta}_Q = \beta_Q, \tilde{\beta}_P = \beta_P, l \geq w(\beta_{\max}) \vee T$, then the risk of this algorithm satisfies*

$$R(\hat{f}) = \sup_{(Q,P)\in F(\beta_Q,\beta_P,\epsilon)} \mathbb{E}(\hat{f}(X) - f(X))^2 \leq$$

$$C_U \cdot \left(n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} \ln^4(n_{\max}) + \ln^8(n_Q) \cdot (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{\ln^4(n_Q)}{n_Q} \right),$$

for some constant $C_U > 0$ that only depends on $\beta_Q, \beta_P, u_1, u_2, M, T, d, l, L_P, L_Q, C^L, C^U$ and not on n_Q, n_P, ϵ.

The next theorem provides a lower bound for the minimax risk and shows that CT algorithm is minimax optimal up to a logarithmic factor.

Theorem 2 (Minimax lower bound). *There exists a constant $C_L > 0$ that only depends on $\beta_Q, \beta_P, u_1, u_2, M, T, d, L_P, L_Q$ and not on n_Q, n_P, ϵ such that*

$$R_{\beta_Q,\beta_P,\epsilon} \geq C_L \cdot \left(n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} + (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{1}{n_Q} \right).$$

Theorems 1 and 2 together show that the non-asymptotic minimax risk of transfer learning for nonparametric regression $R_{\beta_Q,\beta_P,\epsilon}$ is proportional to

$$n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} + (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{1}{n_Q}. \tag{3}$$

Comparing this risk with the minimax risk of nonparametric regression with the observations from the target domain only, which is proportional to $n_Q^{-\frac{2\beta_Q}{2\beta_Q+d}}$, we can see when and how transfer learning improve the estimation accuracy for f. The sufficient and necessary condition is that the bias strength $\epsilon \ll n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$ and either the source domain has a smoother mean function $\beta_P > \beta_Q$ or much more observations $n_P \gg n_Q$.

The second term $(\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$ in (3) represents the influence of the bias strength to the difficulty of the current problem. It has two phase transition points. The first is $n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$. If the bias strength is larger than it then the minimax risk (3) is as large as the minimax risk of regression with the target domain data only, which is proportional to $n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$. If the bias strength is smaller than it then whether the minimax risk (3) is smaller than $n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$ does not depend on the bias strength. In other words, $n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}$ is the maximum tolerable bias strength for transfer learning to help and quantifies the robustness of this model. The second phase transition point is $n_Q^{-\frac{\beta_Q+d}{2\beta_Q+d}} \vee n_Q^{\frac{\beta_Q}{2\beta_Q+d}} \cdot n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}}$. Whether the bias strength is larger than it determines whether the influence of the bias strength is the dominating term. In other words, if the bias strength is smaller than it then transfer learning can work as if there is no bias.

The first term in equation (3) is equivalent to the minimax rate for nonparametric regression over a $\beta_{\max}$-smooth Hölder class with $n_{\max}$ observations. This suggests that transfer learning can benefit from larger sample sizes and improved smoothness, regardless of whether these advantages are present in different domains. Essentially, transfer learning allows for the transfer of sample size and smoothness to a common domain.

3.4. *Discussion.* We now take a closer look at the minimax risk in cases where the bias strength is strong enough to not be the dominant term in the minimax risk.We explore two unique phenomena displayed by the minimax risk: *auto-smoothing* and *super-acceleration.*

- **Auto-smoothing:** When $\beta_P < \beta_Q$, the minimax rate is

$$n_{\max}^{-\frac{2\beta_Q}{2\beta_Q+d}} + (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{1}{n_Q},$$

  which does not depend on β_P. This implies that even if g is highly irregular ($\beta_P \approx 0$), it is still possible to estimate f as if g is a β_Q-smooth function. The CT algorithm only relies on $\tilde{\beta}_Q$ and $\tilde{\beta}_{\max}$, thus it is not affected by $\tilde{\beta}_P$ if $\tilde{\beta}_P \leq \tilde{\beta}_Q$. This aligns with the auto-smoothing phenomenon observed in minimax theory.
- **Super-acceleration:** In transfer learning, a common question of interest is whether and to what extent observations from the source domain can significantly improve estimation accuracy in the target domain. In this scenario, if the source domain has a smoother mean function but a smaller sample size, i.e. $\beta_P > \beta_Q$ and $n_P < n_Q$, and the bias strength ϵ is sufficient, then the minimax risk for transfer learning is $R_{\beta_Q,\beta_P,\epsilon} = n_Q^{-\frac{2\beta_P}{2\beta_P+d}}$, which is smaller than both the minimax risk for estimating f using data from the target domain alone and the minimax risk for estimating g using data from the source domain alone. This phenomenon is referred to as *super-acceleration.* This provides

new insights into transfer learning by demonstrating that it can significantly enhance performance on the target domain even if the task in the source domain is more difficult (based on data from the source domain alone). Similarly, super-acceleration also occurs if the source domain has a rougher mean function and more observations.
On the other hand, if the source domain has a smoother mean function and a larger sample size, it is not surprising that transfer learning can improve the convergence rate. In this case, f can be estimated as accurately as g, as the minimax risk for transfer learning is of order $n_P^{-\frac{2\beta_P}{2\beta_P+d}}$ when ϵ is sufficiently small. There have been other results in transfer learning for different tasks where the best one can do on the target domain is as good as the performance of the corresponding task on the source domain. This kind of acceleration is referred to as *normal acceleration*. The following table summarizes different cases.

	$\beta_Q > \beta_P$	$\beta_Q = \beta_P$	$\beta_Q < \beta_P$
$n_Q \gg n_P$	no acceleration	no acceleration	super-acceleration
$n_Q \propto n_P$	no acceleration	no acceleration	normal acceleration
$n_Q \ll n_P$	super-acceleration	normal acceleration	normal acceleration

4. Adaptive Confidence Thresholding Algorithm.

Section 3 establishes the non-asymptotic minimax risk for estimation over the parameter space $F(\beta_Q, \beta_P, \epsilon)$ and the optimality of the CT algorithm. However, the CT algorithm requires the knowledge of the smoothness parameters β_Q and β_P, which are typically unknown. A natural and important question is whether it is possible to construct a data-driven algorithm that adaptively achieves the optimal risk simultaneously over a wide rage of parameter spaces.

In this section we develop an adaptive algorithm, called adaptive confidence thresholding (ACT) algorithm, that is based on the CT algorithm and consists of three main steps:

- **Step 1: Constructing a set of smoothness parameter pairs.** Since the CT algorithm only depends on $\tilde{\beta}_Q$ and $\tilde{\beta}_{\max}$, we construct a finite set $S_Q \subset \mathbb{R}$ for $\tilde{\beta}_Q$'s and a finite set $S_{\max}$ for $\tilde{\beta}_{\max}$'s. $\tilde{\beta}_Q$ and $\tilde{\beta}_{\max}$ are to be chosen from S_Q and $S_{\max}$ respectively. S_Q and $S_{\max}$ are both arithmetic sequences. The common differences of these two sequences are roughly proportion to $\frac{1}{\ln(n_Q)}$ and $\frac{1}{\ln(n_{\max})}$ respectively.
- **Step 2: Selecting the best pair of smoothness parameters.** For each pair of $\tilde{\beta}_Q$ and $\tilde{\beta}_{\max}$ we can construct an estimator $\hat{f}_{\mathrm{ct}}$ as in the CT algorithm. We select the best smoothness parameters β_Q^* and $\beta_{\max}^*$ by minimizing the empirical MSE on the validation data $\mathcal{D}_{Q,2}$.

- **Step 3: Plugging the selected smoothness parameters into the CT algorithm.** Run the CT algorithm with $(\beta_Q^*, \beta_{\max}^*)$ as the smoothness parameters.

The ACT algorithm is summarized in Algorithm 2.

Algorithm 2: $\mathcal{A}^2(l)$

```
Input: polynomial degree $l$.
Split $\mathcal{D}_Q$ into $\mathcal{D}_{Q,1} = \{(X_1, Y_1), \ldots, (X_{\lfloor \frac{n_Q}{2} \rfloor}, Y_{\lfloor \frac{n_Q}{2} \rfloor})\}$ and
  $\mathcal{D}_{Q,2} = \{(X_{\lfloor \frac{n_Q}{2} \rfloor + 1}, Y_{\lfloor \frac{n_Q}{2} \rfloor + 1}), \ldots, (X_{n_Q}, Y_{n_Q})\}$.
Let $n_{Q,1} = |\mathcal{D}_{Q,1}|$, $\tilde{n}_{\max} = \max(n_{Q,1}, n_P)$. Let $\tilde{n} = \min(\ln(\tilde{n}_{\max}), n_{Q,1})$
Let $S_{\max} = \{\frac{1}{\tilde{n}}, \frac{2}{\tilde{n}}, \ldots, \frac{\lfloor (l+1)\cdot\tilde{n} \rfloor}{\tilde{n}}\}$. Let $S_Q = \{\frac{1}{\ln(n_{Q,1})}, \frac{2}{\ln(n_{Q,1})}, \ldots, \frac{\lfloor (l+1)\cdot\ln(n_{Q,1}) \rfloor}{\ln(n_{Q,1})}\}$
for $\tilde{\beta}_Q \in S_Q, \tilde{\beta}_{\max} \in S_{\max}$ do
    Calculate $\tilde{b} = \frac{1}{\lfloor \tilde{n}_{\max}^{\frac{1}{2\tilde{\beta}_{\max}+d}} \rfloor}$, $\tilde{b}_Q = \frac{1}{\lfloor n_{Q,1}^{\frac{1}{2\tilde{\beta}_Q+d}} \rfloor}$.
    if $n_{Q,1} > n_P$ then
        fit local polynomial regression on $\mathcal{D}_{Q,1}$ with bandwidth $\tilde{b}$,
          $\hat{f}_{\mathrm{raw}}(\cdot; \tilde{\beta}_{\max}) = \hat{f}_{\mathrm{lpr}}(\cdot; \mathcal{D}_{Q,1}, l, \tilde{b})$,
    else
        fit local polynomial regression on $\mathcal{D}_P$ with bandwidth $\tilde{b}$,
          $\hat{f}_{\mathrm{raw}}(\cdot; \tilde{\beta}_{\max}) = \hat{f}_{\mathrm{lpr}}(\cdot; \mathcal{D}_P, l, \tilde{b})$.
    Fit local polynomial regression on $\mathcal{D}_{Q,1}$ with bandwidth $\tilde{b}_Q$, and calculate
      estimates and confidence interval length.
        $\hat{f}_{\mathrm{ref}}(\cdot; \tilde{\beta}_Q) = \hat{f}_{\mathrm{lpr}}(\cdot; \mathcal{D}_{Q,1}, l, \tilde{b}_Q)$,
        $L_{CI}(\tilde{\beta}_Q) = L_{\mathrm{CI,lpr}}(\mathcal{D}_{Q,1}, l, \tilde{b}_Q, \tilde{\beta}_Q)$.
Estimate $\hat{\psi}, \beta_Q^*, \beta_{\max}^*$ by
  $\hat{\psi}, \beta_Q^*, \beta_{\max}^* = \underset{\psi \in \mathrm{Poly}(\sqrt{\ln(n_Q)}, l), \tilde{\beta}_Q \in S_Q, \tilde{\beta}_{\max} \in S_{\max}}{\arg\min} \sum_{i=n_{Q,1}+1}^{n_Q} [Y_i - \hat{f}_{\mathrm{ct}}(X_i; \psi, \tilde{\beta}_Q, \tilde{\beta}_{\max})]^2$,
  where $\hat{f}_{\mathrm{ct}}(x; \psi, \tilde{\beta}_Q, \tilde{\beta}_{\max}) = \tilde{\mu}_{\mathrm{ct}}(\hat{f}_{\mathrm{ref}}(x; \tilde{\beta}_Q), \hat{f}_{\mathrm{raw}}(x; \tilde{\beta}_{\max}) + \psi(x), L_{CI}(\tilde{\beta}_Q)/2)$.
Truncate the estimate at $n_Q$,
  $\hat{f}_{\mathrm{ada}}(x) = \mathrm{sgn}(\hat{f}_{\mathrm{ct}}(x; \hat{\psi}, \beta_Q^*, \beta_{\max}^*)) \cdot (|\hat{f}_{\mathrm{ct}}(x; \hat{\psi}, \beta_Q^*, \beta_{\max}^*)| \wedge \tilde{n}_Q)$
```

Note that in Step 2 we minimize the empirical mean squared error on validation data $\mathcal{D}_{Q,2}$ to select the best polynomial function ψ for each pair of $\tilde{\beta}_Q$ and $\tilde{\beta}_{\max}$ and in Step 3 we minimize the same empirical mean squared error on the same validation data to select the best pair of $\tilde{\beta}_Q$ and $\tilde{\beta}_{\max}$.

Therefore we can combine these two steps in the algorithm table into one step, where we minimize empirical mean squared error on validation data among all choices of $(\tilde{\beta}_Q, \tilde{\beta}_{\max})$ and polynomial function ψ.

THEOREM 3 (Adaptive upper bound). *Suppose in Algorithm 2, $l \geq w(\beta_{\max}) \vee T$, then the risk of this algorithm satisfies*

$$R(\hat{f}_{\text{ada}}) = \sup_{(Q,P)\in F(\beta_Q,\beta_P,\epsilon)} \mathbb{E}(\hat{f}_{\text{ada}}(X) - f(X))^2$$
$$\leq C_U^{\text{ada}} \cdot \left(n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} \cdot \ln^4(n_{\max}) + \ln^8(n_Q) \cdot (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{\ln^4(n_Q)}{n_Q} \right),$$

for some constant $C_U^{\text{ada}} > 0$ not depending on n_Q, n_P, ϵ.

Therefore the data-driven estimator simultaneously achieves the minimax risk, up to a logarithmic factor, for a large collection of parameter spaces.

REMARK 2. Theorem 3 holds when the intermediate term $L_{\text{CI,lpr}}(\mathcal{D}_{Q,1}, l, \tilde{b}_Q, \tilde{\beta}_Q)$ is equal to $C \cdot \sqrt{\ln(|\mathcal{D}_{Q,1}|)} \cdot (\tilde{b}_Q^{-\tilde{\beta}_Q} + \frac{\ln^2(|\mathcal{D}_{Q,1}|)}{\sqrt{|\mathcal{D}_{Q,1}| \cdot \tilde{b}_Q^d}})$ for any constant $C > 0$. In Algorithm 2, C is taken to be 2. In practice, it may be beneficial to tune this constant C while using the algorithm. This process can be easily integrated into the algorithm, by tuning C along with the parameters $\tilde{\beta}_Q$ and $\tilde{\beta}_{\max}$ on the second half of the dataset.

5. Simulation.

In this section, we evaluate the performance of the ACT algorithm through simulations and compare it to existing methods. The numerical results further support our theoretical analysis.

Recall that the minimax risk (3) is affected by both the sample size and the bias strength. To demonstrate their impact on the empirical performance, we conduct two series of experiments. In the first series, we fix all other parameters and vary the sample size. In the second series, we fix all other parameters and vary the bias strength. In all experiments, we set the dimension to 1, the covariate distributions on both the source and target domains to uniform distribution on $[0, 1]$, and the random noise on both domains to normal random variables with zero mean and standard deviation 1/3. We evaluate the performance of all algorithms using the mean squared error (MSE), which is the expected squared L_2 distance between the estimator and the true mean function. The MSE is calculated by averaging 2000 random repeated experiments.

In the first series of experiments, we investigate the influence of sample size by fixing the bias strength and varying the sample size. Specifically we let the mean functions be

$$
\begin{aligned}
f(x) &= \sin(10\pi x) + x^{3/2} - 0.1x + (0.1 - |x - 0.5|)_+, \\
g(x) &= \sin(10\pi x) + x^{3/2}.
\end{aligned}
$$

Therefore g differ from f by a linear function and a small spike with width 0.2 and height 0.1. In this case $f \in \mathcal{H}(1, 40)$ and $g \in \mathcal{H}(3/2, 40)$. The sample size of the target domain is fixed at 200. The sample sizes of the source domain are taken to be $(300, 600, 1200, 2400, 4800)$. In this series of experiments, g is smoother than f and n_P is greater than n_Q, so g is easier to estimate. We compare the performance of the ACT algorithm to that of local polynomial regression using only data from the target domain. The bandwidth for local polynomial regression is determined through a five-fold cross-validation method. By comparing the performance of ACT to local polynomial regression, we are able to gauge the improvement gained through transfer learning with various sample sizes from the target domain.

In the second series of experiments, we investigate the impact of bias strength by fixing the sample size and varying the bias strength. Specifically we let the mean functions be

$$
f(x) = \sin(10\pi x) + x^{3/2},
$$

$$
g(x) = \sin(10\pi x) + x^{3/2} - 0.1x + (3 - \frac{6}{l_{wid}}|x - 0.5|)_+,
$$

where l_{wid} is taken to be $0, 0.005, 0.01, 0.015, 0.02$ respectively in each of the five experiments. In this case g is equal to f plus a linear function plus a spike with width l_{wid} and height 3. Note $\epsilon = l_{wid} * 3/2$ in this case. Therefore the bias strengths are $(0, 0.00375, 0.0075, 0.01125, 0.015)$. For each of the latter four cases $l_{wid} \in (0.005, 0.01, 0.015, 0.02)$, $g \in \mathcal{H}(1, \frac{6}{l_{wid}} + 40)$ and in the first case where $l_{wid} = 0$, $g \in \mathcal{H}(3/2, 40)$. The sample sizes of the source and target domains are fixed at 200 and 600, respectively. We compare the performance of ACT with local polynomial regression using only the observations from the target domain to study the effects of transfer learning with varying bias strengths. Additionally, comparisons are made with the performance of local polynomial regression using only the observations from the source domain to estimate g. These comparisons help illustrate the super-acceleration phenomenon. Both local polynomial regressions are fitted using bandwidths selected through five-fold cross-validation.

Figure 2a presents the results of the first series of experiments, specifically the MSEs of local polynomial regression with cross validation and the

ACT algorithm for various sample sizes. As noted, in the first series of experiments, g is smoother and has more corresponding observations, making it easier to estimate. The plot clearly demonstrates the gap in performance between the ACT algorithm and local polynomial regression as predicted by theory. Additionally, the plot indicates that the ACT algorithm's performance improves as the sample size from the source domain increases, however, this improvement seems to level off when n_P is large ($n_P > 2400$). This is also consistent with the minimax theory. The minimax risk (3) in this case is proportional to

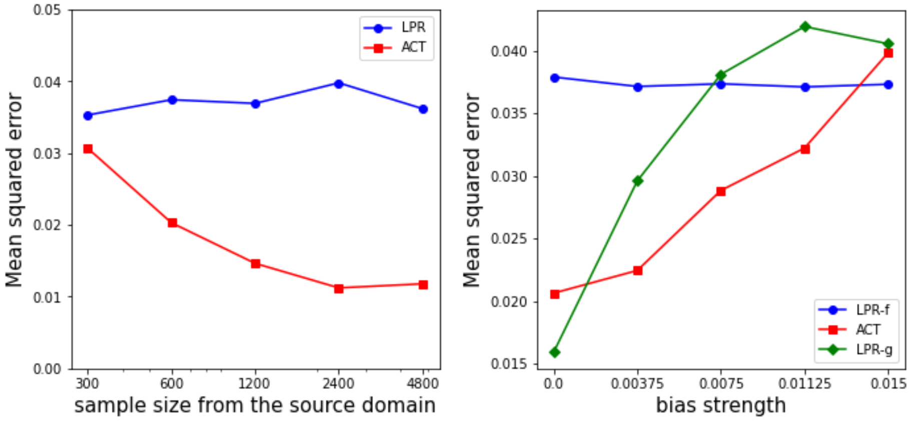
$$n_P^{-\frac{3}{4}} + (\epsilon \wedge n_Q^{-\frac{1}{3}}) \cdot n_Q^{-\frac{1}{3}} + n_Q^{-1},$$

which decreases as n_P grows when n_P is not large and keeps fixed when n_P is large enough such that $n_P^{-\frac{3}{4}}$ is dominated by the following two terms.

Figure 2b illustrates the simulation results of the second series of experiments. Specifically, it shows the MSEs of local polynomial regression with cross validation for both f and g and ACT algorithm with different bias strength. We first compared the ACT algorithm and local polynomial regression for estimating f using observations from the target domain only. The results showed a clear gap in the MSE between the two methods when the bias strength was small enough ($\epsilon < 0.01125$). As the bias strength increased, the MSE of ACT grew and eventually became as large as the MSE of local polynomial regression when the bias strength was large enough ($\epsilon > 0.015$). These findings are consistent with the theory of minimax risk, which predicts that transfer learning can improve performance when bias strength is small and worsen as it increases. To further illustrate different types of acceleration, we also compared the performance of local polynomial regression for estimating g. In the special case of $\epsilon = 0$, where g is as smooth as f, normal acceleration was observed, as discussed in Section 3. The results showed that ACT performed worse than estimating g with local polynomial regression but better than estimating f with local polynomial regression. In the general case where $\epsilon > 0$, g is rougher than f but has more observations. The theory predicts a super-acceleration phenomenon if the bias strength is small enough. The results showed that when the bias strength was small but nonzero ($0.00375 < \epsilon < 0.01125$), the ACT algorithm outperformed local polynomial regression for both estimating f and g. This validates the theoretical predictions.

6. Application.

In this section, an application of the adaptive estimator is demonstrated using the wine quality data from Cortez et al. (2009). The dataset comprises both red and white wine quality, which share the



(a) Experiments on different n_P (b) Experiments on different ϵ

Fig 2: MSEs of different regression methods. Blue: MSE of local polynomial regression on the target domain. Red: MSE of ACT on the target domain. Green: MSE of local polynomial regression on the source domain.

same features and outcome (wine quality). The aim is to build a regression model that predicts wine quality based on all features. The white wine dataset serves as the source domain and the red wine dataset as the target domain. The objective is to investigate if using the white wine dataset can enhance the prediction of red wine quality.

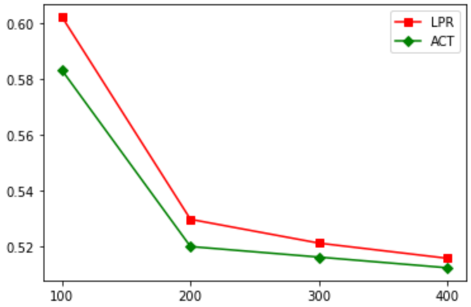


Fig 3: MSEs of different regression methods. Red: MSE of local polynomial regression on the target domain. Green: MSE of ACT.

As in Section 5, we compare the performance of local polynomial regression applied directly to the red wine dataset with that of our transfer learning algorithm. Both algorithms are based on local polynomial regression, which is suitable for low-dimensional problems. However, the original dataset has

13 features, which are too many for local polynomial regression with the given sample size. To address this, we select the most influential feature, “alcohol,” using feature importance ranking with random forest (Breiman, 2001) and only use this feature. Both local polynomial regression and the transfer learning algorithm have tuning parameters, so to compare them fairly, we use half of the training samples as the validation dataset to tune the parameters for both algorithms. The degrees of all local polynomials used in both algorithms are set to be 1. The degree of the polynomial that is used to approximate $f - g$ in ACT algorithm is also set to be 1. We let $n_P = 4898$ and $n_Q = (100, 200, 300, 400)$. The remaining observations in the target domain serve as the test data to evaluate the performance of both algorithms, which is characterized by the MSE.

Figure 3 shows the MSEs of the two algorithms with different numbers of target sample size. As more observations in the target domain are used, the relative contribution from the source dataset decreases. However, the proposed adaptive estimator consistently outperforms the naive local polynomial regression. This shows that in this application, the performance of the target task can be significantly improved by transfer learning when the source domain has many more observations.

7. Multiple Source Domains.

We have so far focused on the single source domain setting. In practical applications, it is common to have data from multiple source domains. In this section, we will expand our analysis to encompass the scenario of utilizing data from multiple source domains, and generalize the procedures and results from the single source domain case to this setting.

We consider the following model where observations from multiple source distributions $P_1, \ldots, P_K$ and one target distribution Q are available. Suppose there are n_{P_j} observations $\{(X'_{1,j}, Y'_{1,j}), \ldots, (X'_{n_{P_j},j}, Y'_{n_{P_j},j})\}$ from P_j for each $j = 1, \ldots, K$ and n_Q observations $\{(X_1, Y_1), \ldots, (X_{n_Q}, Y_{n_Q})\}$ from Q. All the observations are independent. Similar to the single-source model, let

$$\begin{aligned} Y_i &= f(X_i) + z_i, \quad i = 1, \ldots, n_Q \\ Y'_{i,j} &= g_j(X'_{i,j}) + z'_{i,j}, \quad i = 1, \ldots, n_{P_j},\ j = 1, \ldots, K. \end{aligned}$$

where $\{z_i\}$ and $\{z'_{i,j}\}$ are i.i.d. zero mean random noises. The parameter space is defined as follows:

$$F(\beta_Q, \beta_P, \epsilon, u_1, u_2, M, T, \overrightarrow{L_P}, L_Q) = \Big\{ (Q, P_1, \ldots, P_K) : f \in \mathcal{H}(\beta_Q, L_Q),$$

$$g_j \in \mathcal{H}(\beta_P, L_{P_j}), f - g_j \in \Psi(\epsilon, M, T), z_1 | X_1, z'_{1,j} | X'_{1,j} \in G(u_1, u_2), j = 1, \ldots, K \Big\},$$

where $\overrightarrow{L_P} = (L_{P_1}, \ldots, L_{P_K})$. This space will be denoted by F for simplicity when there is no confusion.

We establish the minimax risk in this section, We also construct a data-driven algorithm, which is an extension of Algorithm 2, that adaptively achieves the minimax risk up to a logarithmic factor. For reasons of space, the algorithm is given in the Supplementary Material (Cai and Pu, 2022).

THEOREM 4 (lower bound). *Let $\beta_{\max} = \max(\beta_Q, \beta_P)$ and $n_P = \sum_{j=1}^{K} n_{P,j}$. Let all assumptions be satisfied, there exists some constant $C > 0$ that only depends on $\beta_Q, \beta_P, u_1, u_2, M, K, d, L_P, L_Q$ and not on $n_Q, n_{P,1}, \ldots, n_{P,K}, \epsilon$ such that*

$$R_{\beta_Q,\beta_P,\epsilon} \geq C \cdot \left(n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} + (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{1}{n_Q} \right).$$

THEOREM 5 (adaptive upper bound). *Suppose in Algorithm 2, $l \geq w(\beta_{\max}) \vee T$, then the risk of this algorithm satisfies*

$$R(n_Q; n_P) \leq C \cdot \left(n_{\max}^{-\frac{2\beta_{\max}}{2\beta_{\max}+d}} \ln^4(n_{\max}) + \ln^8(n_Q) \cdot (\epsilon \wedge n_Q^{-\frac{\beta_Q}{2\beta_Q+d}}) \cdot n_Q^{-\frac{\beta_Q}{2\beta_Q+d}} + \frac{\ln^4(n_Q)}{n_Q} \right),$$

for some constant $C > 0$ that only depends on $\beta_Q, \beta_P, u_1, u_2, M, K, d$ and not on $n_Q, n_{P,1}, \ldots, n_{P,K}, \epsilon$.

8. Discussion.

We studied in the present paper transfer learning for nonparametric regression under the posterior drift model and established the minimax risk, which quantifies when and how much data from the source domains can improve the performance of nonparametric regression in the target domain. A novel, data-driven algorithm is developed and shown to be adaptively minimax optimal, up to a logarithmic factor, over a wide range of parameter spaces.

The minimax risk of this problem exhibits interesting and novel phenomena. The "auto-smoothing" phenomenon demonstrates that transfer learning can smooth the mean function of the source domain when it is rougher than that of the target domain. The "super-acceleration" phenomenon shows that even if the task of the source domain is more difficult, it may still be beneficial for the regression task in the target domain in certain cases. Further research in other transfer learning problems could yield similar phenomena.

We use the L_1 norm to measure bias strength in this paper, but it is easy to generalize to all L_p norms. This is because L_1 norm is smaller than or equal to all L_p norms for $p \geq 1$. Additionally, polynomial functions are used to approximate the difference between the mean functions of the source and target domains, but it could be interesting to consider other collections of functions in the future. These functions should be easier to estimate than the source and target mean functions, and examples could include infinitely differentiable functions or general Hölder functions with smoothness larger than $\beta_{\max}$.

In this paper, we consider the common support Ξ of the covariates of the source and target domains to be a hypercube of dimension d with edges of length 1, and develop methods and theory for this case. These results can also be generalized to other types of supports. Specifically, by using linear transformations, our results can be extended to all hypercube-shaped supports. Additionally, we can further generalize our results to more general types of supports by making an assumption on the measure of points not contained in a grid of hypercubes with edge length δ. If this measure is bounded by $O(\delta^{\zeta})$ for some $\zeta > 0$, our methods and theory can be applied to that support. Examples of supports that satisfy this assumption include all bounded convex sets with $\zeta = 1$. Our methods and upper bounds can be adapted to these other supports by considering only the hypercubes contained within them and ignoring the remaining points. The risk of the generalized algorithm is then upper bounded by a constant times the corresponding upper bound for the hypercube support plus $O(\delta^{\zeta})$. When ζ is large enough in relation to the smoothness parameters of the problem, this upper bound matches the lower bound.

References.

Apostolopoulos, I. D. and Mpesiana, T. A. (2020). Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2):635–640.

Ben-David, S., Blitzer, J., Crammer, K., Pereira, F., et al. (2007). Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems*, 19:137.

Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J. (2008). Learning bounds for domain adaptation. In *Proc. Conf. Empirical Methods in Natural Language*, pages 120–128.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Cai, T. T. and Pu, H. (2022). Supplement to “Transfer learning for nonparametric regression: Non-asymptotic minimax rate and adaptive procedure”.

Cai, T. T. and Wei, H. (2021). Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. *The Annals of Statistics*, 49(1):100–128.

Cao, B., Pan, S. J., Zhang, Y., Yeung, D.-Y., and Yang, Q. (2010). Adaptive transfer learning. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 407–412.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553.

Daumé III, H. (2009). Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

Fan, J. (1993). Local linear regression smoothers and their minimax efficiencies. *The Annals of Statistics*, pages 196–216.

Fan, J. and Gijbels, I. (1992). Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 20:2008–2036.

Györfi, L., Kohler, M., Krzyżak, A., and Walk, H. (2002). *A distribution-free theory of nonparametric regression*, volume 1. Springer.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2006). Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems*, 19:601–608.

Li, S., Cai, T. T., and Li, H. (2022a). Transfer learning for high-dimensional linear regression: Prediction, estimation, and minimax optimality. *Journal of the Royal Statistical Society: Series B*, (to appear).

Li, S., Cai, T. T., and Li, H. (2022b). Transfer learning in large-scale gaussian graphical models with false discovery rate control. *Journal of the American Statistical Association*, (to appear).

Li, S., Zhang, L., Cai, T. T., and Li, H. (2021). Estimation and inference for high-dimensional generalized linear models with knowledge transfer. *Technical Report*.

Mansour, Y., Mohri, M., and Rostamizadeh, A. (2009). Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*.

Mei, S., Li, H., Fan, J., Zhu, X., and Dyer, C. R. (2014). Inferring air pollution by sniffing social media. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 534–539. IEEE.

Nguyen-Tuong, D., Peters, J., Seeger, M., and Schölkopf, B. (2008a). Learning inverse dynamics: a comparison. In *European Symposium on Artificial Neural Networks*, number CONF.

Nguyen-Tuong, D., Seeger, M., and Peters, J. (2008b). Computed torque control with nonparametric regression models. In *Proc. American Control Conference*, pages 212–217. IEEE.

Reeve, H. W. J., Cannings, T. I., and Samworth, R. J. (2021). Adaptive transfer learning. *The Annals of Statistics*, 49(6):3618–3649.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244.

Stone, C. J. (1977). Consistent nonparametric regression. *The Annals of Statistics*, 5:595–620.

Sugiyama, M., Nakajima, S., Kashima, H., Von Buenau, P., and Kawanabe, M. (2007). Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, volume 7, pages 1433–1440. Citeseer.

Tsybakov, A. B. (1986). Robust reconstruction of functions by the local-approximation method. *Problemy Peredachi Informatsii*, 22(2):69–84.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition, pages 7167–7176.

Vijayakumar, S., D'souza, A., Shibata, T., Conradt, J., and Schaal, S. (2002). Statistical learning for humanoid robots. *Autonomous Robots*, 12(1):55–69.

Wang, X., Oliva, J. B., Schneider, J. G., and Póczos, B. (2016). Nonparametric risk and stability analysis for multi-task learning problems. In *IJCAI*, pages 2146–2152.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

Wen, J., Yu, C.-N., and Greiner, R. (2014). Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *International Conference on Machine Learning*, pages 631–639. PMLR.

Xiao, Z., Linton, O. B., Carroll, R. J., and Mammen, E. (2003). More efficient local polynomial estimation in nonparametric regression with autocorrelated errors. *Journal of the American Statistical Association*, 98(464):980–992.

Yeung, D.-Y. and Zhang, Y. (2009). Learning inverse dynamics by gaussian process begrression under the multi-task learning framework. In *The Path to Autonomous Robots*, pages 1–12. Springer.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

DEPARTMENT OF STATISTICS AND DATA SCIENCE
THE WHARTON SCHOOL
UNIVERSITY OF PENNSYLVANIA
PHILADELPHIA, PENNSYLVANIA 19104
USA
E-MAIL: tcai@wharton.upenn.edu
hpu@wharton.upenn.edu
URL: http://www-stat.wharton.upenn.edu/∼tcai/