# Optimal Tuning for Divide-and-conquer Kernel Ridge Regression with Massive Data

**Ganggang Xu**[1] **Zuofeng Shang**[2] **Guang Cheng**[3]

## Abstract

Divide-and-conquer is a powerful approach for large and massive data analysis. In the nonparameteric regression setting, although various theoretical frameworks have been established to achieve optimality in estimation or hypothesis testing, how to choose the tuning parameter in a practically effective way is still an open problem. In this paper, we propose  a data-driven procedure based on divide-and-conquer for selecting the tuning parameters in kernel ridge regression by modifying the popular Generalized Cross-validation (GCV, Wahba, 1990). While the proposed criterion is computationally scalable for massive data sets, it is also shown under mild conditions to be asymptotically optimal in the sense that minimizing the proposed distributed-GCV (dGCV) criterion is equivalent to minimizing the true global conditional empirical loss of the averaged function estimator, extending the existing optimality results of GCV to the divide-and-conquer framework.

## 1. Introduction

Massive data made available in various research areas have imposed new challenges for data scientists. With a large to massive sample size, many sophisticated statistical tools are no longer applicable simply due to formidable computational costs and/or memory requirements. Even when the computation is possible on more advanced machines, it is still appealing to develop accurate statistical procedures at much lower computational costs. The divide-and-conquer strategy has become a popular tool for regression models. With carefully designed algorithms, such a strategy has proven to be effective in Linear models (Chen & Xie, 2014; Lu et al., 2016), Partially linear models (Zhao et al., 2016) and Nonparametric regression models (Zhang et al., 2015; Lin et al., 2017; Shang & Cheng, 2017; Guo et al., 2017). In this paper, we shall focus on the divide-and-conquer kernel ridge regression where the selection of the penalty parameter is of vital importance but still remains unsettled.

Suppose we have independent and identically distributed samples $\{(x_i, y_i) \in \mathcal{X} \times \mathbb{R}\}_{i=1,\dots N}$ from a joint probability measure $\mathbb{P}_{Y,X}$. The goal is to study the association between the covariate $x_i$ and the response $y_i$ through the following nonparametric model

$$y_i = f_0(x_i) + \varepsilon_i, \quad i = 1, \dots, N, \tag{1}$$

where $f_0(\cdot) : \mathcal{X} \to \mathbb{R}$ is the function of interest and $\varepsilon_i$ is a random error term with mean zero and a common variance $\sigma^2$. One popular method to estimate $f_0(\cdot)$ is the *Kernel Ridge Regression* (Shawe-Taylor & Cristianini, 2004) which essentially aims at finding a projection of $f_0(\cdot)$ into a reproducing kernel Hilbert space (RKHS), denoted as $\mathcal{H}$, with a norm $\|\cdot\|_{\mathcal{H}}$. Specifically, the kernel ridge regression estimator is then defined as

$$\widehat{f} = \arg\min_{f \in \mathcal{H}} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}} \right\}, \tag{2}$$

where $\lambda \geq 0$ controls trade-off between goodness-of-fit and smoothness of $f$.

It is well known that computing $\widehat{f}$ requires $O(N^3)$ floating operations and $O(N^2)$ memory; see (5). When $N$ is large, such requirements can be prohibitive. To overcome this, Zhang et al. (2015) proposed the following "divide-and-conquer" algorithm: (i) Randomly divide the entire sample $\{(x_1, y_1), \dots, (x_N, y_N)\}$ to $m$ disjoint "smaller" subsets, denoted by $S_1, \dots, S_m$; (ii) For each subset $S_k$, find $\widehat{f}_k = \arg\min_{f \in \mathcal{H}} \left\{ \frac{1}{n_k} \sum_{i \in S_k} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}} \right\}$, where $n_k$ is the size of $S_k$; (iii) The final nonparametric estimator is given by

$$\bar{f}(x) = \frac{1}{m} \sum_{k=1}^{m} \widehat{f}_k(x). \tag{3}$$

[1] Department of Mathematical Sciences, Binghamton University, the State University of New York, Binghamton, NY, USA [2] Department of Mathematical Sciences, IUPUI, Indianapolis, IN, USA [3] Department of Statistics, Purdue University, West Lafayette, IN, USA . Correspondence to: Ganggang Xu <gang@math.binghamton.edu>.

*Proceedings of the $35^{th}$ International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

Such a "divide-and-conquer" strategy reduces computing time from $O(N^3)$ to $O(N^3/m^2)$ and memory usage from $O(N^2)$ to $O(N^2/m^2)$. Both savings could be substantial as $m$ grows. Furthermore, Zhang et al. (2015) shows that as long as $m$ does not grow too fast, the averaged estimator $\bar{f}$ achieves the same minimax optimal estimation rate as the oracle estimate $\widehat{f}$, i.e., (2), that uses all data points at once. In this sense, the divide-and-conquer algorithm is quite appealing as it achieves an ideal balance between the computational cost and the statistical efficiency.

However, the aforementioned statistical efficiency depends critically on a careful choice of tuning parameter $\lambda$ in all sub-samples. In this paper, we define a new data-driven criterion named "distributed generalized cross-validation" (dGCV) to choose tuning parameters for kernel ridge regression under the divide-and-conquer framework. The computational cost of the proposed criterion remains the same as $O(N^3/m^2)$. More importantly, we show that the proposed method enjoys similar theoretical optimality as the well-known GCV criterion (Craven & Wahba, 1978) in the sense that the resulting divide-and-conquer estimate minimizes the true empirical loss function asymptotically.

**Related work** The optimal choice of tuning parameter $\lambda$ has been well studied for kernel ridge regression when the entire data set can be fitted at once. Examples include Mallow's CP (Mallows, 2000), Generalized cross-validation (GCV, Craven & Wahba, 1978) and Generalized approximated cross-validation (Xiang & Wahba, 1996). However, if we naively apply these traditional tuning methods in each sub-sample to pick an optimal $\lambda_k$ in the above step (ii), the averaged function estimator $\bar{f}$ subsequently obtained using (3) will be sub-optimal. As pointed out by existing literature (e.g. Zhang et al., 2015; Blanchard & Mücke, 2016; Chang et al., 2017), the optimal tuning parameter should be chosen in accordance with the order of *the entire sample size*, i.e., $N$, such that we intentionally allow the resulting sub-estimator $\widehat{f}_k$ to over-fit the sub-sample $S_k$ for each $k = 1, \ldots, m$. Based on the order of the optimal choice of $\lambda$, Zhang et al. (2015) proposed a heuristic data-driven approach to empirically choose an optimal $\lambda$. However, the theoretical properties of this approach remain unclear.

## 2. Kernel Ridge Regression Estimation

In this section, we briefly review kernel ridge regression (Shawe-Taylor & Cristianini, 2004). The reproducing kernel Hilbert space, denoted as $\mathcal{H}$, is a Hilbert space induced by a symmetric nonnegative definite kernel function $K(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ satisfying

$$\langle g(\cdot), K(x,\cdot)\rangle_{\mathcal{H}} = g(x) \text{ for any } g \in \mathcal{H}.$$

The kernel function $K(\cdot,\cdot)$ is called the reproducing kernel of the Hilbert space $\mathcal{H}$ equipped with the norm $\|g\|_{\mathcal{H}} = \sqrt{\langle g(\cdot), g(\cdot)\rangle_{\mathcal{H}}}$. Using the Mercer's theorem, under some regularity conditions, the kernel function $K(\cdot,\cdot)$ possesses the expansion $K(x,z) = \sum_{j=1}^{\infty} \mu_j \psi_j(x) \psi_j(z)$, where $\mu_1 \geq \mu_2 \geq \ldots$ is a sequence of decreasing eigenvalues and $\{\psi_1(\cdot), \psi_2(\cdot), \ldots\}$ is a family of orthonormal basis functions of $L^2(\mathbb{P}_X)$. The smoothness of $g \in \mathcal{H}$ is characterized by the decaying rate of the eigenvalues $\{\mu_j\}_{j=1}^{\infty}$. There are three types of estimation considered in this paper, including smoothing spline (Wahba, 1990) as a special case.

**Finite rank:** There exists some integer $r$ such that $\mu_j = 0$ for $j > r$. For example, with scalars $x, z$, the polynomial kernel $K(x,z) = (1 + xz)^r$ has a finite rank $r + 1$, and induces a space of polynomial functions with degree at most $r$. This corresponds to the parametric ridge regression.

**Exponentially decaying:** There exist some $\alpha, r > 0$ such that $\mu_j \asymp \exp(-\alpha j^r)$. Exponentially decaying kernels include the Gaussian kernel $K(x,z) = \exp(-\|x-z\|_2^2/\phi^2)$, where $\phi > 0$ is the scale parameter and $\|\cdot\|_2$ is the Euclidean norm.

**Polynomially decaying:** There exists some $r > 0$ such that $\mu_j \asymp j^{-2r}$. The polynomially decaying class includes many smoothing spline kernels of the Sobolev space (Wahba, 1990). For example, kernel function $K(x,z) = 1 + \min(x,z)$ induces the Sobolev space of Lipschitz functions with smoothness $\nu = 1$ and has polynomially decaying eigenvalues.

With observed data, using the representor theorem (Wahba, 1990), it can be shown that the solution to the minimization problem (2) takes the following form

$$\widehat{f}(x) = \sum_{i=1}^{N} \beta_i K(x_i, x), \tag{4}$$

where $\beta_1, \ldots, \beta_N \in \mathbb{R}$. Furthermore, based on the observed sample, the parameter vector $\boldsymbol{\beta} = (\beta_1, \ldots \beta_N)^T$ can be estimated by minimizing the following criterion

$$\frac{1}{N}(\boldsymbol{Y} - \boldsymbol{\beta}^T\mathbf{K})^T(\boldsymbol{Y} - \boldsymbol{\beta}^T\mathbf{K}) + \lambda \boldsymbol{\beta}^T\mathbf{K}\boldsymbol{\beta}, \tag{5}$$

where $\boldsymbol{Y} = (y_1, \ldots, y_N)^T$ and $\mathbf{K} = [K(x_i, x_j)]_{1 \leq i,j \leq N}$.

We next apply the above idea to sub-estimation. Denote $(\boldsymbol{y}_1, \boldsymbol{x}_1), \ldots, (\boldsymbol{y}_m, \boldsymbol{x}_m)$ as a random partition of the entire data with $\boldsymbol{y}_k = (y_{k,1}, \ldots, y_{k,n_k})^T$ and $\boldsymbol{x}_k = (x_{k,1}, \ldots, x_{k,n_k})^T$. Define vectors $\boldsymbol{f}_k = (f_0(x_{k,1}), \ldots, f_0(x_{k,n_k}))^T$ and $\boldsymbol{\varepsilon}_k = \boldsymbol{y}_k - \boldsymbol{f}_k$. Define the sub-kernel matrices $\mathbf{K}_{kl} = [K(x_i, x_j)]_{i \in S_k, j \in S_l}$ for $l, k = 1, \ldots, m$. It is straightforward to show that the minimizer of (5) with $\mathbf{K}$ replaced by $\mathbf{K}_{kk}$ is of the form $\widehat{\boldsymbol{\beta}}_k = (\mathbf{K}_{kk} + n_k \lambda \mathbf{I}_k)^{-1} \boldsymbol{y}_k$, and the individual function estimator $\widehat{f}_k(x)$ can be written as

$$\widehat{f}_k(x) = \sum_{i \in S_k} \widehat{\beta}_{k,i} K(x_i, x), \tag{6}$$

where $\widehat{\beta}_{k,i}$ is the entry of $\widehat{\boldsymbol{\beta}}_k$ corresponding to $x_{k,i}$, $k = 1, \ldots, m$.

# 3. Tuning Parameter Selection

## 3.1. Sub-GCV Score: Local Optimality

In this section, we define the GCV score for each sub-estimation, named as sub-GCV score, and discuss its theoretical property. Define the empirical loss function for $\widehat{f}_k$ as follows

$$L_k(\lambda|\boldsymbol{x}_k) = \frac{1}{n_k} \sum_{i \in S_k} w_i \left\{ \widehat{f}_k(x_i) - f_0(x_i) \right\}^2, \tag{7}$$

where $w_i \geq 0$ is some weight assigned to each observation $(y_i, x_i)$ and satisfies $\sum_{i \in S_k} w_i = n_k$. The introduction of weights in (7) helps reducing computational cost; see Section 3.4. The tuning parameter $\lambda$ is referred to as "locally optimal" if it only minimizes local empirical loss $L_k(\lambda|\boldsymbol{x}_k)$. When only focused on a single sub-data set, such a "locally-optimal" choice of tuning parameter $\lambda$ has been well studied in (Craven & Wahba, 1978; Li, 1986; Gu, 2013; Wood, 2004; Gu & Ma, 2005; Xu & Huang, 2012), among which the most popular method remains to be the Generalized Cross-Validation (Craven & Wahba, 1978).

Using the function estimator $\widehat{f}_k(x)$, the predicted values for the vector $\boldsymbol{y}_k$ can be written as $\widehat{\boldsymbol{y}}_k = \mathbf{A}_{kk}(\lambda)\boldsymbol{y}_k$, where $\mathbf{A}_{kk}(\lambda) = \mathbf{K}_{kk}(\mathbf{K}_{kk} + n_k\lambda\mathbf{I}_k)^{-1}$. Here the matrix $\mathbf{A}_{kk}(\lambda)$ is often known as the hat matrix. Using the above notations, the sub-GCV score is defined as

$$\text{GCV}_k(\lambda) = \frac{n_k^{-1}(\widehat{\boldsymbol{y}}_k - \boldsymbol{y}_k)^T \mathbf{W}_k (\widehat{\boldsymbol{y}}_k - \boldsymbol{y}_k)}{\{1 + n_k^{-1}\text{tr}\{\mathbf{A}_{kk}(\lambda)\mathbf{W}_k\}\}^2}, \tag{8}$$

where $\mathbf{W}_k = \text{diag}\{w_i, i \in S_k\}$, $k = 1, \ldots, m$. It is well known that $\text{GCV}_k(\lambda)$ enjoys nice asymptotic properties. For example, under mild conditions, Gu (2013) showed that, as $n_k \to \infty$,

$$\text{GCV}_k(\lambda) - L_k(\lambda|\boldsymbol{x}_k) - \frac{1}{n_k}\boldsymbol{\varepsilon}_k^T \mathbf{W}_k \boldsymbol{\varepsilon}_k = o_{\mathbb{P}_\varepsilon}\{L_k(\lambda|\boldsymbol{x}_k)\},$$

$k = 1, \ldots, m$. This property essentially asserts that, minimizing $\text{GCV}_k(\lambda)$ with respect to $\lambda$ is asymptotically equivalently to minimizing the local "golden criterion" $L_k(\lambda|\boldsymbol{x}_k)$.

## 3.2. Local-Optimality v.s. Global-Optimality

In this section, we explain why the use of $\text{GCV}_k(\lambda)$ in each subsample does not lead to an optimal averaged estimate $\bar{f}$. We first derive conditional risks for both $\widehat{f}_k$ and $\bar{f}$. For the former, some basic algebra yields that the conditional risk $R_k(\lambda|\boldsymbol{x}_k) = \mathbb{E}_\varepsilon\{L_k(\lambda|\boldsymbol{x}_k)\}$ is of the form

$$\begin{aligned} R_k(\lambda|\boldsymbol{x}_k) = {} & \frac{1}{n_k}\sum_{i \in S_k} w_i \text{Var}_\varepsilon\left\{\widehat{f}_k(x_i)\right\} \\ & + \frac{1}{n_k}\sum_{i \in S_k} w_i \left\{\mathbb{E}_\varepsilon \widehat{f}_k(x_i) - f_0(x_i)\right\}^2, \end{aligned} \tag{9}$$

where the expectation is taken with respect to the probability measure $\mathbb{P}_\varepsilon$. As for the latter, we first define the empirical loss function of $\bar{f}$ as

$$\bar{L}(\lambda|\boldsymbol{X}) = \frac{1}{N}\sum_{i=1}^{N} w_i\{\bar{f}(x_i) - f_0(x_i)\}^2, \tag{10}$$

where $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ denotes the collection of all covariates and $w_i \geq 0$ are the associated weights with observation $i$ such that $\sum_{i=1}^{N} w_i = N$. Similarly, the corresponding conditional risk $\bar{R}(\lambda|\boldsymbol{X}) = \mathbb{E}_\varepsilon\{\bar{L}(\lambda|\boldsymbol{X})\}$ has the following form

$$\begin{aligned} \bar{R}(\lambda|\boldsymbol{X}) = {} & \frac{1}{N}\sum_{i=1}^{N} w_i \left[\frac{1}{m}\sum_{k=1}^{m}\left\{\mathbb{E}_\varepsilon \widehat{f}_k(x_i) - f_0(x_i)\right\}\right]^2 \\ & + \frac{1}{m^2 N}\sum_{k=1}^{m}\sum_{i=1}^{N} w_i \text{Var}_\varepsilon\left\{\widehat{f}_k(x_i)\right\}. \end{aligned} \tag{11}$$

The form of (9) illustrates that, roughly speaking, a "locally optimal" choice of $\lambda$ (that minimizes (7)) tries to strike a good balance of variance and bias for each sub-estimate $\widehat{f}_k$. On the contrary, a "globally optimal" $\lambda$, which is defined to minimize (10), puts much less emphasis on the variance of $\widehat{f}_k$ (by a factor of $1/m$) than on the bias of $\widehat{f}_k$; see (11). Consequently, to obtain a "globally optimal" $\bar{f}$, one needs to intentionally choose a "smaller" $\lambda$ such that each individual function estimator $\widehat{f}_k$ overfits data set $S_k$, which leads to reduced bias $\mathbb{E}_\varepsilon \widehat{f}_k(x_i) - f_0(x_i)$ and inflated variance $\text{Var}_\varepsilon\left\{\widehat{f}_k(x_i)\right\}$. Then by taking $\bar{f} = \frac{1}{m}\sum_{j=1}^{m}\widehat{f}_j$, the variance of $\bar{f}$ can be effectively reduced by a factor of $1/m$ while keeping its bias at the same level as those of individual $\widehat{f}_j$'s. The above risk analysis confirms the heuristics in Zhang et al. (2015).

## 3.3. Distributed Generalized Cross-Validation

The discussions in Section 3.2 motivate the main result of this paper: distributed GCV score, denoted by dGCV. This data-driven tool in selecting $\lambda$ is computationally efficient for massive data as analyzed in Section 3.4.

Using the solution (6), it is straightforward to show that the predicted values of all data points $\boldsymbol{y}_l$ in the subset $S_l$

using $\widehat{f}_k$ take the form $\widehat{\boldsymbol{y}}_{kl} = \mathbf{A}_{kl}\boldsymbol{y}_k$, where $\mathbf{A}_{kl}(\lambda) = \mathbf{K}_{kl}^T(\mathbf{K}_{kk} + n_k\lambda\mathbf{I}_k)^{-1}$. Define the pooled vector of responses $\boldsymbol{Y} = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_m^T)^T$. Then the predicted value of $\boldsymbol{Y}$ using the averaged estimator $\bar{f}$ is of the form

$$\widehat{\boldsymbol{Y}} = \left(\frac{1}{m}\sum_{k=1}^{m}\widehat{\boldsymbol{y}}_{k1}^T, \ldots, \frac{1}{m}\sum_{k=1}^{m}\widehat{\boldsymbol{y}}_{km}^T\right)^T = \bar{\mathbf{A}}_m(\lambda)\boldsymbol{Y},$$

where the averaged hat matrix $\bar{\mathbf{A}}_m(\lambda)$ is defined as follows

$$\bar{\mathbf{A}}_m(\lambda) = \frac{1}{m}\begin{pmatrix} \mathbf{A}_{11}(\lambda) & \mathbf{A}_{12}(\lambda) & \cdots & \mathbf{A}_{1m}(\lambda) \\ \mathbf{A}_{21}(\lambda) & \mathbf{A}_{22}(\lambda) & \cdots & \mathbf{A}_{2m}(\lambda) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{m1}(\lambda) & \mathbf{A}_{m2}(\lambda) & \cdots & \mathbf{A}_{mm}(\lambda) \end{pmatrix}.$$

Furthermore, the global conditional risk function (11) can be conveniently re-written as

$$\begin{aligned}\bar{R}(\lambda|\boldsymbol{X}) = \frac{1}{N}\boldsymbol{F}^T\{\mathbf{I} - \bar{\mathbf{A}}_m(\lambda)\}^T\mathbf{W}\{\mathbf{I} - \bar{\mathbf{A}}_m(\lambda)\}\boldsymbol{F} \\ + \frac{\sigma^2}{N}\text{tr}\left\{\bar{\mathbf{A}}_m^T(\lambda)\mathbf{W}\bar{\mathbf{A}}_m(\lambda)\right\},\end{aligned} \tag{12}$$

where vector of true values $\boldsymbol{F} = (\boldsymbol{f}_1^T, \ldots, \boldsymbol{f}_m^T)^T$ and $\mathbf{W} = \text{diag}\{w_1, \ldots, w_N\}$. Obviously the risk function above cannot be used to select $\lambda$ in practice since the vector $\boldsymbol{F}$ is unknown. Following Gu (2013), we can define an unbiased estimator of $\bar{R}(\lambda|\boldsymbol{X}) + \sigma^2$ as follows

$$\begin{aligned}\bar{U}(\lambda|\boldsymbol{X}) = \frac{1}{N}\boldsymbol{Y}^T\{\mathbf{I} - \bar{\mathbf{A}}_m(\lambda)\}^T\mathbf{W}\{\mathbf{I} - \bar{\mathbf{A}}_m(\lambda)\}\boldsymbol{Y} \\ + \frac{2\sigma^2}{N}\text{tr}\left\{\bar{\mathbf{A}}_m(\lambda)\mathbf{W}\right\}.\end{aligned} \tag{13}$$

It is straightforward to show that $\mathbb{E}_\varepsilon\{\bar{U}(\lambda|\boldsymbol{X})\} = \bar{R}(\lambda|\boldsymbol{X}) + \sigma^2$. The above $\bar{U}(\lambda|\boldsymbol{X})$ can be viewed as an extension of the Mallow's CP (Mallows, 2000) to the divide-and-conquer scenario.

Similar to Gu (2013); Xu & Huang (2012), the Lemma 1 in Section 4 states that under some mild conditions, minimizing $\bar{U}(\lambda|\boldsymbol{X})$ and $\bar{L}(\lambda|\boldsymbol{X})$ with respect to $\lambda$ is asymptotically equivalent. In this sense, the $\lambda$ chosen by minimizing $\bar{U}(\lambda|\boldsymbol{X})$ is therefore "globally optimal." However, a major drawback of $\bar{U}(\lambda|\boldsymbol{X})$ is that it utilizes the knowledge of $\sigma^2$, which in practice often needs to be estimated. To overcome this, we propose the following modification of the GCV score

$$\text{dGCV}(\lambda|\boldsymbol{X}) = \frac{\frac{1}{N}\sum_{i=1}^{N} w_i\left\{y_i - \bar{f}(x_i)\right\}^2}{\left[1 - \frac{1}{Nm}\sum_{k=1}^{m}\text{tr}\{\mathbf{A}_{kk}(\lambda)\mathbf{W}_k\}\right]^2}, \tag{14}$$

where $\mathbf{W}_k = \text{diag}\{w_i, i \in S_k\}$. Intuitively, consider $\tilde{\sigma}^2 = N^{-1}\sum_{i=1}^{N} w_i\left\{y_i - \bar{f}(x_i)\right\}^2$ as an estimator of $\sigma^2$ and use the fact that $(1-x)^{-2} \approx 1 + 2x$ as $x \to 0$, the $\bar{U}(\lambda|\boldsymbol{X})$ defined in (13) essentially can be viewed as the first order Taylor expansion of the dGCV$(\lambda|\boldsymbol{X})$. However, in the definition of dGCV$(\lambda|\boldsymbol{X})$, it does not require any information of $\sigma^2$. Note that dGCV incorporates information across all sub-samples, which explains its superior empirical performance. In fact, Theorem 1 in Section 4 shows that under some conditions, minimizing dGCV$(\lambda|\boldsymbol{X})$ and the "golden criterion" $\bar{L}(\lambda|\boldsymbol{X})$ with respect to $\lambda$ are also asymptotically equivalent.

### 3.4. Computational Complexity of dGCV

The computation of dGCV$(\lambda|\boldsymbol{X})$ in (14) for a given $\lambda$ consists of two parts: the first part involves computing the trace of individual hat matrices, tr$\{\mathbf{A}_{kk}(\lambda)\mathbf{W}_k\}$, $k = 1, \ldots, m$, which requires $O(N^3/m^2)$ floating operations and a memory usage of $O(N^2/m^2)$; the second part is to evaluate the predicted value of $\bar{f}(x_i)$ for which $w_i \neq 0$, which costs $O(NN_w)$ floating operations and a memory usage of $O(N)$, where $N_w$ denotes the number of nonzero $w_i$'s. Hence, the total computation cost of dGCV$(\lambda|\boldsymbol{X})$ is of the order $O(N^3/m^2 + NN_w)$. In most applications, the number of folds $m$ generally cannot exceed $\sqrt{N}$ for $\bar{f}$ to reach the optimal convergence rate (Zhang et al., 2015). In such cases, one can simply use $w_1 = \cdots = w_N = 1$, which results in the computational cost of the order $O(N^3/m^2)$ for one evaluation of dGCV$(\lambda|\boldsymbol{X})$. This is the same as that of the divide-and-conquer algorithm proposed in Zhang et al. (2015).

In some applications where $m$ is much larger than $\sqrt{N}$, the computational cost of dGCV$(\lambda|\boldsymbol{X})$ becomes $O(NN_w)$. In this case, we may want to only choose $m^*$ out of $m$ sub-data sets for saving computational costs. To achieve that, we need to choose weights $w_i$'s properly. For example, we can set $w_i = N/(\sum_{k=1}^{m^*} n_k)$ if $i \in \cup_{k=1}^{m^*} S_k$ and $w_i = 0$ otherwise. Under this setting, the dGCV$(\lambda|\boldsymbol{X})$ in (14) becomes

$$\text{dGCV}^*(\lambda|\boldsymbol{X}) = \frac{\frac{1}{N_{m^*}}\sum_{i\in\cup_{k=1}^{m^*}S_k}\left\{y_i - \bar{f}(x_i)\right\}^2}{\left[1 - \frac{1}{mN_{m^*}}\sum_{k=1}^{m^*}\text{tr}\{\mathbf{A}_{kk}(\lambda)\}\right]^2}, \tag{15}$$

where $N_{m^*} = n_1 + \cdots + n_{m^*}$. Using (15) instead of (14), we only need to evaluate $\bar{f}(x_i)$ for $x_i$'s in $m^*$ subsets and the computation time is reduced to $O(N^2m^*/m)$. We applied (15) to the Million Song Data set considered in Section 6, which yields good results in both prediction and computation time.

Optimization of dGCV$(\lambda|\boldsymbol{X})$ or dGCV$^*(\lambda|\boldsymbol{X})$ can be carried out using a simple one-dimensional grid search. Since the first and second derivatives of dGCV$(\lambda|\boldsymbol{X})$ or dGCV$^*(\lambda|\boldsymbol{X})$ can be easily computed using similar arguments in Wood (2004); Xu & Huang (2012), it can also be optimized using the Newton-Raphson algorithm with the same computational costs.

**Remark 1.** *We want to mention that dGCV$(\lambda|\boldsymbol{X})$ can also be used to choose other tuning parameters in the kernel function. For example, if the Gaussian kernel $K(x,z) = \exp(-\|x-z\|_2^2/\phi)$ is used, dGCV is also a function of the bandwidth parameter $\phi$, and thus can be used to choose the optimal $\phi$ as well.*

## 4. Asymptotic Properties

In this section, we will show that the proposed dGCV criterion in (14) is "globally optimal" under some conditions. We first introduce some notation. Denote $\mathbb{P}_X$, $\mathbb{P}_\varepsilon$, $\mathbb{P}_{\varepsilon,X}$ as the probability measures of covariate $X$, error process $\varepsilon$ and their joint probability measure. Similarly, $\mathbb{E}_\varepsilon$ and $\mathrm{Var}_\varepsilon$ denote the expectation and variance under the probability measure $\mathbb{P}_\varepsilon$. Let $\lambda_{\max}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$ and $\mathrm{tr}(\mathbf{A})$ be the largest eigenvalue and the largest singular value of the matrix $\mathbf{A}$, respectively. We use $\xrightarrow{\mathbb{P}}$ to denote the convergence in probability measure $\mathbb{P}$ and $O_{\mathbb{P}}(\cdot)$, $o_{\mathbb{P}}(\cdot)$ as defined in the conventional way. For any function $f(x) : \mathcal{X} \to \mathbb{R}$, let $\|f\|_{\sup} = \sup_{x\in\mathcal{X}} |f(x)|$ and $\mathbb{P}f = \int_{\mathcal{X}} f(x)\, d\mathbb{P}$. Finally, let $\mathbb{P}_n$ denote the empirical probability measure based on i.i.d samples of size $n$ from the probability measure $\mathbb{P}$.

### 4.1. Asymptotic Optimality of dGCV

The following regularity conditions are needed to show the optimality of dGCV.

**[C1]** $\frac{1}{m}\sum_{l=1}^m \lambda_{\max}\left\{(\mathbf{K}_{ll} + n_l\lambda\mathbf{I}_l)^{-2}\left(\frac{1}{m}\sum_{k=1}^m \mathbf{K}_{kl}^T\mathbf{K}_{kl}\right)\right\} = O_{\mathbb{P}_X}(1)$;

**[C2]** $N\bar{R}(\lambda|\boldsymbol{X}) \xrightarrow{\mathbb{P}_X} \infty$ as $N \to \infty$;

**[C3]** (a) $\max_{1\le i\le N} w_i \le W$ for some constant $W > 0$; (b) $\frac{1}{Nm}\sum_{k=1}^m \mathrm{tr}\{\mathbf{A}_{kk}(\lambda)\} = o_{\mathbb{P}_X}(1)$.

**[C4]** $\frac{[N^{-1}\mathbf{tr}\{\bar{\mathbf{A}}_m(\lambda)\mathbf{W}\}]^2}{[N^{-1}\mathbf{tr}\{\bar{\mathbf{A}}_m^T(\lambda)\mathbf{W}\bar{\mathbf{A}}_m(\lambda)\}]} = o_{\mathbb{P}_X}(1)$.

Intuitively, condition C1 requires that some similarities among sub-data sets. If all $\mathbf{K}_{kl}$'s are similar to the the matrix $\mathbf{K}_{ll}$, we can expect $\lambda_{\max}\left\{(\mathbf{K}_{ll} + n_l\lambda\mathbf{I}_l)^{-2}\left(\frac{1}{m}\sum_{k=1}^m \mathbf{K}_{kl}^T\mathbf{K}_{kl}\right)\right\} \le 1$, in which case C1 holds. In the journal version of this paper, we shall show that one sufficient condition for C1 to hold is to ensure that the "maximal marginal degrees of freedom" $d = N\max_{1\le i\le N} \mathrm{diag}\{\mathbf{K}(\mathbf{K}+N\lambda\mathbf{I}_N)^{-1}\}$ (Bach, 2013) is sufficiently small compared to $N/m$. Condition C2 is a widely used condition to ensure the optimality of the GCV to hold, for example, see Craven & Wahba (1978); Li (1986); Gu & Ma (2005); Xu & Huang (2012). It is a mild condition for nonparametric regression problems, where the parametric rate $O(N^{-1})$ is unattainable for the estimation risk. For example, for kernel ridge regression models with polynomially or exponentially decaying kernel functions, condition C2 holds (Zhang et al., 2015). However, it does raise a flag for the application of the dGCV when a finite rank kernel is used, in which case the optimal rate of $\bar{R}(\lambda|\boldsymbol{X})$ is of the order $O(N^{-1})$ (Zhang et al., 2015). Nevertheless, without condition C2, it is questionable whether there exists an asymptotically optimal selection procedure for the tuning parameter $\lambda$ (Li, 1986). It turns out that, under conditions C1-C2, $\bar{U}(\lambda|\boldsymbol{X})$ defined in (13) is "globally optimal."

**Lemma 1.** *Under Conditions C1–C2, for a fixed $\lambda$, we have that*

$$\bar{U}(\lambda|\boldsymbol{X}) - \bar{L}(\lambda|\boldsymbol{X}) - \frac{1}{N}\boldsymbol{\varepsilon}^T\mathbf{W}\boldsymbol{\varepsilon} = o_{\mathbb{P}_{\varepsilon,X}}\{\bar{L}(\lambda|\boldsymbol{X})\}. \quad (16)$$

The key of the proof this Lemma is to carefully show that the variance of the difference in the left-hand side of equation (16) is of the order $o\{\bar{R}^2(\lambda|\boldsymbol{X})\}$ and the difference $\bar{L}(\lambda|\boldsymbol{X}) - \bar{R}(\lambda|\boldsymbol{X})$ is of the order $o_{\mathbb{P}_{\varepsilon,X}}\{\bar{R}^2(\lambda|\boldsymbol{X})\}$, both of which require condition C2. The proof is similar to that of Theorem 2.1 in Xu & Huang (2012).

Lemma 1 states that when $\sigma^2$ is known, minimizing $\bar{U}(\lambda|\boldsymbol{X})$ with respect to $\lambda$ is asymptotically equivalent to minimizing the empirical true loss function $\bar{L}(\lambda|\boldsymbol{X})$. However, it is rarely the case that one has complete knowledge of $\sigma^2$. In this sense, the proposed dGCV is more practical and it can be shown to be "globally optimal" as well, under some additional conditions.

**Theorem 1.** *Under Conditions C1–C4, for a fixed $\lambda$, we have that*

$$\mathrm{dGCV}(\lambda|\mathbf{X}) - \bar{L}(\lambda|\mathbf{X}) - \frac{1}{N}\boldsymbol{\varepsilon}^T\mathbf{W}\boldsymbol{\varepsilon} = o_{\mathbb{P}_{\varepsilon,X}}\{(\bar{L}(\lambda|\mathbf{x})\}. \quad (17)$$

The proof of Theorem (17) makes use of the similarity between $\bar{U}(\lambda|\boldsymbol{X})$ and $\mathrm{dGCV}(\lambda|\mathbf{X})$, where the former is very close to the first order Taylor expansion of the latter using the fact that $(1-x)^{-2} = 1 + 2x + 3(1-x^*)^{-4}x^2$ for some $x^* \in (0, x)$. By carefully bounding the difference between $\bar{U}(\lambda|\boldsymbol{X})$ and $\mathrm{dGCV}(\lambda|\mathbf{X})$ under conditions C1-C4 and invoking Lemma 1, we can prove Theorem 1.

Similar to Lemma 1, Theorem 1 shows that minimizing $\mathrm{dGCV}(\lambda|\boldsymbol{X})$ amounts to minimizing the true conditional loss function $\bar{L}(\lambda|\boldsymbol{X})$, although additional conditions C3-C4 are needed. Condition C3 is pretty mild in that it essentially requires that sufficient number of $w_i$'s are nonzero and the effective number of parameters to be negligible compared to the sample size, which is typically true for non-parametric function estimators in most settings of interests. In addition, C3 becomes trivial when $m \to \infty$ because by definition we have that $\mathrm{tr}\{\mathbf{A}_{kk}(\lambda)\} \le n_k$, $k = 1, \ldots, m$. When the entire data set is used at once $(m = 1)$, condition C4 reduces to the well known condition

$[N^{-1}\mathrm{tr}\{\mathbf{A}(\lambda)\}]^2/[N^{-1}\mathrm{tr}\{\mathbf{A}^2(\lambda)\}] = o(1)$ in the literature (Craven & Wahba, 1978; Li, 1986; Gu & Ma, 2005; Xu & Huang, 2012). For example, for smoothing splines, we typically have $\mathrm{tr}\{\mathbf{A}(\lambda)\} = O(\lambda^{-1/s})$ and $\mathrm{tr}\{\mathbf{A}^2(\lambda)\} \asymp O(\lambda^{-1/s})$ for some $s > 1$. Then as long as $\lambda^{-1/s}/N \to 0$, which covers the most region of practical interests for $\lambda$, we have that $[N^{-1}\mathrm{tr}\{\mathbf{A}(\lambda)\}]^2/[N^{-1}\mathrm{tr}\{\mathbf{A}^2(\lambda)\}] \to 0$ as $N \to \infty$. Condition C4 can be viewed as an extension of this commonly used condition to the divide-and-conquer regime.

**Remark 2.** *Theoretical proof of whether or not the function estimator $\bar{f}$ coupled the optimal $\widehat{\lambda} = \arg_{\min} \mathrm{dGCV}(\lambda|\mathbf{X})$ truly achieves the minimax rate in the literature (e.g. Zhang et al., 2015) is difficult because $\widehat{\lambda}$ is a random quantity depending on various factors. In practice, one way to ensure minimax rate of $\bar{f}$ is to search for the optimal $\lambda$ by minimizing $\mathrm{dGCV}(\lambda|\mathbf{X})$ within a region guided by the theoretically optimal rate of $\lambda$ in the literature for various kernels. For example, in Section 6, the optimal rate of $\lambda$ is $O(1/N)$ for the Gaussin kernel (Zhang et al., 2015) and we therefore searched for best $\lambda$ within the set* $\{0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}/N$.

**Remark 3.** *One benefit of using high level conditions such as C1, C2 and C4 is that they do not involve the response variable and can be computed efficiently using sample data. To deal with the randomness in covariate X, one can bootstrap/resample/subsample from the observed data, which is especially suitable when the sample size under consideration is extremely large. Through this resampling strategy, one can empirically verify C1, C2 and C4, although rigorous justification of such strategy has not been established and will be an interesting topic for future research.*

## 5. Simulation Studies

In this section, we conduct a simulation study to illustrate the effectiveness of $\mathrm{dGCV}(\lambda)$ in choosing the optimal $\lambda$ for the divide-and-conquer function estimator. The data were simulated from the model $y = f_0(x) + \varepsilon$, where $f_0(x) = 2|x - 1/2|$ for $x \in [0, 1]$ and $\varepsilon \sim N(0, 0.5^2)$. The covariate $x_i$'s were independently generated from the uniform distribution over the interval $[0, 1]$. For each simulation run, we first generated a data set of the size $N = mn$ and then randomly partition the data sets into $m$ sub-data sets of equal sizes. The divide-and-conquer estimator $\bar{f}$ was obtained as given in (3). The true function $f_0(x)$ belongs to the Sobolev space of Lipschitz functions on $[0, 1]$, hence we used the reproducing kernel $K(x, z) = 1 + \min(x, z)$ and the associated norm $\|f\|^2_{\mathcal{H}} = f^2(0) + \int_0^1 \{f'(x)\}^2\, dx$.

In all simulation runs, the tuning parameter $\lambda$ was selected by a grid search for $\log(\lambda)$ over 30 equally-spaced grid points over the interval $[-12, 1]$. Three approaches were used for selection of $\lambda$: (i) the distributed GCV (dGCV) proposed in (14); (ii) the naive GCV applied to each sub-dataset (nGCV) which applies the GCV defined in (8) to choose the best $\lambda$ for each individual $\widehat{f}_k$ and then average them using (3); and (iii) the true conditional loss function (TrueLoss) $\bar{L}(\lambda|\boldsymbol{X})$ defined in (10). For all three approaches, we set the weights $w_i = 1$ for all $i = 1, \dots, N$. The last approach is not practically feasible since it requires the knowledge of the true function $f_0$. Rather, it is served as the "golden criterion" to show the effectiveness of the other two approaches. Summary statistics based on 100 simulation runs were illustrated in Figure 1(a)-(f).

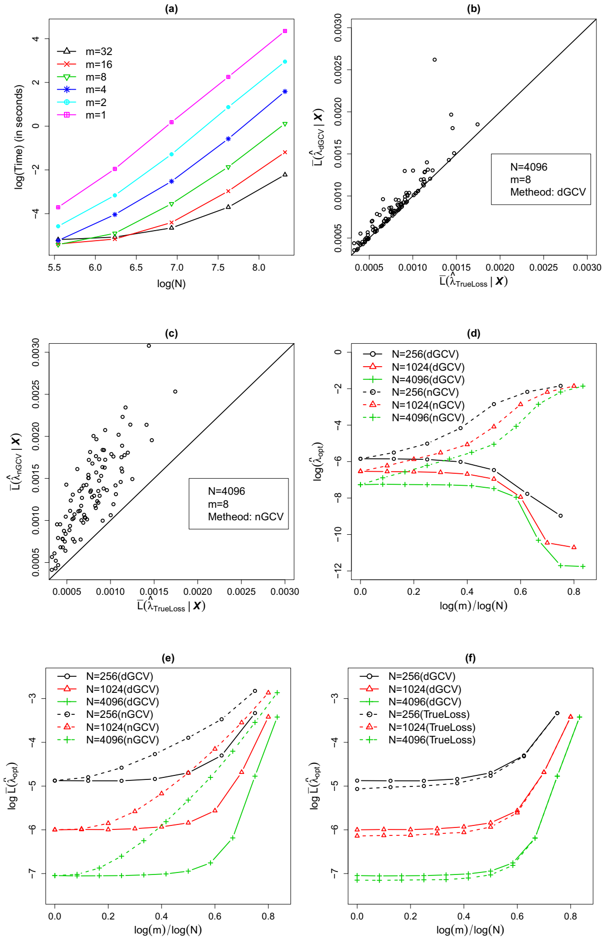

*Figure 1.* (a) the logarithm of computational time (in seconds) v.s. $\log(N)$; (b)-(c): scatter plots of true empirical losses of function estimators; (d) the logarithm of averages of selected $\lambda$ v.s. $\log(m)/\log(N)$; (e)-(f): the logarithm of true empirical losses v.s. $\log(m)/\log(N)$. Note that in (d)-(f), $\hat{\lambda}_{\mathrm{opt}}$ in the y-axis denotes one of $\hat{\lambda}_{\mathrm{dGCV}}$, $\hat{\lambda}_{\mathrm{nGCV}}$ and $\hat{\lambda}_{\mathrm{TrueLoss}}$ for each curve.

Figure 1(a) illustrates the computational complexity of one evaluation of $\mathrm{dGCV}(\lambda)$ for $N = 2^i$, $i = 8, 9, 10, 11, 12$ and $m = 1, 2, 4, 8, 16, 32$. All simulation runs were carried out in the software R on a cluster of 100 Linux machines with a total of 100 CPU cores, with each core running at approximately 2 GFLOPS. We can clearly see that by using the divide-and-conquer strategy, the computational time of the dGCV can be greatly reduced compared to the case when all data were used at once (when $m = 1$).

In Figure 1(b)-(c), we give some comparisons of the dGCV method and the nGCV method. Figure 1(b) shows the scatter plot of true empirical losses, as defined in (10), of the function estimators obtained by minimizing dGCV($\lambda$) versus minimizing the unattainable "golden criterion" (10) over 100 simulation runs. As we can see, majority of points are concentrated around the $45^o$ straight line, which supports our theoretical findings in Theorem 1. On the contrary, Figure 1(c) shows that true empirical losses of the function estimator based on the nGCV approach are generally larger than the minimum possible true losses, indicating that such function estimators are indeed only "locally" optimal but not "globally optimal."

In Figure 1(d)-(f), we used $N = 2^i$ and $m = 2^j$ for $j = 0, 1, \ldots, i-2$ and $i = 8, 10, 12$ so that there were at least four data points in each sub-data set. To better understand the differences between the distributed GCV and the naive GCV approaches, Figure 1(d) shows how the logarithm of the averages of selected tuning parameters (over 100 simulation runs), denoted as $\log(\widehat{\lambda}_{opt})$, for each method changes as $m$ increases. As we can see, when $m = 1$ they are identical. However, as $m$ increases, the $\lambda$ selected by the naive GCV approach consistently increases whereas the $\lambda$ selected by the dGCV method stays about the same until $m$ gets really large and is always smaller than the $\lambda$ selected by the nGCV method. This is consistent with findings in Zhang et al. (2015) where they argue that the locally optimal rate of $\lambda$ for each individual $\widehat{f}_k$ is of the order $O(n^{-2/3})$ with $n = N/m$ whereas the globally optimal rate for $\lambda$ is of the order $O(N^{-2/3})$.

The y-axis of Figure 1(e)-(f) is the logarithm of estimation errors $\log \overline{L}(\widehat{\lambda}_{opt})$, where $\overline{L}(\widehat{\lambda}_{opt})$ stands for the averaged true conditional loss defined in (10) over 100 simulation runs using different selection approaches for $\lambda$. We can see from Figure 1(e)-(f) that as long as $m$ is not too large compare to $N$, the proposed dGCV($\lambda$) is quite robust in terms of controlling the estimation error as $m$ grows and is almost identical to that of using the true loss function, which is considered as a "golden criterion." This is consistent with our Theorem 1. In contrast, estimation errors of the nGCV approach quickly inflates as $m$ increases, which is expected according to our discussion in subsection 3.2. Finally, it is interesting to point out that as the $\lambda$ selected by the dGCV method starts to drop in Figure 1(d), the estimation errors in Figure 1(e)-(f) start to inflate as well.

## 6. The Million Song Dataset

In this section, we applied the dGCV$^*$ tuning method to the Million Song Dataset, which consist of $463,715$ training examples and $51,630$ testing examples. Each observation is a song track released between the year 1922 and 2011. The response variable $y_i$ is the year when the song is released and the covariate $x_i$ is a 90-dimensional vector, consists of timbre information of the song. We refer to Bertin-Mahieux et al. (2011) for more details on this data set. The goal is to use the timbre information of the song to predict the year when the song was released using the kernel ridge regression. The same dataset has been analyzed by Zhang et al. (2015), but without addressing the issue of selecting an optimal tuning parameter. Our dGCV$^*$ method demonstrated significant empirical advantages over theirs.

Following Zhang et al. (2015), the feature vectors were normalized so that they have mean $0$ and standard deviation $1$ and the Gaussian kernel function $K(x, z) = \exp(-\|x - z\|_2^2/\phi)$ was used for the kernel ridge regression. Seven partitions $m \in \{32, 38, 48, 64, 96, 128, 256\}$ were used for the divide-and-conquer kernel ridge regression. Aside from the penalty parameter $\lambda$ in (2), the bandwidth $\phi$ is also known to have important impact on the prediction accuracy. To find the best combination of $(\lambda, \phi)$ for each partition $m$, we perform a 2-dimensional search with $\lambda \in \{0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}/N$ and $\phi \in \{2, 3, 4, 5, 6, 7\}$ by minimizing (15) with $K = \lceil m/10 \rceil$, where $\lceil a \rceil$ is the smallest integer that is greater than $a$. Note that in this case, dGCV$^*(\lambda|\boldsymbol{X})$ is also a function of $\phi$. The experiment was conducted in Matlab using a Windows computer with 16GB of memory and a single-threaded 3.5Ghz CPU. For each $(\lambda, \phi)$ pair, the computation time for (15) are $1,058$s ($m = 32$), 840s ($m = 38$), 577s ($m = 48$), 476s ($m = 96$), 453s ($m = 128$) and 457s ($m = 256$), which are reasonable for a data set with almost half-million observations.

The grid search gave the optimal choice of $\lambda = 0.5/N$ and $\phi = 3$ for most of case scenarios. From Figure 2(a)-(b), we can see that the choice of the bandwidth parameter $\phi$ has a great impacts on the dGCV score as well as the penalty parameter $\lambda$. It seems that the latter provides some additional small adjustments after a good value of $\phi$ is chosen.

In Zhang et al. (2015), the authors used a fixed value $\lambda = 1/N$ and a $\phi = 6$ chosen by the cross-validation for their kernel ridge regression model. In Figure 2(c), we can see that such a choice leads to a much worse prediction mean squared error (PMSE) on the testing samples. Using the proposed dGCV criterion, our choice of $\lambda$ and $\phi$ yields almost identical prediction accuracy as the minimum possible PMSE on the testing samples obtained over all 36 grid points.

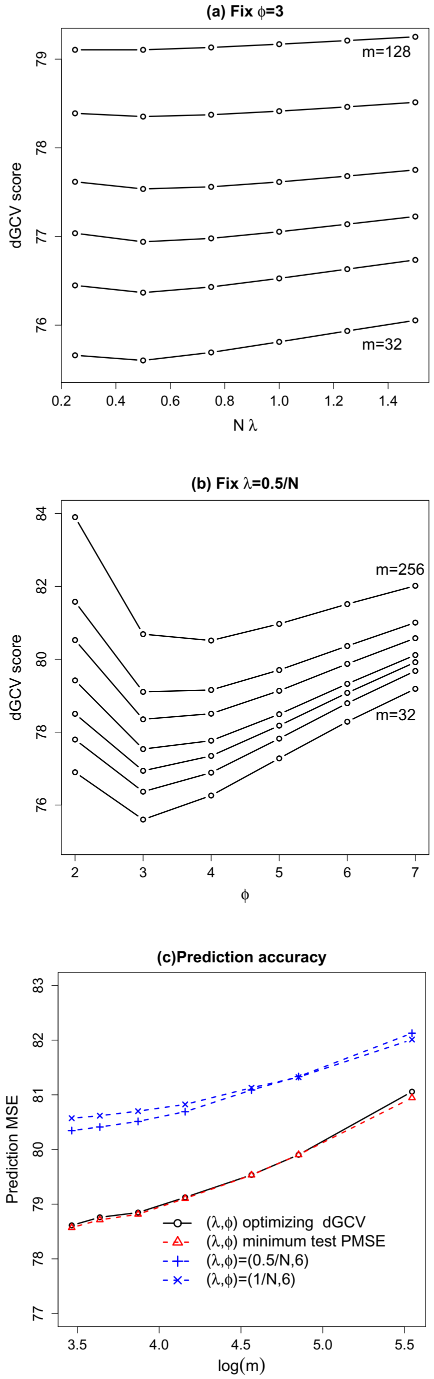

*Figure 2.* (a) dGCV score v.s. $N\lambda$ with $m = 32$ (the bottommost) to $m = 128$ (the uppermost); (b) dGCV score v.s. $\phi$ with $m = 32$ (the bottommost) to $m = 256$ (the uppermost); (c) The prediction mean squared errors on the testing samples v.s. $\log(m)$.

## 7. Conclusion

In this paper, we proposed a data-driven criterion named dGCV that can be used to empirically selecting the critical tuning parameter $\lambda$ for divide-and-conquer kernel ridge regression. Not only the proposed approach is computationally scalable even for massive data sets, we have also theoretically shown that it is asymptotically optimal in the sense that minimizing dGCV is equivalent to minimizing the true global conditional empirical loss, extending the existing optimality results of GCV to the divide-and-conquer framework.

In the future work, we plan to improve the current high-level conditions listed in C1-C4 to much lower level conditions that can be readily checked theoretically for various types of kernel functions as investigated in Zhang et al. (2015). We conjecture that this is at least possible for polynomially decaying kernel functions based on the rich literature on their design and hat matrices. Furthermore, so far we have presumed a fixed $m$. Another part of the future work is to investigate the growth rate of $m$ for some specific kernels under which Theorem 1 still holds, following the framework proposed in Shang & Cheng (2017).

### Acknowledgements

Ganggang Xu's research is partially supported by Collaboration Grants for Mathematicians (Award ID: 524205) from Simons Foundation and NSF Award SES-1758605. Zuofeng Shang's research is supported by NSF Award DMS-1764280. Guang Cheng's research is partially supported by NSF CAREER Award DMS-1151692, DMS-1418042, DMS-1712907, DMS-1811812 and Office of Naval Research (ONR N00014-15-1-2331, ONR N00014-18-1-2759).

### References

## References

Bach, F. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pp. 185–209, 2013.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. The million song dataset. In *Ismir*, volume 2, pp. 10, 2011.

Blanchard, G. and Mücke, N. Parallelizing spectral algorithms for kernel learning. *arXiv preprint arXiv:1610.07487*, 2016.

Chang, X., Lin, S.-B., and Zhou, D.-X. Distributed semi-supervised learning with kernel ridge regression. *Journal of Machine Learning Research*, 18(46):1–22, 2017.

Chen, X. and Xie, M. A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, 24(4):1655–1684, 2014.

Craven, P. and Wahba, G. Smoothing noisy data with spline functions. *Numerische mathematik*, 31(4):377–403, 1978.

Gu, C. *Smoothing spline ANOVA models*, volume 297. Springer Science & Business Media, 2013.

Gu, C. and Ma, P. Optimal smoothing in nonparametric mixed-effect models. *The Annals of Statistics*, 33(3): 1357–1379, 2005.

Guo, Z.-C., Shi, L., and Wu, Q. Learning theory of distributed regression with bias corrected regularization kernel network. *Journal of Machine Learning Research*, 18 (118):1–25, 2017.

Li, K.-C. Asymptotic optimality of cl and generalized cross-validation in ridge regression with application to spline smoothing. *The Annals of Statistics*, 14(3):1101–1112, 1986.

Lin, S.-B., Guo, X., and Zhou, D.-X. Distributed learning with regularized least squares. *The Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Lu, J., Cheng, G., and Liu, H. Nonparametric heterogeneity testing for massive data. *arXiv preprint arXiv:1601.06212*, 2016.

Mallows, C. L. Some comments on cp. *Technometrics*, 42 (1):87–94, 2000.

Shang, Z. and Cheng, G. Computational limits of a distributed algorithm for smoothing spline. *The Journal of Machine Learning Research*, 18(1):3809–3845, 2017.

Shawe-Taylor, J. and Cristianini, N. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Wahba, G. *Spline models for observational data*, volume 59. Siam, 1990.

Wood, S. N. Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686, 2004.

Xiang, D. and Wahba, G. A generalized approximate cross validation for smoothing splines with non-gaussian data. *Statistica Sinica*, 6(3):675–692, 1996.

Xu, G. and Huang, J. Z. Asymptotic optimality and efficient computation of the leave-subject-out cross-validation. *The Annals of Statistics*, 40(6):3003–3030, 2012.

Zhang, Y., Duchi, J., and Wainwright, M. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

Zhao, T., Cheng, G., and Liu, H. A partially linear framework for massive heterogeneous data. *Annals of statistics*, 44(4):1400, 2016.