Divide-and-Conquer Learning with Nyström: Optimal Rate and Algorithm

Rong Yin[1,2], Yong Liu[1,2*], Lijing Lu[1,2], Weiping Wang[1], Dan Meng[1]
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China[1]
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China[2]
{yinrong, liuyong, lulijing, wangweiping, mengdan}@iie.ac.cn

Abstract

Kernel Regularized Least Squares (KRLS) is a fundamental learner in machine learning. However, due to the high time and space requirements, it has no capability to large scale scenarios. Therefore, we propose DC-NY, a novel algorithm that combines divide-and-conquer method, Nyström, conjugate gradient, and preconditioning to scale up KRLS, has the same accuracy of exact KRLS and the minimum time and space complexity compared to the state-of-the-art approximate KRLS estimates. We present a theoretical analysis of DC-NY, including a novel error decomposition with the optimal statistical accuracy guarantees. Extensive experimental results on several real-world large-scale datasets containing up to 1M data points show that DC-NY significantly outperforms the state-of-the-art approximate KRLS estimates.

Introduction

In nonparametric statistical learning, kernel methods (Liu et al. 2018; Li et al. 2018; Liu and Liao 2014; Ding et al. 2018) have made remarkable achievements for kernel regularized least squares (KRLS) (Liu et al. 2017; Taylor and Cristianini 2004; Yin et al. 2019; Liu and Liao 2015; Liu, Jiang, and Liao 2014) by projecting data into high-dimensional space. Unfortunately, due to high time and memory consumption, typically at least quadratic in the number of examples, KRLS is unfeasible to deal with large-scale learning despite excellent theoretical guarantee.

To address the problems, a variety of practical approximate approaches have been designed to avoid the costly expense of finding an exact minimizer: (1) Iterative optimization (Lo et al. 2008). It provides regularization against over-fitting and improves computational efficiency by limited and small iterations. The representative include gradient descent (Carratino, Rudi, and Rosasco 2018; Lin and Cevher 2018), preconditioned conjugate gradient (Fasshauer and Mccourt 2012; Yang, Pilanci, and Wainwright 2015; Gonen, Orabona, and Shalev-Shwartz 2016; Ma and Belkin 2017), and accelerated extensions (Raskutti, Wainwright, and Yu 2014; Cutajar et al. 2016; Bo et al. 2014); (2) Random projections (Williams and Seeger 2001; Smola 2000). It reduces dimensions of data to reduce the cost of matrix multiplication. Classical examples include Nyström (Rudi, Carratino, and Rosasco 2017; Tu et al. 2016) and random features (Rudi, Camoriano, and Rosasco 2016; Rahimi and Recht 2007); (3) Distributed learning (Guo, Lin, and Shi 2017). It computes KRLS in parallel by dividing into some subsets and then merge the result from each subset to get the final approximation. Recently, combinations of those accelerated algorithms have also captured a lot of attention, of which learning properties have been explored including the combination of divide-and-conquer and SGD (Lin and Cevher 2018) and the combination of divide-and-conquer and random features (Li, Liu, and Wang 2019). Even though the state-of-the-art KRLS estimates can preserve the same optimal statistical accuracy of exact KRLS, the computational requirements of them are still prohibitive faced with large-scale datasets, namely, there are no corresponding computational lower bounds.

In this paper, we investigate the algorithm of combining divide-and-conquer, Nyström, conjugate gradient and preconditioning to deal with extremely large-scale applications, which achieves the same accuracy of exact kernel regularized least squares with only a fraction of computations. Complexity analysis shows that the proposed algorithm solve KRLS with $\max(\frac{Nm}{p}, m^3)$ time and $\frac{Nm}{p}$ space, where N is the number of data points, m is the sampling scale, and p is the number of partitions. Our theoretical analysis derives optimal statistical rates in a basic setting. Under benign conditions, the regularization parameter $\lambda \asymp 1/\sqrt{N}$, the proposed algorithm can reach the optimal convergence rate $1/N$. Most importantly, extensive experimental results on large-scale datasets containing up to 1M data points show that the proposed algorithm can process millions of data points in just several seconds and has absolute advantage over the state-of-the-art approximate KRLS in terms of efficiency and accuracy. To the best of our knowledge, it is the first time utilizing these approximate methods and achieving fruitful results.

The reminder of the paper is organized as below: Section 2 introduces the related work. Section 3 states the background and the proposed approximate KRLS estimator. The corresponding theoretical assessments follows in section 4. Finally, we present the experiments and conclusions. The detail of proof and other information are shown in Appendix.

*Corresponding authors: Yong Liu

Table 1: Computational complexity of the classical approximation algorithms in KRLS estimates under the condition of the same statistical accuracy. The second, third, fourth, fifth and sixth columns correspond to the training time complexity, space complexity, the number of partitions p, sampling scale m, and the value of r and γ, respectively. N denotes the number of training data, where $N/p > m$.

Algorithms	Time	Space	p	m	r, γ
KRLS (Caponnetto and Vito 2007)	N^3	N^2	/	/	/
Iterative (Avron, Clarkson, and Woodruff 2017)	N^2	N^2	/	/	/
Random Features (Rudi, Camoriano, and Rosasco 2016)	Nm^2	Nm	/	$N^{\frac{(2r-1)\gamma+1}{2r+\gamma}}$	[1/2,1], [0,1]
Nyström (Rudi, Camoriano, and Rosasco 2015)	Nm^2	Nm	/	$N^{\frac{1}{2r+\gamma}}$	[1/2,1], (0,1]
Nyström-iterative (Rudi, Carratino, and Rosasco 2017)	$Nm + m^3$	Nm	/	$N^{\frac{1}{2r+\gamma}}$	[1/2,1], (0,1]
DC (Guo, Lin, and Shi 2017)	$\frac{N^3}{p^3}$	$\frac{N^2}{p^2}$	$N^{\frac{2r-1}{2r+\gamma}}$	/	[1/2,1], (0,1]
DC-Random Features (Li, Liu, and Wang 2019)	$\frac{Nm^2}{p}$	$\frac{Nm}{p}$	$N^{\frac{2r-1}{2r+\gamma}}$	$N^{\frac{(2r-1)\gamma+1}{2r+\gamma}}$	[1/2,1], [0,1]
DC-NY(This paper)	$\max(\frac{Nm}{p}, m^3)$	$\frac{Nm}{p}$	$N^{\frac{2r-1}{2r+\gamma}}$	$N^{\frac{1}{2r+\gamma}}$	(1/2,1], (0,1]

Related Work

To overcome the computational and memory bottleneck of KRLS, practical algorithms are developed, including Nyström approach (Rudi, Carratino, and Rosasco 2017; Camoriano et al. 2016) and divide-and-conquer (Zhang, Duchi, and Wainwright 2013; Li, Liu, and Wang 2019) of which statistical properties are well studied. Nyström (Rudi, Camoriano, and Rosasco 2015; Camoriano et al. 2016) tactfully constructs some small-scale matrices, by sampling the dataset, to approximate the raw kernel matrix so that the time and space complexity can make a sudden drop. The typical Nyström method FALKON, proposed in (Rudi, Carratino, and Rosasco 2017), combines Nyström and preconditioned conjugate gradient (PCG) to obtain the optimal statistical accuracy. Divide-and-conquer methods divide the dataset into several small blocks so as to reduce the number of data in one processor. The representational method (Guo, Lin, and Shi 2017) utilizes divide-and-conquer to obtain a substantial reduction in computation time.

Further, our work combines divide-and-conquer and Nyström to approximate KRLS, and accelerates the solution with PCG so as to obtain high computation gains and sound statistical guarantees. It is a non-trivial extension of these approximate approaches with technical challenges in algorithm design and theoretical analysis. Based on a novel partitioning, clever scaling, and the standard integral operator framework, the error is bounded tightly to obtain the optimal statistical performance. Table 1 shows the detail time and space complexity of the state-of-the-art KRLS estimators with the optimal statistical accuracy. Evidently, a substantial step in provably reducing the time complexity is taken by us when we have the same statistical accuracy. In detail, compared to the state-of-the-art approximate KRLS estimates, we reduce the time complexity at least by a factor of $\min(m, \frac{N}{p}\frac{1}{m})$, where the number of data points in each processor $\frac{N}{p}$ is bigger than the sampling scale m. Considering the concrete values of m and p this leads to a computational cost for optimal generalization, we reduce the space complexity by a factor of $N^{\frac{(2r-1)\gamma}{2r+\gamma}}$ compared to the state-of-the-art KRLS estimate, where $\frac{(2r-1)\gamma}{2r+\gamma} > 0$.

Statistical and Computational Trade-offs in KRLS

Let ρ be a probability measure on $\mathbf{X} \times \mathbb{R}$, which is fixed but unknown and where, $\mathbf{X}$ and $\mathbb{R}$ are the input and output spaces. Data $(x_i, y_i)_{i=1}^N$ are sampled identically and independently from $\mathbf{X} \times \mathbb{R}$ with respect ρ. In the supervised learning, the problem of estimating a function from random noisy data can be formalized as minimizing the expected risk

$$\inf_{f \in \mathcal{H}} \mathcal{E}(f), \quad \mathcal{E}(f) = \int (f(x) - y)^2 d\rho(x, y), \tag{1}$$

where $\mathcal{H}$ is a space of candidate solutions. An ideal empirical solution $\hat{f}$ should correspond to small excess risk

$$\mathcal{R}(\hat{f}) = \mathcal{E}(\hat{f}) - \inf_{f \in \mathcal{H}} \mathcal{E}(f). \tag{2}$$

Kernel Regularized Least Squares (KRLS)

Kernel Regularized Least Squares (KRLS) introduces the kernel trick which is based on choosing a separable reproducing kernel Hibert space (RKHS) $\mathcal{H}$ with inner product $< \cdot, \cdot >_{\mathcal{H}}$. The reproducing kernel $K : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ is a positive definite kernel, measurable and uniformly bounded. We denote with K_x the function $K(x, \cdot)$ and have $(\mathbf{K}_N)_{ij} = K(x_i, x_j)$ for all $x_1, \ldots, x_N \in \mathbf{X}$. The KRLS method for solving the problem in Eq.(1) can be expressed as

$$\hat{f}_{N,\lambda} = \arg\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2, \lambda > 0. \tag{3}$$

It is obvious that a solution $\hat{f}_{N,\lambda}$ exists and is unique. According to the representer theorem (Schölkopf et al. 2002), Eq. (3) can be transferred to

$$\hat{f}_{N,\lambda}(x) = \sum_{i=1}^{N} \hat{\alpha}_i K(x_i, x) \tag{4}$$

with

$$\hat{\boldsymbol{\alpha}} = (\mathbf{K}_N + \lambda N \mathbf{I})^{-1} y,$$

where $x_1, \ldots, x_N$ are the data points, $y = (y_1, \ldots, y_N)$ are the corresponding labels, and $\mathbf{K}_N$ is the kernel matrix.

Solving Eq.(4) is challenging as N increases: the time cost is $\mathcal{O}(N^3)$ for the linear system and the space cost is $\mathcal{O}(N^2)$ for storing the kernel matrix. To accelerate KRLS with accuracy guarantees, we proposed the following methods.

Divide-and-Conquer KRLS with Nyström

In this paper, we exploit divide-and-conquer learning, along with Nyström, preconditioning and conjugate gradient, to process the large-scale problem, which has the minimum time and space complexity compared to the state-of-the-art KRLS estimates, and has the same accuracy of exact kernel regularized least squares. Meanwhile, we provide a solid theoretical proof to guarantee the optimal convergence rate.

KRLS with Nyström We consider Nyström subsampling to reduce computational requirements, which uses a smaller matrix obtained from random sampling to approximate the empirical kernel matrix. Thus, a smaller hypothesis space $\mathcal{H}_m$ is introduced

$$\mathcal{H}_m = \{f | f = \sum_{i=1}^{m} \alpha_i K(\tilde{x}_i, \cdot), \boldsymbol{\alpha} \in \mathbb{R}^m\},$$

where sampling scale $m \leq N$, $\{\tilde{x}_1, \ldots, \tilde{x}_m\}$ are Nyström centers sampled uniformly at random without replacement from the training set. The corresponding minimizer over the space $\mathcal{H}_m$ is in the form:

$$\begin{aligned} &\tilde{f}^m_\lambda(x) = \sum_{i=1}^{m} \tilde{\alpha}_i K(\tilde{x}_i, x) \\ &\text{with} \\ &\tilde{\boldsymbol{\alpha}} = (\underbrace{\mathbf{K}_{Nm}^T \mathbf{K}_{Nm} + \lambda N \mathbf{K}_{mm}}_{\mathbf{H}})^\dagger \underbrace{\mathbf{K}_{Nm}^T y}_{\mathbf{z}}, \end{aligned} \tag{5}$$

where $\mathbf{H}^\dagger$ denotes the Moore-Penorse pseudoinverse of a matrix $\mathbf{H}$, $(\mathbf{K}_{Nm})_{ij} = K(x_i, \tilde{x}_j)$ with $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, m\}$, $(\mathbf{K}_{mm})_{kj} = K(\tilde{x}_k, \tilde{x}_j)$ with $k, j \in \{1, \ldots, s\}$ and $\mathbf{y} = [y_1, \ldots, y_m]^T \in \mathbb{R}^N$. Note that this Nyström KRLS is a linear system.

KRLS with Nyström and PCG Gradient methods (Rudi, Carratino, and Rosasco 2017; Kumar, Mohri, and Talwalkar 2012; Dieuleveut and Bach 2016) are general strategies for the unconstrained optimization problem, which are simplicity, low iteration consumption. The conjugate gradient method (Saad 1996) is one of the most popular ones, which does not need to specify the step-size and whose speed of convergence can benefit from preconditioning. We can use it to compute the coefficients $\tilde{\boldsymbol{\alpha}}$ in Eq.(5).

The idea behind preconditioning is to use a suitable matrix $\mathbf{P}$ to define an equivalent linear system with better condition number. Therefore, we consider random projections to approximately compute a preconditioner:

$$\mathbf{P} = \frac{1}{\sqrt{N}} \mathbf{T}^{-1} \mathbf{A}^{-1}, \tag{6}$$

where $\mathbf{T} = \text{chol}(\mathbf{K}_{mm})$ and $\mathbf{A} = \text{chol}(\frac{1}{m}\mathbf{T}\mathbf{T}^T + \lambda \mathbf{I})$. chol() represents the Cholesky decomposition.

Therefore, the Nyström KRLS with conjugate gradient and preconditioning can be seen as solving the following system

$$\begin{aligned} &\mathbf{P}^T \mathbf{H} \hat{\boldsymbol{\alpha}} = \mathbf{P}^T \mathbf{z}, \\ &\text{with} \\ &\hat{f}^m_\lambda(x) = \sum_{i=1}^{m} \hat{\alpha}_i K(\tilde{x}_i, x), \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{\alpha}}$ is solved via t-step conjugate gradient algorithm and $t \in \mathbb{N}$.

Divide-and-Conquer KRLS with Nyström and PCG For further saving computational costs, we consider divide-and-conquer based on the solver in Eq.(7). Given the training dataset D: $\{(x_i, y_i)\}_{i=1}^N$, the data in D be randomly partitioned into p disjoint subsets $\{D_j\}_{j=1}^p$ with $|D_1| = |D_2| = \ldots = |D_p| = n$. The j-th data subset is sent to the j-th local processor, where $j = [1, p]$, which is separately used to the solver in Eq.(7) to get $\hat{\boldsymbol{\alpha}}_j$ and the estimator $\hat{f}^m_{D_j,\lambda}$. Namely, in each local processor, we use the Nyström method in Eq.(7) to solve KRLS. Then, the final training estimator $\hat{f}^m_{D,\lambda}$ is obtained by combining the individual estimators to the center, whose formula is as follows:

$$\hat{f}^m_{D,\lambda} = \frac{1}{p} \sum_{j=1}^{p} \hat{f}^m_{D_j,\lambda}. \tag{8}$$

Algorithm 1 Divide-and-Conquer KRLS with Nyström (DC-NY)

Input: Training dataset $\{x_i\}_{i=1}^N \in \mathbb{R}^{N \times d}$, $\{y_i\}_{i=1}^N \in \mathbb{R}^N$, kernel parameter, regularization parameter λ, sampling scale m, number of iterations t, number of partitions p.

Output: $\hat{f}^m_{D,\lambda}(x)$

1: randomly partition the training dataset into p disjoint subsets $\{D_j\}_{j=1}^p$.
2: sent each data subset to each local processor.
3: // **In parallel: take j-th local processor for example**
4: get $\hat{\boldsymbol{\alpha}}_j$ based on j-th training subset by solving the problem in Eq.(7).
5: get the estimator $\hat{f}^m_{D_j,\lambda}(x) = \sum_{i=1}^m \hat{\alpha}_{j,i} K(x_i, x)$ in j-th local processor.
6: // **End parallelism**
7: compute the final estimator by synthesizing each one: $\hat{f}^m_{D,\lambda}(x) = \frac{1}{p} \sum_{j=1}^p \hat{f}^m_{D_j,\lambda}(x)$.

In the prediction stage, based on the trained $\hat{\boldsymbol{\alpha}}_j$, the new query point x_{test} is transmitted to each local processor to get an estimation $\hat{f}^m_{D_j,\lambda}(x_{test})$, then we get the finally predicted estimation $\hat{f}^m_{D,\lambda}(x_{test}) = \frac{1}{p} \sum_{j=1}^p \hat{f}^m_{D_j,\lambda}(x_{test})$ by synthesizing each $\hat{f}^m_{D_j,\lambda}(x_{test})$.

The detailed process of proposed algorithm (DC-NY) is summarized in Algorithm 1.

Theoretical Assessments

In this section, for exploring the generalization ability, we firstly introduce four standard assumptions, which are widely used in statistical learning of squared loss (Smale and Zhou 2007; Caponnetto and Vito 2007; Rudi, Carratino, and Rosasco 2017; Li, Liu, and Wang 2019). Under the basic assumptions, the theoretical bound of the proposed algorithm is provided, which is the same as that of the exact Kernel Regularized Least Squares (KRLS).

Assumption 1 There exists an $f_{\mathcal{H}} \in \mathcal{H}$ such that

$$\mathcal{E}(f_{\mathcal{H}}) = \min_{f \in \mathcal{H}} \mathcal{E}(f). \tag{9}$$

The above assumption is standard in kernel-based nonparametric regression (Smale and Zhou 2007; Caponnetto and Vito 2007), which shows that the problem in Eq.(1) has at least a solution. We also need a basic assumption on data distribution to derive probabilistic results.

Assumption 2 Let z_x be the random variable $z_x = y - f_{\mathcal{H}}(x)$, with $x \in X$, and y distributed according to $\rho(y|x)$. Then, there exists $b, \sigma > 0$ such that

$$\mathbb{E}|z_x|^e \leq \frac{1}{2} p! b^{e-2} \sigma^2$$

for any $e \geq 2$, almost everywhere on X.

This assumption is related to a noise assumption in the regression model, used to control random quantities, and holds the bounded y. When $|y| \leq b, \forall b > 1$, the assumption is satisfied with $\sigma = b$.

Assumption 3 Let C be the covariance operator as

$$C : \mathcal{H} \to \mathcal{H}, \langle f, Cg \rangle_{\mathcal{H}} = \int_{\mathbf{X}} f(x) g(x) d\rho_X(x), \forall f, g \in \mathcal{H}.$$

For $\lambda > 0$, we define the random variable $\mathcal{N}_x(\lambda) = \langle K_x, (C + \lambda \mathbf{I})^{-1} K_x \rangle_{\mathcal{H}}$ with $x \in \mathbf{X}$ distributed according to $\rho_{\mathbf{X}}$ and let $\mathcal{N}(\lambda) = \mathbb{E}\mathcal{N}_x(\lambda), \mathcal{N}_\infty(\lambda) = \sup_{x \in \mathbf{X}} \mathcal{N}_x(\lambda)$. The kernel K is measurable, C is bounded. Moreover, for all $\lambda > 0$ and a $Q > 0$,

$$\mathcal{N}_\infty(\lambda) < \infty, \tag{10}$$

$$\mathcal{N}(\lambda) \leq Q \lambda^{-\gamma}, \quad 0 < \gamma \leq 1. \tag{11}$$

This assumption controls the variance of the estimator (Rudi, Camoriano, and Rosasco 2015). Typically, this assumption ensures that the covariance operator is a well defined linear, continuous, self-adjoint, positive operator. If the kernel satisfied $\sup_{x \in X} K(x, x) = \kappa^2 < \infty$, we have $\mathcal{N}_\infty(\lambda) \leq \kappa^2 / \lambda$ for all $\lambda > 0$. γ affects the size of RKHS $\mathcal{H}$, namely it quantifies the capacity assumption. The more benign situation with smaller $\mathcal{H}$ is obtained when $\gamma \to 0$. Note that, because the operator C is trace class, Eq.(11) always holds for $\gamma = 1$.

The bias/approximation error of KRLS can be controlled by the following assumption (Rudi, Camoriano, and Rosasco 2015).

Assumption 4 There exists $s > 0, 1 \leq R < \infty$, such that

$$\left\| C^{-s} f_{\mathcal{H}} \right\|_{\mathcal{H}} < R. \tag{12}$$

The degree, $f_{\mathcal{H}}$ can be well approximated by functions in the RKHS $\mathcal{H}$, can be quantified by this assumption. This assumption can be seen as regularity of $f_{\mathcal{H}}$.

In the following, we quantify the quality of empirical solutions of Eq.(1) obtained by schemes of Eq.(8) in terms of the quantities in Assumptions 1, 2, 3 and 4.

Theorem 1. *Under Assumptions 1,2,3 and 4, let $\delta \in (0, 1]$, $r = 1/2 + \min(s, 1/2)$, $\gamma \in (0, 1]$, $\lambda = N^{-\frac{1}{2r+\gamma}}$, $N \geq n_0$ with $n_0 \in \mathbb{N}$, and $\hat{f}^m_{D,\lambda}$ be the estimator. When*

$$t \geq \mathcal{O}(\log(N)), \quad p \leq N^{\frac{2r-1}{2r+\gamma}}, \quad m \geq \mathcal{O}(N^{\frac{1}{2r+\gamma}}),$$

with probability $1 - \delta$, we have

$$\mathbb{E}[\mathcal{E}(\hat{f}^m_{D,\lambda})] - \mathcal{E}(f_{\mathcal{H}}) = \mathcal{O}(N^{-\frac{2r}{2r+\gamma}}). \tag{13}$$

Proof. The proof is given in Appendix. □

Remark 1. *Note that, $\mathcal{O}(N^{-\frac{2r}{2r+\gamma}})$ is the optimal convergence rate (Caponnetto and Vito 2007). Under the same constraints, the convergence rate of the proposed algorithm (DC-NY) is the same as the bounds of the exact KRLS (Steinwart et al. 2009), DC (Guo, Lin, and Shi 2017), Nyström (Rudi, Camoriano, and Rosasco 2015) and Nyström-iterative (Rudi, Carratino, and Rosasco 2017) by $p \leq N^{\frac{2r-1}{2r+\gamma}}$ and $m \geq \mathcal{O}\left(N^{\frac{1}{2r+\gamma}}\right)$. Although Random Features (Rudi, Camoriano, and Rosasco 2016) and DC-Random Features (Li, Liu, and Wang 2019) can also obtain the same optimal convergence rate as DC-NY, their corresponding $m \geq \mathcal{O}\left(N^{\frac{(2r-1)\gamma+1}{2r+\gamma}}\right)$ is bigger than our $m \geq \mathcal{O}\left(N^{\frac{1}{2r+\gamma}}\right)$. Namely, under the same m, the accuracy of the proposed method is theoretically better than that of Random Features and DC-Random Features. In the best case ($r = 1$ and $\gamma \asymp 0$), $\lambda \asymp 1/\sqrt{N}$, DC-NY can reach the convergence rate $1/N$. Theoretical analysis demonstrate that our algorithm is sound and effective.*

Remark 2. *The proposed algorithm DC-NY is an extraordinary expansion of approximate KRLS. We succeed in conquering the bottleneck that optimal learning rate for the combination of distributed learning algorithm, Nyström algorithm, preconditioner and conjugate gradient. By introducing a novel technique of error decomposition and applying standard integral operator framework, the proposed algorithm achieves a tight bound under some basic assumptions. This is the first time that combining these approximate methods, achieving the optimal statistical accuracy and the minimum time and space complexity.*

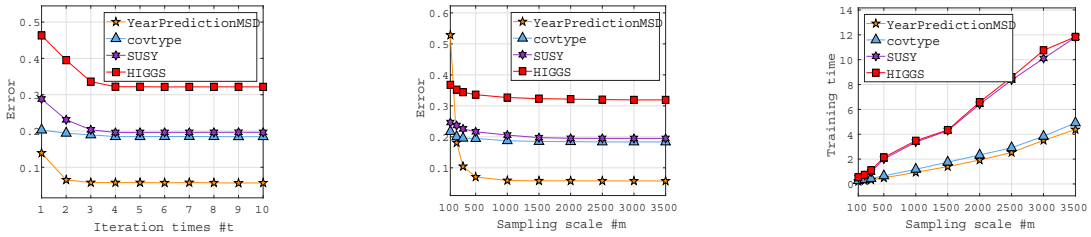
Complexity Analysis

The dataset is divided into p subsets so that the number of data in each processor is N/p.

The matrices $\mathbf{T}$ and $\mathbf{A}$ in Eq.(6) are upper-triangular matrices, so that the time complexity of $\mathbf{A}^{-1}\mathbf{x}$ and $\mathbf{A}^{-T}\mathbf{x}$

Table 2: Datasets used in this paper.

Dataset	Instance	Feature	$\sqrt{N}$	Type	σ in DC-NY	λ in DC-NY
YearPredictionMSD	463,715	90	373	Regression	5.6	7.6×10^{-6}
covtype	581,012	54	638	Multi-Classification	16	3.8×10^{-6}
SUSY	$1,000,000$	18	837	Bi-Classification	8	1.5×10^{-5}
HIGGS	$1,000,000$	28	837	Bi-Classification	5.6	1.2×10^{-4}



(a) Test error-t (b) Test error-m (c) Training time-m

Figure 1: (a) validates average test error with respect to the iteration times t of the proposed algorithm with $p = 4$ and $m = 2000$, (b) and (c) validate average test error and average training time with respect to the sampling scale m of the proposed algorithm with $t = 7$ and $p = 2$, on YearPredictionMSD, covtype, SUSY and HIGGS datasets.

($\mathbf{x}$ is a vector) are all m^2. The matrix $\mathbf{P}$ does not need to be represented explicitly. The time cost for the preconditioner, namely computing the matrices $\mathbf{T}$ and $\mathbf{A}$, is $\frac{4}{3}m^3$, which includes two Cholesky decompositions and one product of two triangular matrices. The time cost in Nyström with conjugate gradient is $\mathcal{O}(Nmt/p)$. As stated in Theorem 1, using the conjugate gradient solver typically requires $\mathcal{O}(\log(N))$ iterations. Therefore, the time complexity in DC-NY is $\mathcal{O}(\max(Nm/p, m^3))$. Compared to exact kernel regularized least squares whose time complexity is $\mathcal{O}(N^3)$, we reduce the time by a factor of $\min(\frac{N^2p}{m}, \frac{N^3}{m^3})$. Compared to the state-of-the-art approximate KRLS estimates whose time complexity is $\mathcal{O}(\frac{Nm^2}{p})$, we reduce the time by a factor of $\min(m, \frac{N}{p}\frac{1}{m})$, where the number of data in each processor $\frac{N}{p}$ is bigger than the sampling scale m.

In space complexity, the decisive element is the scale of matrix K_{nm} in Eq.(7). Therefore, the required space complexity of the proposed algorithm is only $\mathcal{O}(Nm/p)$ which is the minimum. Compared to the exact kernel regularized least squares whose space complexity is $\mathcal{O}(N^2)$, we reduce the space complexity by a factor of $\frac{Np}{m}$. Considering the concrete values of m and p this leads to a computational cost for optimal generalization, we reduce the space complexity by a factor of $N^{\frac{(2r-1)\gamma}{2r+\gamma}}$ compared to the state-of-the-art KRLS estimate, where $\frac{(2r-1)\gamma}{2r+\gamma} > 0$. Details are shown in Table 1.

To the best of our knowledge, DC-NY currently possesses the best time and space complexity to achieve the optimal statistical accuracy of KRLS.

Error Decomposition

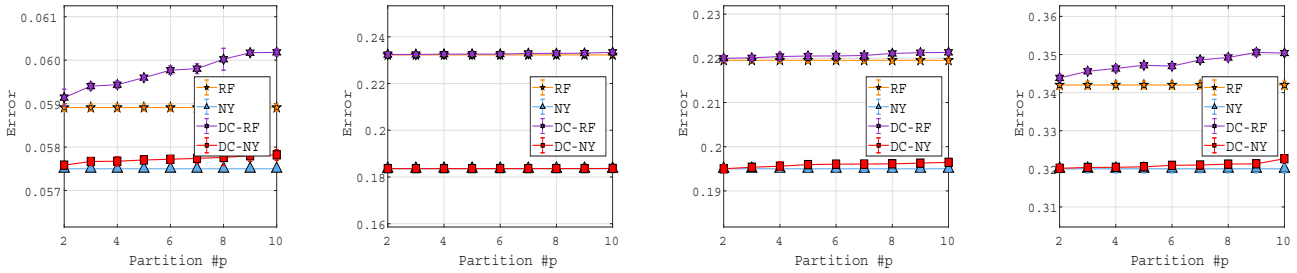
In this section, a novel technique of error decomposition is introduced for DC-NY. The main task of proving the generalization performance is to bound excess risk $\mathcal{R}(\hat{f}) = \mathcal{E}(\hat{f}) - \inf_{f\in\mathcal{H}} \mathcal{E}(f)$.

In order to describe the decomposition of excess risk clearly, we provide some estimators in advance. $\hat{f}^m_{D,\lambda}$ is our estimator in Eq.(8), namely the estimator in Eq.(8) after t iterations of the conjugate gradient algorithm. $\tilde{f}^m_{D,\lambda}$ is the correspondingly estimator by exact Nyström in Eq.(5), namely the estimator in Eq.(8) after infinite iterations of the conjugate gradient algorithm. $f^m_{D_j,\lambda}$ focus on noise-free data on the j-th subset D_j, f^m_λ is the estimator in Eq.(5) on the total dataset D. we obtain the error decomposition in Lemma 1.

Lemma 1. *Let $\hat{f}^m_{D,\lambda}$, $\tilde{f}^m_{D,\lambda}$, $f^m_{D_j,\lambda}$, f^m_λ and $f_\mathcal{H}$ de defined as above, we have*

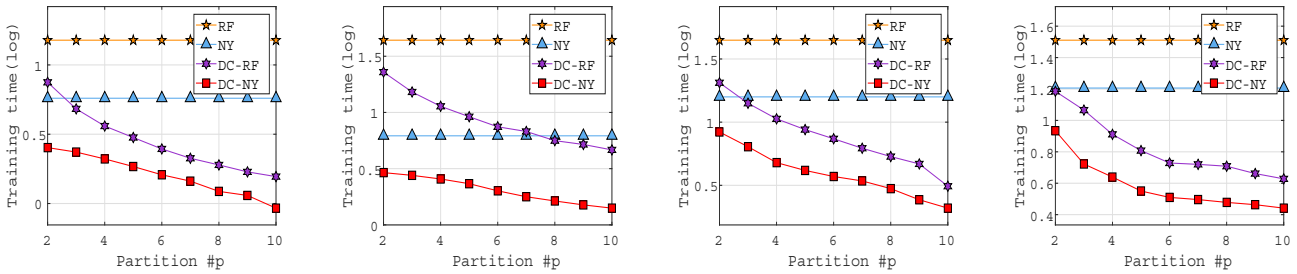
$$\begin{aligned}
&\mathbb{E}\left[\mathcal{E}(\hat{f}^m_{D,\lambda})\right] - \inf_{f\in\mathcal{H}} \mathcal{E}(f) \\
\leq& 2\mathbb{E}\left\|\hat{f}^m_{D,\lambda} - \tilde{f}^m_{D,\lambda}\right\|^2_\rho + \frac{4}{p^2}\sum_{j=1}^{p}\mathbb{E}\left\|\tilde{f}^m_{D_j,\lambda} - f^m_{D_j,\lambda}\right\|^2_\rho \\
&+ \left(\frac{8}{p^2} + \frac{4}{p}\right)\sum_{j=1}^{p}\mathbb{E}\left\|f^m_{D_j,\lambda} - f^m_\lambda\right\|^2_\rho \\
&+ \left(\frac{8}{p^2} + \frac{4}{p}\right)\sum_{j=1}^{p}\mathbb{E}\left\|f^m_\lambda - f_\mathcal{H}\right\|^2_\rho
\end{aligned} \tag{14}$$

The error decomposition and the standard integral operator framework are the keys to guarantee the generalization performance of the proposed algorithm. Our error bound is not a simple sum of block errors. Instead, it uses clever scaling and novel partitioning, then each error is transformed reasonably and bounded tightly based on the integral operator. The error bound obtained by the conventional summation of block errors is much larger than that obtained by our



(a) YearPredictionMSD (b) covtype (c) SUSY (d) HIGGS

Figure 2: Average test error and partitions p of various algorithms on YearPredictionMSD, covtype, SUSY and HIGGS datasets. With $t = 7$ and $m = 2500$.



(a) YearPredictionMSD (b) covtype (c) SUSY (d) HIGGS

Figure 3: Average training time and partitions p of various algorithms on YearPredictionMSD, covtype, SUSY and HIGGS datasets. With $t = 7$ and $m = 2500$.

method. The detail proof are shown in Appendix.

Experiment

We empirically analyze the performance of the proposed algorithm compared with the classical algorithms, considering a Gaussian kernel of width σ. The form of kernel function is as below $K(x_1, x_2) = e^{-\frac{1}{2\sigma^2}(x_1 - x_2)^2}$. Each experiment is measured on a server with 2.40GHz Intel(R) Xeon(R) E5-2630 v3 CPU and 32 GB of RAM in Matlab.

Dataset Preparation

The comparative experiments are based on four real-world datasets: SUSY, HIGGS, YearPredictionMSD and covtype, from website [1]. Each features has been normalized substracting its mean and dividing for its variance. The details are shown in Table 2. We randomly sample 1×10^6 data points on SUSY and HIGGS, use the whole of YearPredictionMSD and covtype, and then randomly divide each experimental dataset into training set and prediction set, of which the training set accounts for 70%.

Baselines and Parameters

In order to avoid contingency, each experiment is repeated 10 times. For ensuring fairness, we use the same way to tune parameters σ in $2^{[-2:+0.5:10]}$ and λ in $2^{[-21:+1:3]}$, on each dataset and algorithm. Maybe the selected parameters are not optimal, but they are sufficient to achieve satisfactory results. The detail of parameters σ and λ in DC-NY are displayed in Table 2.

We will compare our method with 3 methods. (1) RF: it is the abbreviation of Random Features (Rudi, Camoriano, and Rosasco 2016), which is a classical approximate KRLS. We use the code from website [2]. (2) NY: it represents Nyström-iterative (Rudi, Carratino, and Rosasco 2017), which combine Nyström with iterative methods. The code is from website [3]. (3) DC-RF: it represents the algorithm (Li, Liu, and Wang 2019): combining Random Features with Divide-and-Conquer. The code is from website [4]. (4) DC-NY: this is the proposed algorithm.

The error is measured with RMSE for regression problems, and with classification error for the classification problems, to be consistent with the literature.

Parameters Evaluation

As shown in (a) of Figure 1, when $t > 4$, the errors of DC-NY have converged on four datasets, which is consistent with our theoretical analysis $t > \mathcal{O}(\log(N))$.

In (b) and (c) of Figure 1, the test errors of DC-NY decline significantly when sampling scale m is a small number, and when $m = \sqrt{N}$, DC-NY has converged on four datasets which is in line with the theoretical reasoning. The bigger the m is, the longer the training time is. Therefore, in prac-

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

[2] https://github.com/superlj666

[3] https://github.com/LCSL/FALKON-paper

[4] https://github.com/superlj666

Table 3: Comparison of average training time (left) in seconds and average test error (right) in solving KRLS between RF, NY, DC-RF and DC-NY algorithms on YearPredictionMSD, covtype, SUSY and HIGGS datasets, with partitions $p = 2$, iteration times $t = 7$, sampling scale $m = 1500$ and 2000. We bold the numbers of the best algorithm. YPMSD is the abbreviation of YearPredictionMSD.

Dataset	RF(m = 1500)		NY(m = 1500)		DC-RF(m = 1500)		DC-NY(m = 1500)	
	time	error	time	error	time	error	time	error
YPMSD	7.270	0.061 ± 0.00008	2.90	**0.058±0.00012**	3.560	0.062 ± 0.00007	**1.60**	**0.058±0.00013**
covtype	17.90	0.233 ± 0.00079	5.02	**0.185±0.00050**	9.060	0.234 ± 0.00013	**2.56**	**0.185±0.00027**
SUSY	19.24	0.221 ± 0.00031	8.32	**0.197±0.00068**	9.020	0.223 ± 0.00027	**4.09**	**0.197±0.00064**
HIGGS	20.87	0.346 ± 0.00094	9.02	**0.323±0.00048**	10.84	0.347 ± 0.00070	**4.34**	**0.323±0.00057**
Dataset	RF(m = 2000)		NY(m = 2000)		DC-RF(m = 2000)		DC-NY(m = 2000)	
	time	error	time	error	time	error	time	error
YPMSD	10.68	0.060 ± 0.00007	3.49	**0.058±0.00016**	5.590	0.061 ± 0.00007	**2.09**	**0.058±0.00003**
covtype	25.79	0.233 ± 0.00041	5.62	**0.185±0.00067**	12.88	0.234 ± 0.00067	**2.63**	**0.185±0.00047**
SUSY	27.06	0.220 ± 0.00035	12.9	**0.195±0.00065**	13.67	0.222 ± 0.00061	**6.40**	**0.195±0.00059**
HIGGS	22.58	0.343 ± 0.00118	9.14	**0.321±0.00050**	11.68	0.345 ± 0.00108	**5.19**	0.322 ± 0.00053

tice, we only need to take a small m to obtain a satisfactory error, which will result in significant savings in computing resources.

Comparison with Baselines

Figure 2 shows how the number of partition affects the error of the algorithms on test sets. The horizontal coordinate represents the number of partitions p, and the vertical coordinate the average test errors of different algorithms. DC-NY keeps the optimal accuracy level which is consistent with the theoretical analysis. With the increase of the number of partition p, the errors increase in each algorithms and our algorithm provides competitive accuracy. Taking the same m and p, RF and DC-RF have bigger error than DC-NY, which is in line with the theoretical analysis.

Figure 3 shows the training time of the algorithms on train sets with respect to the number of partition p. The vertical coordinate is the training time (logarithmizing it) of different algorithms in seconds. With the increase of p, the training time decreases in divide-and-conquer algorithms (DC-RF and DC-NY). Our algorithm has a significant advantage over other algorithms in the training time. On covtype, the time cost of DC-RF with $p = 10$ is higher than that of DC-NY with $p = 2$, that is to say, our algorithm requires less expensive hardware devices, under the same scenario and time cost. The bigger the number of data in a subprocessor, the more obvious the time advantage of the proposed algorithm is. Apparently, combining Figure 2 with Figure 3, we get that DC-NY can use fewer hardware devices (processors) to reach a smaller error under the same time cost, which is consistent with the statistical analysis.

Table 3 shows the specific numerical information of experimental results when $m = 1500$ and $m = 2000$. Apparently, DC-NY always keeps the least time consumption than other algorithms on four datasets. Even on SUSY and HIGGS datatset of millions of points, the training time of DC-NY is always a few seconds. In test error, DC-NY keeps the optimal value or just has a little gap with the optimal, which validate the effectiveness of the proposed algorithm.

The empirical performance verifies our theoretical results that the proposed algorithm has a prominent advantage in speed while achieving satisfactory accuracy.

Conclusions

We focus on the trade-off between statistical performance and computational requirements to propose a novel approximate KRLS estimator DC-NY, which achieves the same accuracy of exact KRLS and has the minimum time complexity and space complexity, compared to the state-of-the-art approximate KRLS estimates. The empirical performance verifies the theoretical analysis. In the future, we try to explore other distributed methods to solve KRLS.

Acknowledgments

Thank my boyfriend Lei Wang for his support. This work was supported in part by the National Natural Science Foundation of China (No.61703396, No.61673293), the CCF-Tencent Open Fund, the Youth Innovation Promotion Association CAS, the Excellent Talent Introduction of Institute of Information Engineering of CAS (No. Y7Z0111107), and the Beijing Municipal Science and Technology Project (No. Z191100007119002).

References

Avron, H.; Clarkson, K. L.; and Woodruff, D. P. 2017. Faster kernel ridge regression using sketching and preconditioning. *Siam Journal on Matrix Analysis & Applications* 38(4):1116–1138.

Bo, D.; Bo, X.; He, N.; Liang, Y.; Raj, A.; Balcan, M. F.; and Le, S. 2014. Scalable kernel methods via doubly stochastic gradients. In *International Conference on Neural Information Processing Systems*, 3041–3049.

Camoriano, R.; Angles, T.; Rudi, A.; and Rosasco, L. 2016. Nytro: When subsampling meets early stopping. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 1403–1411.

Caponnetto, A., and Vito, E. D. 2007. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics* 7(3):331–368.

Carratino, L.; Rudi, A.; and Rosasco, L. 2018. Learning with sgd and random features. In *Advances in Neural Information Processing Systems*, 10192–10203.

Cutajar, K.; Osborne, M.; Cunningham, J.; and Filippone, M. 2016. Preconditioning kernel matrices. In *International Conference on Machine Learning*, 2529–2538.

Dieuleveut, A., and Bach, F. 2016. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics* 44(4):1363–1399.

Ding, L.; Liao, S.; Liu, Y.; Yang, P.; and Gao, X. 2018. Randomized kernel selection with spectra of multilevel circulant matrices. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2910–2917.

Fasshauer, G. E., and Mccourt, M. J. 2012. Stable evaluation of gaussian radial basis function interpolants. *Siam Jouranl on Scientific Couputing* 34(2):A737–A762.

Gonen, A.; Orabona, F.; and Shalev-Shwartz, S. 2016. Solving ridge regression using sketched preconditioned svrg. In *International Conference on International Conference on Machine Learning*, 1397–1405.

Guo, Z. C.; Lin, S. B.; and Shi, L. 2017. Distributed learning with multi-penalty regularization. *Applied & Computational Harmonic Analysis*.

Kumar, S.; Mohri, M.; and Talwalkar, A. 2012. Sampling methods for the nyström method. *Journal of Machine Learning Research* 13(Apr):981–1006.

Li, J.; Liu, Y.; Yin, R.; Zhang, H.; Ding, L.; and Wang, W. 2018. Multi-class learning: from theory to algorithm. In *Advances in Neural Information Processing Systems 31 (NIPS)*, 1586–1595.

Li, J.; Liu, Y.; and Wang, W. 2019. Distributed learning with random features. *arXiv preprint arXiv:1906.03155*.

Lin, J., and Cevher, V. 2018. Optimal distributed learning with multi-pass stochastic gradient methods. In *Proceedings of the 35th International Conference on Machine Learning*, 3098–3107.

Liu, Y., and Liao, S. 2014. Kernel selection with spectral perturbation stability of kernel matrix. *SCIENCE CHINA Information Sciences* 57(11):1–10.

Liu, Y., and Liao, S. 2015. Eigenvalues ratio for kernel selection of kernel methods. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2814–2820.

Liu, Y.; Liao, S.; Lin, H.; Yue, Y.; and Wang, W. 2017. Infinite kernel learning: generalization bounds and algorithms. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, 2280–2286.

Liu, Y.; Lin, H.; Ding, L.; Wang, W.; and Liao, S. 2018. Fast cross-validation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, 2497–2503.

Liu, Y.; Jiang, S.; and Liao, S. 2014. Efficient approximation of cross-validation for kernel methods using Bouligand influence function. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 324–332.

Lo, G. L.; Rosasco, L.; Odone, F.; De, V. E.; and Verri, A. 2008. Spectral algorithms for supervised learning. *Neural Computation* 20(7):1873–1897.

Ma, S., and Belkin, M. 2017. Diving into the shallows: a computational perspective on large-scale shallow learning. *arXiv preprint arXiv:1703.10622*.

Rahimi, A., and Recht, B. 2007. Random features for large-scale kernel machines. In *International Conference on Neural Information Processing Systems*, 1177–1184.

Raskutti, G.; Wainwright, M. J.; and Yu, B. 2014. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *The Journal of Machine Learning Research* 15(1):335–366.

Rudi, A.; Camoriano, R.; and Rosasco, L. 2015. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, 1657–1665.

Rudi, A.; Camoriano, R.; and Rosasco, L. 2016. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, 3215—3225.

Rudi, A.; Carratino, L.; and Rosasco, L. 2017. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, 3888–3898.

Saad, Y. 1996. *Iterative Methods for Sparse Linear Systems*. SIAM.

Schölkopf, B.; Smola, A. J.; Bach, F.; et al. 2002. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Smale, S., and Zhou, D. X. 2007. Learning theory estimates via integral operators and their approximations. *Constructive Approximation* 26(2):153–172.

Smola, A. J. 2000. Sparse greedy matrix approximation for machine learning. In *Proceedings of the 17th International Conference on Machine Learning*, 911–911.

Steinwart, I.; Hush, D. R.; Scovel, C.; et al. 2009. Optimal rates for regularized least squares regression. In *Annual Conference on Learning Theory*.

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

Tu, S.; Roelofs, R.; Venkataraman, S.; and Recht, B. 2016. Large scale kernel learning using block coordinate descent. *arXiv preprint arXiv:1602.05310*.

Williams, C. K., and Seeger, M. 2001. Using the nyström method to speed up kernel machines. In *International Conference on Neural Information Processing Systems*, 682–688.

Yang, Y.; Pilanci, M.; and Wainwright, M. J. 2015. Randomized sketches for kernels: Fast and optimal non-parametric regression. *The Annals of Statistics* 45(3):991–1023.

Yin, R.; Liu, Y.; Wang, W.; and Meng, D. 2019. Sketch kernel ridge regression using circulant matrix: Algorithm and theory. *IEEE transactions on neural networks and learning systems*.

Zhang, Y.; Duchi, J. C.; and Wainwright, M. J. 2013. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research* 30(1):592–617.