Distributed Ranking with Communications: Approximation Analysis and Applications

Hong Chen,[1] Yingjie Wang,[2] Yulong Wang,[2*] Feng Zheng[3*]
[1]College of Science, Huazhong Agricultural University, China
[2]College of Informatics, Huazhong Agricultural University, China
[3]Department of Computer Science and Engineering, Southern University of Science and Technology, China
chenh@mail.hzau.edu.cn, yjaywang@126.com, wangyulong6251@gmail.com, zhengf@sustech.edu.cn

Abstract

Learning theory of distributed algorithms has recently attracted enormous attention in the machine learning community. However, most of existing works focus on learning problem with pointwise loss and does not consider the communication among local processors. In this paper, we propose a new distributed pairwise ranking with communication (called DLSRank-C) based on the Newton-Raphson iteration, and establish its learning rate analysis in probability. Theoretical and empirical assessments demonstrate the effectiveness of DLSRank-C under mild conditions.

Introduction

Distributed learning under divide and conquer strategy has attracted increasing attention recently, since data are often stored in multiple servers for many applications (Zhang, Duchi, and Wainwright 2015; Hsieh, Si, and Dhillon 2014; Xu et al. 2016; Guo, Lin, and Shi 2019). In kernel methods, the distributed learning system depends on three key ingredients including local kernel machines, communication, and synthesization (Lin, Wang, and Zhou 2020). Each local kernel machine tackles the data subset with the computation feasibility. The communication strategy aims to exchange some important information among subsets, e.g., gradients (Zeng and Yin 2018) and local predictor (Huang and Huo 2019). Based on the above building blocks, the global machine is constructed by the average over local estimators (Lin, Wang, and Zhou 2020; Li, Liu, and Wang 2019).

Usually, the communication strategy for distributed learning systems is useful to enlarge the number of local machines to reach fast learning rate (i.e., the speed of convergence in the generalization errors). In particular, the communication strategy based on Newton-Raphson iteration is incorporated into the framework of distributed algorithms, e.g., linear ridge regression (Huang and Huo 2019) and kernel ridge regression (Lin, Wang, and Zhou 2020; Li, Liu, and Wang 2019). However, the existing works are limited to the regression problem with the pointwise loss, e.g., the least squared loss. It is natural and important to further investigate theory foundations of distributed pairwise learning with communication, e.g., learning under MEE principle (Hu et al. 2013), pairwise ranking (Cortes, Mohri, and Rastogi 2007; Agarwal and Niyogi 2009; Chen 2012; Kriukova, Pereverzyev, and Tkachenko 2016; Kriukova et al. 2016). Indeed, there are some efforts to characterize the generalization bounds of MEE-based distributed algorithms in (Hu, Wu, and Zhou 2020; Guo, Hu, and Wu 2020) and the distributed least square ranking (DLSRank) (Chen, Li, and Pan 2019). However, both of them do not consider the communication strategy among local processors and the weighted average can not improve the approximation ability of each local machine (Lin, Wang, and Zhou 2020). To improve the approximation of distributed pairwise learning, we introduce an efficient communication strategy and synthesization method to the ranking problem.

Following the operator representation of ranking in (Chen 2012; Chen et al. 2013; Kriukova, Pereverzyev, and Tkachenko 2016) and the communication strategy in (Lin, Wang, and Zhou 2020; Li, Liu, and Wang 2019), we formulate a distributed ranking algorithm, called distributed least square ranking with communication (DLSRank-C). Under this communication strategy, the proposed distributed algorithm can better utilize the exchangeable information of local machines to improve the approximation stability of final predictor. Learning theory analysis supports the motivation of algorithmic design, where the faster learning rates can be obtained with the help of communication strategy than the previous distributed ranking in (Chen, Li, and Pan 2019). Our main tools to achieve this goal are the operator representation for the solution of distributed least square ranking (Chen 2012; Kriukova, Pereverzyev, and Tkachenko 2016), the operator decomposition and approximation strategy (Lin, Guo, and Zhou 2017; Chang, Lin, and Zhou 2017; Guo, Shi, and Wu 2017), and the Newton-Raphson iteration (Lin, Wang, and Zhou 2020). Furthermore, this paper establish the learning rates of DLSRank-C in probability, which is different from the related result in expectation (Chen, Li, and Pan 2019). The probability version usually is desirable crucial to characterize the generalization performance of DLSRank-C in a single trial (Lin, Wang, and Zhou 2020).

*Corresponding authors.

Distributed Ranking with Communication

In this section, we employ the communication strategy for DLSRank to improve its approximation ability.

Distributed Least Square Ranking (DLSRank)

Let $\mathcal{Z} := (\mathcal{X}, \mathcal{Y}) \subset \mathbb{R}^{p+1}$ be a compact metric space, where $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset [-M, M]$ for some positive constant M. Assume that observations $D := \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ are independently drawn from an intrinsic Borel probability measure ρ on $\mathcal{Z}$. The primary purpose of least squares ranking is to find a function: $f : \mathcal{X} \to \mathbb{R}$, by means of empirical observations, such that the ranking risk

$$\mathcal{E}(f) = \int_{\mathcal{Z}} \int_{\mathcal{Z}} (y - y' - (f(\mathbf{x}) - f(\mathbf{x}')))^2 d\rho(\mathbf{x}, y) d\rho(\mathbf{x}', y') \tag{1}$$

as small as possible. As illustrated in (Chen 2012; Chen et al. 2013; Kriukova, Pereverzyev, and Tkachenko 2016; Hu et al. 2013), the optimal score predictor under the criterion (1) is the regression function defined by

$$f_\rho(\mathbf{x}) = \int_{\mathcal{Y}} y d\rho(y|X = \mathbf{x}), \mathbf{x} \in \mathcal{X},$$

where $\rho(y|X = \mathbf{x})$ denotes the conditional distribution for given input $\mathbf{x}$.

This section recalls the distributed least squares ranking. Let $D = \cup_{j=1}^m D_j$ and each subset $D_j := \{(\mathbf{x}_i^j, y_j^j)\}_{i=1}^{|D_j|}$ be stored in the j-th local machine for $1 \le j \le m$. Here, $|D|$ denotes the cardinality of D with $|D| = \sum_{j=1}^m |D_j|$. The hypothesis space used here is the reproducing kernel Hilbert space (RKHS) $(\mathcal{H}_K, \|\cdot\|_K)$ associated with a mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (Aronszajn 1950; Cucker and Zhou 2007).

The DLSRank, with a regularized parameter $\lambda > 0$, is defined by

$$\bar{f}_{D,\lambda}^0 = \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2} f_{D_j,\lambda}, \tag{2}$$

where the least squares ranking (LSRank)

$$f_{D_j,\lambda} = \arg\min_{f \in \mathcal{H}_K} \{\mathcal{E}_{D_j}(f) + \lambda \|f\|_K^2\} \tag{3}$$

and

$$\mathcal{E}_{D_j}(f) = \frac{1}{|D_j|^2} \sum_{i,k=1}^{|D_j|} (y_i^j - y_k^j - (f(\mathbf{x}_i^j) - f(\mathbf{x}_k^j)))^2.$$

The learning rates of (2) have been investigated in (Chen, Li, and Pan 2019) with the help of operator approximation (Smale and Zhou 2007; Lin, Guo, and Zhou 2017; Chang, Lin, and Zhou 2017). However, the learning performance of DLSRank degrades when m increases, since the weighted averaging in (2) cannot improve the approximation ability of DLSRank in each local machine (Huang and Huo 2019; Lin, Wang, and Zhou 2020). Therefore, it is necessary to incorporate some communication strategies for improving its approximation ability.

Distributed Least Squares Ranking with Communication (DLSRank-C)

In this section, we state the DLSRank with communication based on Newton-Raphson iteration (Huang and Huo 2019; Lin, Wang, and Zhou 2020).

The following operators have been used for learning theory analysis, see, e.g., (Smale and Zhou 2005, 2007; Rosasco, Belkin, and Vito 2010; Sun and Wu 2009). Let $S_D : \mathcal{H}_K \to \mathbb{R}^{|D|}$ be the sampling operator defined by $S_D f := (f(\mathbf{x}))_{(\mathbf{x},y)\in D}$ and let $S_D^T : \mathbb{R}^{|D|} \to \mathcal{H}_K$ be its scaled adjoint operator defined by

$$S_D^T \mathbf{c} := \frac{1}{|D|} \sum_{i=1}^{|D|} c_i K_{\mathbf{x}_i}, \mathbf{c} = (c_1, \cdots, c_{|D|})^T \in \mathbb{R}^{|D|},$$

where $K_{\mathbf{x}} = K(\mathbf{x}, \cdot)$. The ranking integral operator L_K, introduced in (Chen 2012), is given by

$$L_K f = \int_{\mathcal{X}} \int_{\mathcal{X}} f(\mathbf{x})(K_{\mathbf{x}} - K_{\mathbf{x}'}) d\rho_{\mathcal{X}}(\mathbf{x}) d\rho_{\mathcal{X}}(\mathbf{x}'),$$

where $\rho_{\mathcal{X}}$ is the margin distribution of ρ with respect to $\mathcal{X}$. The empirical ranking operator is denoted as

$$L_{K,D} f := S_D^T W_D S_D f = \frac{1}{|D|^2} \sum_{(\mathbf{x},y)(\mathbf{x}',y')\in D} f(\mathbf{x})(K_{\mathbf{x}} - K_{\mathbf{x}'}),$$

where

$$W_D = \mathbb{I}_{|D|} - \frac{1}{|D|} \mathbf{1}_{|D|} \mathbf{1}_{|D|}^T = \frac{1}{|D|}(|D|\mathbb{I} - \mathbf{1}_D \mathbf{1}_D^T),$$

$\mathbb{I}_{|D|}$ is the $|D|$-order identity matrix and $\mathbf{1}_{|D|} = (1, \cdots, 1)^T \in \mathbb{R}^{|D|}$.

As stated in (Chen 2012; Kriukova, Pereverzyev, and Tkachenko 2016; Chen, Li, and Pan 2019), for given observations D, the representations of LSRank (3) and DLSRank (2) are

$$f_{D,\lambda} = (L_{K,D} + \frac{\lambda}{2}\mathbb{I}_{|D|})^{-1} S_D^T W_D Y_D$$

and

$$\bar{f}_{D,\lambda}^0 = \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2} (L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1} S_{D_j}^T W_{D_j} Y_{D_j},$$

where $Y_D = (y)_{(\mathbf{x},y)\in D} \in \mathbb{R}^{|D|}$ and $Y_{D_j} = (y)_{(\mathbf{x},y)\in D_j} \in \mathbb{R}^{|D_j|}$. By direct computation, we deduce that, $\forall f \in \mathcal{H}_K$,

$$f_{D,\lambda} = f - (L_{K,D} + \frac{\lambda}{2}\mathbb{I}_{|D|})^{-1}[(L_{K,D} + \frac{\lambda}{2}\mathbb{I}_{|D|})f - S_D^T W_D Y_D] \tag{4}$$

and

$$\bar{f}_{D,\lambda}^0 = f - \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2} (L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1} [(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})f - S_{D_j}^T W_{D_j} Y_{D_j}]. \tag{5}$$

Following the proof of Lemma 2 in (Chen 2012), we know that the gradient of regularized risk (3) over $\mathcal{H}_K$ on f is

$$\begin{aligned} G_{D_j,\lambda,f} &= \frac{4}{|D_j|^2}\sum_{(\mathbf{x},y),(\mathbf{x}',y')\in D_j}(yK_{\mathbf{x}'} - yK_{\mathbf{x}} \\ &\quad + f(\mathbf{x})K_{\mathbf{x}} - f(\mathbf{x})K_{\mathbf{x}'}) + 2\lambda f \\ &= 4(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})f - 4S_{D_j}^T W_{D_j} Y_{D_j} \end{aligned}$$

and its Hessian $H_{D_j,\lambda} = 4(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})$. Then, both (4) and (5) can be regarded as the well-known Newton-Raphson iteration. Inspired by the recent works in (Lin, Wang, and Zhou 2020; Li, Liu, and Wang 2019), we propose a distributed least-squares ranking with communication strategy based on Newton-Raphson iteration, which is formed as

$$\begin{aligned} \bar{f}_{D,\lambda}^l &= \bar{f}_{D,\lambda}^{l-1} - \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1} \\ &\quad \big[(L_{K,D} + \frac{\lambda}{2}\mathbb{I}_{|D|})\bar{f}_{D,\lambda}^{l-1} - S_D^T W_D Y_D\big] \\ &= \bar{f}_{D,\lambda}^{l-1} - \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2} H_{D_j,\lambda}^{-1} G_{D,\lambda,\bar{f}_{D,\lambda}^{l-1}}, \quad (6) \end{aligned}$$

where l denote the l-th iteration and

$$G_{D,\lambda,\bar{f}_{D,\lambda}^{l-1}} = \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2} G_{D_j,\lambda,\bar{f}_{D,\lambda}^{l-1}}$$

is the global gradient that is achieved via communicating the local gradients.

Approximation Analysis

In this section, we aim to provide the theoretical analysis on the approximation ability of $\bar{f}_{D,\lambda}^0$ and $\bar{f}_{D,\lambda}^l$ in probability. Define a stepping-stone function as

$$f_\lambda = \arg\min_{f\in\mathcal{H}_K}\{\mathcal{E}(f) + \lambda\|f\|_K^2\}.$$

The following inequalities, established in Lemma 2 and Proposition 2 of (Chen, Li, and Pan 2019), are used for our error analysis.

Lemma 1 *For $f_{D_j,\lambda}$ defined in (3) and $f_\rho \in \mathcal{H}_K$, there holds*

$$\|f_{D_j,\lambda} - f_\lambda\|_K \le T_{j1} + T_{j2}\|f_\lambda\|_K,$$

where

$$T_{j1} = \|(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1}(S_{D_j}^T W_{D_j} Y_{D_j} - L_K f_\rho)\|_K$$

and

$$T_{j2} = \|(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1}(L_{K,D_j} - L_K)\|_K.$$

Lemma 2 *For any $0 < \delta < 1$, with confidence at least $1-\delta$, there hold*

$$T_{j1} \le \frac{48\kappa M\log(2/\delta)}{\lambda\sqrt{|D_j|}} + \frac{1}{\lambda|D_j|}\|L_K f_\rho\|_K$$

and

$$T_{j2} \le \frac{54\kappa}{\lambda\sqrt{|D_j|}}(\log(2/\delta)+1),$$

where $\kappa = \sup_{\mathbf{x}\in\mathcal{X}} K(\mathbf{x},\mathbf{x})$, $\forall \mathbf{x}\in\mathcal{X}$.

Theorem 1 *Assume that $L_K^{-r} f_\rho \in \mathcal{H}_K$ with $0 < r \le 1$, where L_K^r is the r-th power of L_K. For $\bar{f}_{D,\lambda}^0$ defined in (5), there holds*

$$\|\bar{f}_{D,\lambda}^0 - f_\rho\|_K \le C_0\log(2/\delta)\Big(\sum_{j=1}^m \frac{|D_j|^{\frac{3}{2}}}{\sum_{k=1}^m |D_k|^2}\Big)^{\frac{r}{1+r}}$$

with confidence at least $1-\delta$, where C_0 is a positive constant that depends on M, κ and λ.

Proof: According to the definition of $\bar{f}_{D,\lambda}^0$, we get

$$\bar{f}_{D,\lambda}^0 - f_\lambda = \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}(f_{D_j,\lambda} - f_\lambda). \quad (7)$$

Combining the above equality with Lemmas 1 and 2, we obtain

$$\begin{aligned} &\|\bar{f}_{D,\lambda}^0 - f_\lambda\|_K \\ &\le \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}\Big\{\frac{48\kappa M\log(2/\delta)}{\lambda\sqrt{|D_j|}} + \frac{1}{\lambda|D_j|}\|L_K f_\rho\|_K \\ &\qquad + \frac{54\kappa}{\lambda\sqrt{|D_j|}}(\log(2/\delta)+1)\|f_\lambda\|_K\Big\} \\ &\le C_1\log(2/\delta)\lambda^{-1}\sum_{j=1}^m \frac{|D_j|^{\frac{3}{2}}}{\sum_{k=1}^m |D_k|^2} \end{aligned} \quad (8)$$

with confidence at least $1-\delta$.

According to Proposition 5 in (Chen 2012), we have

$$\|f_\lambda - f_\rho\|_K \le \|L_K^{-r} f_\rho\|_K \lambda^r,\ r\in(0,1]. \quad (9)$$

Based on above two inequalities (8) and (9), we deduce that, with confidence at least $1-\delta$,

$$\begin{aligned} &\|\bar{f}_{D,\lambda}^0 - f_\rho\|_K \\ &\le \|\bar{f}_{D,\lambda}^0 - f_\lambda\|_K + \|f_\lambda - f_\rho\|_K \\ &\le C_1\log(2/\delta)\lambda^{-1}\sum_{j=1}^m \frac{|D_j|^{\frac{3}{2}}}{\sum_{k=1}^m |D_k|^2} + \|L_K^{-r} f_\rho\|_K \lambda^r. \end{aligned}$$

By taking $\lambda^{1+r} = \mathcal{O}(\sum_{j=1}^m \frac{|D_j|^{\frac{3}{2}}}{\sum_{k=1}^m |D_k|^2})$, we get the desired result. □

Remark 1 *When $m = 1$, the learning rate $\mathcal{O}(\log(2/\delta)|D|^{\frac{-r}{2+2r}})$ is consistent with the convergence analysis for DLSRank in (Chen, Li, and Pan 2019). When $|D_1| = \cdots = |D_m| = \frac{|D|}{m}$, the learning rate is $\mathcal{O}(\log(2/\delta)(\frac{m}{|D|})^{\frac{r}{2(1+r)}})$. The consistency of DLSRank can be guaranteed as $\lim_{|D|\to\infty}\frac{m}{|D|} = 0$.*

Now we establish the convergence analysis of DLSRank-C. The error decomposition is crucial for our analysis.

Lemma 3 *Let $f_{D,\lambda}$ and $f_{D,\lambda}^l$ be defined in (4) and (6) respectively. We have*

$$\begin{aligned} &\|\bar{f}_{D,\lambda}^l - f_{D,\lambda}\|_K \\ &\le \Big[\frac{4}{\lambda}\sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}\|L_K - L_{K,D_j}\|_K\Big]^l \|\bar{f}_{D,\lambda}^0 - f_{D,\lambda}\|_K. \end{aligned}$$

Proof: Recall that

$$\bar{f}^l_{D,\lambda} = \bar{f}^{l-1}_{D,\lambda} - \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1} [(L_{K,D} + \frac{\lambda}{2}\mathbb{I})\bar{f}^{l-1}_{D,\lambda} - S_D^T W_D Y_D]$$

and

$$f_{D,\lambda} = \bar{f}^{l-1}_{D,\lambda} - (L_{K,D} + \frac{\lambda}{2}\mathbb{I})^{-1}[(L_{K,D} + \frac{\lambda}{2}\mathbb{I})\bar{f}^{l-1}_{D,\lambda} - S_D^T W_D Y_D].$$

Then, we have

$$\begin{aligned}
&f_{D,\lambda} - \bar{f}^l_{D,\lambda} \\
&= \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}[(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1} - (L_{K,D} + \frac{\lambda}{2})^{-1}] \\
&\qquad [(L_{K,D} + \frac{\lambda}{2}\mathbb{I})\bar{f}^{l-1}_{D,\lambda} - S_D^T W_D Y_D] \\
&= \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1}(L_{K,D} - L_{K,D_j}) \\
&\qquad (L_{K,D} + \frac{\lambda}{2})^{-1}[(L_{K,D} + \frac{\lambda}{2}\mathbb{I})\bar{f}^{l-1}_{D,\lambda} - S_D^T W_D Y_D] \\
&= \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1} \\
&\qquad (L_{K,D} - L_K)(\bar{f}^{l-1}_{D,\lambda} - f_{D,\lambda}) \\
&+ \sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}(L_{K,D_j} + \frac{\lambda}{2}\mathbb{I}_{|D_j|})^{-1} \\
&\qquad (L_K - L_{K,D_j})(\bar{f}^{l-1}_{D,\lambda} - f_{D,\lambda}).
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\|\bar{f}^l_{D,\lambda} - f_{D,\lambda}\|_K \\
&\le \frac{2}{\lambda}\sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}(\|L_{K,D} - L_K\| \\
&\qquad + \|L_K - L_{K,D_j}\|)\|\bar{f}^{l-1}_{D,\lambda} - f_{\rho,\lambda}\|_K \\
&\le (\frac{4}{\lambda}\sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}\|L_K - L_{K,D_j}\|)\|\bar{f}^{l-1}_{D,\lambda} - f_{D,\lambda}\|_K \\
&\le (\frac{4}{\lambda}\sum_{j=1}^m \frac{|D_j|^2}{\sum_{k=1}^m |D_k|^2}\|L_K - L_{K,D_j}\|)^l\|\bar{f}^0_{D,\lambda} - f_{D,\lambda}\|_K.
\end{aligned} \tag{10}$$

□

The following characterization for operator approximation has been stated in (Chen 2012; Kriukova, Pereverzyev, and Tkachenko 2016).

Lemma 4 *For the training set D_j drawn independently from ρ, and any $\delta \in (0,1)$, we have*

$$\|L_K - L_{K,D_j}\| \le \frac{24\kappa^2}{\sqrt{|D_j|}}\log(2/\delta) + \frac{2\kappa^2}{|D_j|}$$

with confidence at least $1-\delta$.

It is a position to present the learning rate of DLSRank-C.

Theorem 2 *Assume that $L_K^{-r} f_\rho \in \mathcal{H}_K$ with $0 < r \le 1$ and taking $\lambda = |D|^{-\frac{1}{2r+2}}$. For any $\delta \in (0,1)$, with confidence at least $1-\delta$, we have*

$$\begin{aligned}
&\|\bar{f}^l_{D,\lambda} - f_\rho\|_K \\
\le\ & C(\log(2/\delta))^{l+1}\max\{(m^{\frac{1}{2}}|D|^{-\frac{r}{2+2r}})^{l+1}, |D|^{-\frac{r}{2r+2}}\},
\end{aligned}$$

where C is a positive constant that depends on κ, M and λ.

Proof: In terms of Lemmas 3-4 and $|D_j| = |D|/m$, $\forall j = 1, \cdots, m$, we deduce that

$$\begin{aligned}
\|\bar{f}^l_{D,\lambda} - f_{D,\lambda}\|_K &\le \widetilde{C}(\log(\frac{2}{\delta}))^l(\frac{m}{\lambda^2|D|})^{l/2}\|\bar{f}^0_{D,\lambda} - f_{D,\lambda}\|_K \\
&\le \widetilde{C}(\log(\frac{2}{\delta}))^l(\frac{m}{\lambda^2|D|})^{l/2}[\|\bar{f}^0_{D,\lambda} - f_\lambda\|_K \\
&\quad + \|f_{D,\lambda} - f_\lambda\|_K].
\end{aligned}$$

From the proofs of Theorem 1, we know that, with confidence $1-\delta$,

$$\|\bar{f}^0_{D,\lambda} - f_\lambda\|_K \le 48\kappa M\log(2/\delta)(\frac{m}{\lambda^2|D|})^{\frac{1}{2}}$$

and

$$\|f_{D,\lambda} - f_\lambda\|_K \le 48\kappa M\log(2/\delta)(\frac{m}{\lambda^2|D|})^{\frac{1}{2}}.$$

Combining above inequalities, we get with confidence $1-\delta$

$$\|\bar{f}^l_{D,\lambda} - f_{D,\lambda}\|_K \le \widetilde{C}(\log(2/\delta))^{l+1}(\frac{m}{\lambda^2|D|})^{\frac{l+1}{2}}. \tag{11}$$

As shown in Theorem 1, there holds

$$\|f_{D,\lambda} - f_\rho\|_K \le C_2\log(2/\delta)|D|^{-\frac{r}{2r+2}} \tag{12}$$

with confidence at least $1-\delta$. By considering the inequality

$$\|\bar{f}^l_{D,\lambda} - f_\rho\| \le \|\bar{f}^l_{D,\lambda} - f_{D,\lambda}\|_K + \|f_{D,\lambda} - f_\rho\|$$

with (11) and (12), we get the desired result. □

Remark 2 *When $m \le |D|^{\frac{rl}{(1+r)(1+l)}}$, the learning rate is $\mathcal{O}(|D|^{\frac{-r}{2+2r}})$, which is faster than Theorem 1 for DLSRank, e.g., $\mathcal{O}((\frac{|D|}{m})^{\frac{-r}{2r+2}})$. When $m \ge |D|^{\frac{rl}{(1+r)(1+l)}}$ and $\lim_{|D|\to\infty} m^{\frac{1}{2}}|D|^{\frac{-r}{2r+2}} = 0$, the learning rate is $\mathcal{O}(m^{\frac{l+1}{2}}|D|^{\frac{-r(l+1)}{2+2r}})$. Thus, the derived rate is faster than DLSRank in Theorem 1 if $m \ge |D|^{\frac{rl}{(r+1)(l+1)-r}}$. It is clear that DLSRank-C is always faster than DLSRank for any m when $l \to \infty$.*

Remark 3 *It should be noticed that our analysis roots in the operator approximation techniques, which is essentially different from the learning theory analysis for pairiwise ranking via capacity-based concentration estimation (Agarwal et al. 2005; Clémencon, Lugosi, and Vayatis 2008; Rudin and Schapire 2009; Rudin 2009; Rejchel 2012) and algorithmic stability (Cossock and Zhang 2008; Agarwal and Niyogi 2009; Chen et al. 2014).*

	Step 1	Step 2	Step 3	Step 4	Step 5
Training Flow (INDV)	$\mathcal{O}(\frac{\lvert D\rvert^3}{m^3}+m\frac{\lvert D\rvert^2}{m^2}r+\lvert D\rvert)$	$\mathcal{O}(\frac{\lvert D\rvert^2}{m})$	$\mathcal{O}(m\lvert D\rvert)$	$\mathcal{O}(\frac{\lvert D\rvert^2}{m})$	$\mathcal{O}(m\lvert D\rvert)$
Training Flow (TOT)	$\mathcal{O}(\frac{\lvert D\rvert^2 r}{m}+\frac{\lvert D\rvert^3}{m^3}+\frac{l\lvert D\rvert^2}{m}+lm\lvert D\rvert)$				
Testing Flow (INDV)	$\frac{\lvert D'\rvert\lvert D\rvert r}{m}+m\lvert D'\rvert$	$\frac{\lvert D\rvert^2}{m^2}+\frac{\lvert D'\rvert\lvert D\rvert}{m}$	$m\lvert D'\rvert$	$\frac{\lvert D'\rvert\lvert D\rvert}{m}+\frac{\lvert D\rvert^2}{m^2}+m\lvert D'\rvert$	–
Testing Flow (TOT)	$\mathcal{O}(\frac{r\lvert D\rvert\lvert D'\rvert}{m}+ml\lvert D'\rvert+\frac{l\lvert D\rvert^2}{m^2}+\frac{l\lvert D\rvert\lvert D'\rvert}{m})$				

Table 1: Computational complexities of training flow and testing flow. INDV refers to the complexity of computing each individual step, and TOT refers to the total complexities of training (or testing) flow.

Training and Testing Flows for DLSRank-C

In practice, it is difficult to implement DLSRank-C directly because Equation 6, involving operator representations of $G_{D_j,\lambda}$ and $f^{l-1}_{D,\lambda}$, cannot be calculated directly without the given data $D_j, j = 1, ...$ Consequently, we employ an efficient strategy, inspired by (Lin, Wang, and Zhou 2020), to learn DLSRank-C with the help of input data. We give some necessary notations for the training and testing flows.

Notations: The training set $D = \cup_{j=1}^m D_j = \cup_{j=1}^m \{\mathbf{x}_i^j, y_i^j\}_{i=1}^{|D_j|}$ and test set $D' = \{\mathbf{x}'_i, y'_i\}_{i=1}^{|D'|}$ are independently drawn from an unknown distribution ρ. Let $D_j(\mathbf{x}) = \{\mathbf{x}^j : (\mathbf{x}^j, y^j) \in D_j\}$ and the kernel matrix $\mathbf{K}_{D_k,D_j} = (K(\mathbf{a},\mathbf{b}))_{\mathbf{a}\in D_k(\mathbf{x}),\mathbf{b}\in D_j(\mathbf{x})} \in \mathbb{R}^{|D_k|\times|D_j|}$ for any $k, j = 1, \cdots, m$. Additionally, we denote $W_{D_j} = \mathbb{I}_{|D_j|} - \frac{1}{|D_j|}\mathbf{1}_{|D_j|}\mathbf{1}^T_{|D_j|} \in \mathbb{R}^{|D_j|\times|D_j|}$ and $M_{D_j,\lambda} = (W_{D_j}\mathbf{K}_{D_j,D_j} + \frac{\lambda|D_j|}{2}\mathbb{I}_{|D_j|})^{-1}$ for $j = 1, \cdots, m$.

Training flow: The training flow for DLSRank-C can be broken down into following five steps:

Step 1 (Initialization): On the j-th local machine, we obtain $\bar{\alpha}_{D_j} = M_{D_j,\lambda}W_{D_j}Y_{D_j}$ with data D_j for $j = 1, \cdots, m$. Then we communicate j-th dataset $D_j(\mathbf{x})$ to the k-th local machine and store $\mathbf{K}_{D_k,D_j}$ for $k = 1, \cdots, m$. Finally, we can initialize m synthesized global vectors

$$\bar{f}^0_{D,\lambda}(D_k(\mathbf{x})) = \sum_{j=1}^m \frac{|D_j|^2}{\sum_{t=1}^m |D_t|^2}\bar{f}_{D_j,\lambda}(D_k(\mathbf{x}))$$

with $k = 1, \cdots, m$, where $\bar{f}_{D_j,\lambda}(D_k(\mathbf{x})) = \mathbf{K}_{D_k,D_j}\bar{\alpha}_{D_j}$.

In the following steps, we update the global vectors iteratively. For $l = 1, 2, \cdots$, we distribute $\bar{f}^{l-1}_{D,\lambda}(D_j(\mathbf{x}))$ to the j-th local machine for $j = 1, \cdots, m$.

Step 2 (On each local machine): On the j-th local machine, we compute m local gradient vectors

$$G^{l-1}_{D_j,\lambda,\bar{f}^{l-1}_{D,\lambda}(D_k(\mathbf{x}))} := \frac{4\mathbf{K}_{D_k,D_j}}{|D_j|}[W_{D_j}(\bar{f}^{l-1}_{D,\lambda}(D_j(\mathbf{x})) - Y_{D_j})] + 2\lambda\bar{f}^{l-1}_{D,\lambda}(D_k(\mathbf{x}))$$

for each $k = 1, \cdots, m$ and then communicate these gradient vectors to global machine.

Step 3 (On global machine): Based on the local gradient vectors, we obtain m global gradient vectors

$$G^{l-1}_{D,\lambda,\bar{f}^{l-1}_{D,\lambda}(D_k(\mathbf{x}))} := \sum_{j=1}^m \frac{|D_j|^2}{\sum_{t=1}^m |D_t|^2} G^{l-1}_{D_j,\lambda,\bar{f}^{l-1}_{D,\lambda}(D_k(\mathbf{x}))}$$

with $k = 1, \cdots, m$, and distribute these m global gradient vectors to all local machines.

Step 4 (On each local machine): Recall the proposed DLSRank-C in (6), we have the following equivalent transformation

$$H^{-1}_{D_j,\lambda}G_{D,\lambda,\bar{f}^{l-1}_{D,\lambda}} = \frac{1}{2\lambda}[\mathbb{I} - 4H^{-1}_{D_j,\lambda}L_{K,D_j}]G_{D,\lambda,\bar{f}^{l-1}_{D,\lambda}}.$$

As a result, on the j-th local machine, we denote

$$\bar{\gamma}^{l-1}_{D,D_j} := \frac{1}{4}M_{D_j,\lambda}W_{D_j}G^{l-1}_{D,\lambda,\bar{f}^{l-1}_{D,\lambda}(D_j(\mathbf{x}))}.$$

According to the Equation (7) in (Lin, Wang, and Zhou 2020), we can obtain

$$H^{-1}_{D_j,\lambda}L_{K,D_j}G^{l-1}_{D,\lambda,\bar{f}^{l-1}_{D,\lambda}(D_k(\mathbf{x}))} = K_{D_k,D_j}\bar{\gamma}^{l-1}_{D,D_j}$$

and

$$\bar{F}^{l-1}_{D_j,\lambda}(D_k(\mathbf{x})) := \frac{1}{2\lambda}[G^{l-1}_{D,\lambda,\bar{f}^{l-1}_{D,\lambda}(D_k(\mathbf{x}))} - 4K_{D_k,D_j}\bar{\gamma}^{l-1}_{D,D_j}]$$

for $k = 1, \cdots, m$. Finally, we transmit these m vectors to the global machine.

Step 5 (On global machine): We update the global vector

$$\bar{f}^l_{D,\lambda}(D_k(\mathbf{x})) = \bar{f}^{l-1}_{D,\lambda}(D_k(\mathbf{x})) - \sum_{j=1}^m \frac{|D_j|^2}{\sum_{t=1}^m |D_t|^2}\bar{F}^{l-1}_{D_j,\lambda}(D_k(\mathbf{x})),$$

for $k = 1, \cdots, m$. Then, we transmit $\bar{f}^l_{D,\lambda}(D_k(\mathbf{x}))$ to j-th local machine with $j = 1, \cdots, m$ and go back to *Step 2*.

Testing flow: Given $|D'|$ query points $D'(\mathbf{x})$, we can obtain the test vector by following steps:

Step 1 (Initialization): On the global machine, we initialize the test vector

$$\bar{f}^0_{D,\lambda}(D'(\mathbf{x})) = \sum_{j=1}^m \frac{|D_j|^2}{\sum_{t=1}^m |D_t|^2}\mathbf{K}_{D',D_j}\bar{\alpha}_{D_j}.$$

In the following steps, we update the test vectors iteratively. For $l = 1, 2, \cdots$, distribute $\bar{f}^{l-1}_{D,\lambda}(D'(\mathbf{x}))$ to m local machines.

Step 2 (On each local machine): On the j-th local machine, compute the local gradient vector for $j = 1, \cdots, m$,

$$G^{l-1}_{D_j,\lambda,\bar{f}^{l-1}_{D,\lambda}(D'(\mathbf{x}))} = \frac{4\mathbf{K}_{D',D_j}}{|D_j|}[W_{D_j}(\bar{f}^{l-1}_{D,\lambda}(D_j(\mathbf{x})) - Y_{D_j})] + 2\lambda\bar{f}^{l-1}_{D,\lambda}(D'(\mathbf{x})),$$

	$p=3$			$p=5$		
	$m=60$	$m=120$	$m=240$	$m=60$	$m=120$	$m=240$
LSRank	0.0206(0.0043)	0.0206(0.0043)	0.0206(0.0043)	0.0195(0.0032)	0.0195(0.0032)	0.0195(0.0032)
Executive time (s)	27.443(0.8514)	27.443(0.8514)	27.443(0.8514)	26.734(1.0449)	26.734(1.0449)	26.734(1.0449)
DLSRank	0.0216(0.0039)	0.0227(0.0040)	0.0244(0.0033)	0.0208(0.0027)	0.0224(0.0025)	0.0244(0.0029)
Executive time (s)	0.0135(0.0002)	0.0068(0.0001)	0.0050(0.0002)	0.0142(0.0004)	0.0073(0.0002)	0.0053(0.0002)
DLSRank-C($l=2$)	0.0208(0.0042)	0.0215(0.0041)	0.0261(0.0056)	0.0201(0.0032)	0.0240(0.0040)	0.0422(0.0090)
Executive time (s)	0.0176(0.0003)	0.0094(0.0002)	0.0073(0.0003)	0.0183(0.0004)	0.0099(0.0003)	0.0075(0.0003)
DLSRank-C($l=4$)	0.0207(0.0042)	0.0208(0.0042)	0.0215(0.0035)	0.0194(0.0029)	0.0196(0.0029)	0.0202(0.0019)
Executive time (s)	0.0259(0.0005)	0.0144(0.0004)	0.0119(0.0006)	0.0267(0.0007)	0.0150(0.0005)	0.0121(0.0005)
DLSRank-C($l=8$)	0.0206(0.0042)	0.0208(0.0042)	0.0214(0.0035)	0.0194(0.0029)	0.0196(0.0029)	0.0199(0.0021)
Executive time (s)	0.0426(0.0009)	0.0246(0.0008)	0.0210(0.0011)	0.0434(0.0011)	0.0253(0.0009)	0.0213(0.0011)

Table 2: The comparison of misranking risk (standard deviation) and training executive time (standard deviation)

where $\bar{f}_{D,\lambda}^{l-1}(D_j(\mathbf{x}))$ is obtained from the training flow. Then we transmit $G^{l-1}_{D_j,\lambda,\bar{f}_{D,\lambda}^{l-1}(D'(\mathbf{x}))}$ with $j=1,\cdots,m$ to the global machine.

Step 3 (On global machine): Compute the global gradient vector

$$G^{l-1}_{D,\lambda,\bar{f}_{D,\lambda}^{l-1}(D'(\mathbf{x}))}=\sum_{j=1}^{m}\frac{|D_j|^2}{\sum_{t=1}^{m}|D_t|^2}G^{l-1}_{D_j,\lambda,\bar{f}_{D,\lambda}^{l-1}(D'(\mathbf{x}))}.$$

Step 4 (On global machine): We obtain the final vector of prediction

$$\begin{aligned}&\bar{f}_{D,\lambda}^{l}(D'(\mathbf{x}))\\&=\bar{f}_{D,\lambda}^{l-1}(D'(\mathbf{x}))-\frac{1}{2\lambda}\sum_{j=1}^{m}\frac{|D_j|^2}{\sum_{t=1}^{m}|D_t|^2}[G^{l-1}_{D,\lambda,\bar{f}_{D,\lambda}^{l-1}(D'(\mathbf{x}))}\\&\qquad-4\mathbf{K}_{D',D_j}\bar{\gamma}^{l-1}_{D,D_j}]\\&=\frac{2}{\lambda}\sum_{j=1}^{m}\frac{|D_j|^2}{\sum_{t=1}^{m}|D_t|^2}[\mathbf{K}_{D',D_j}\bar{\gamma}^{l-1}_{D,D_j}-\\&\qquad\frac{\mathbf{K}_{D',D_j}}{|D_j|}W_{D_j}(\bar{f}_{D,\lambda}^{l-1}(D_j(\mathbf{x}))-Y_{D_j})].\end{aligned}$$

Remark 4 *Here we compare the computational complexities of LSRank and DLSRank-C. Let r be the complexity of computing a kernel function $K(\cdot,\cdot)$. It is trivial to obtain the computational complexity of LSRank, i.e., training complexity $\mathcal{O}(|D|^2r+|D|^3)$ and testing complexity $\mathcal{O}(|D||D'|r)$. Without considering the time of data transmission in DLSRank-C, the computational complexity analysis of DLSRank-C for the training flow and testing flow is summarized in Table 1.*

Empirical Evaluations

In this section, we evaluate DLSRank-C on some simulated and benchmark datasets to validate our theoretical findings. All experiments were implemented in MATLAB 2019b on an intel Core i7 with 16 GB memory.

Empirical Evaluation on Simulated Data

Inspired by numerical experiments in (Kriukova, Pereverzyev, and Tkachenko 2016), the inputs $\{\mathbf{x}_i\}_{i=1}^{|D|}\in\mathbb{R}^{|D|\times p}$ are randomly chosen from natural number set $\{1,\cdots,100\}$, and the corresponding outputs are

$$y_i=[\|\mathbf{x}_i\|/5]+\epsilon_i,\ \ 1\le i\le|D|,$$

where $[\cdot]$ means the integer part of inputs and ϵ is the noise sampled from the uniform distribution $U(-5,5)$. The RKHS is constructed by Gaussian kernel $K(\mathbf{x},\mathbf{x}')=\exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2d^2})$.

We generate 10000 samples for training and 1000 samples for testing. LSRank, as a baseline, is trained on all samples in a batch. We compare our proposed DLSRank-C with LSRank and DLSRank by carrying out various settings (e.g., dimension $p=3,5$, the number of local machines $m=60,120,240$ and iteration times l = 2, 4, 8). We train the model on training set and evaluate the methods on testing data via following averaged misranking risk (Kriukova, Pereverzyev, and Tkachenko 2016), i.e.,

$$\mathcal{R}(f)=\frac{\sum_{i,j=1}^{n}\mathrm{I}_{\{(y_i>y_j)\wedge(\bar{f}(\mathbf{x}_i)\le\bar{f}(\mathbf{x}_j))\}}}{\sum_{i,j=1}^{n}\mathrm{I}_{\{y_i>y_j\}}},$$

where $\mathrm{I}_{\{\varphi\}}$ is 1 if φ is true and 0 otherwise. The regularization parameter λ and bandwidth d are selected in the grids $\{10^{-2},10^{-1},1,10,100\}$ and $\{1,10,10^2,10^3\}$, respectively. Finally, the averaged performance (e.g., misranking risk and executive time) of different methods is given in Table 2. Moreover, Figure 1 shows the relation between misranking risk, different numbers of communications l and the number of local machines m.

From Table 2 and Figure 1, we can conclude the following assertions: a) When m is not too large, the distributed methods (DLSRank and DLSRank-C) are always comparable to original LSRank. There exists an upper bound of m for DLSRank and DLSRank-C respectively, when larger than it, the misranking risk increases and is far from the original LSRank. This verifies the theoretical statement in Theorems 1 and 2; b) DLSRank-C with more communications, compared with DLSRank, can achieve better performance when the number of local machine increases, which verifies the fact in Remark 2 that the bound of m is determined by the communication times. c) DLSRank-C enjoys smaller mis-

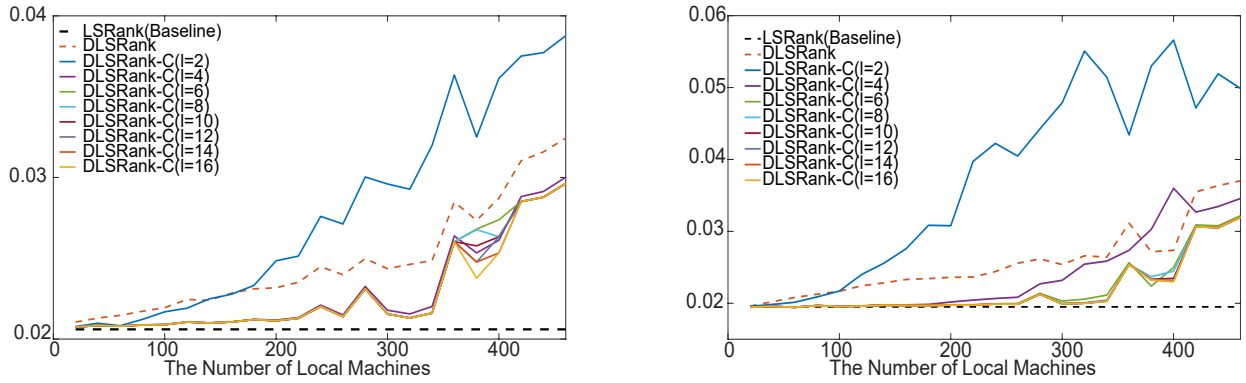

Figure 1: The relation between misranking risk and the number of local machines for fixed numbers of communications. The left figure and right figure are respectively the results on the 3-dimensional data and the 5-dimensional data.

	MovieLens 500-1000			MovieLens 1000-1500		
Method	$m = 60$	$m = 120$	$m = 240$	$m = 60$	$m = 120$	$m = 240$
LSRank	0.4901(0.0162)	0.4901(0.0162)	0.4901(0.0162)	0.4970(0.0207)	0.4970(0.0207)	0.4970(0.0207)
DLSRank	0.4904(0.0178)	0.4907(0.0178)	0.4920(0.0180)	0.4976(0.0192)	0.4979(0.0193)	0.4980(0.0193)
DLSRank-C($l = 2$)	0.4904(0.0179)	0.4905(0.0178)	0.4918(0.0178)	0.4976(0.0193)	0.4972(0.0194)	0.4972(0.0191)
DLSRank-C($l = 4$)	0.4903(0.0179)	0.4904(0.0180)	0.4917(0.0178)	0.4976(0.0192)	0.4972(0.0192)	0.4970(0.0192)
DLSRank-C($l = 8$)	0.4901(0.0177)	0.4901(0.0177)	0.4915(0.0176)	0.4976(0.0192)	0.4972(0.0192)	0.4970(0.0191)

Table 3: The comparison of misranking risk (standard deviation) on testing data

ranking risk than DLSRank when the number of local machines is smaller than the upper bound of m, which verifies the effectiveness of communication strategy. Moreover, we investigate the impact of the parameters (e.g., dimension p and the number of training set) on the ranking performance in the *Supplementary Material*.

Empirical Evaluation on Real-world Data

In daily life, we need some recommendations from other people to look for the meaningful movies. For a moviegoer, the goal here is to produce a list of unseen movies ordered by the predicted preference. For this purpose, we employ DLSRank-C on the movie recommendation task and compare its performance with other competitors such as DLSRank and LSRank. All data used here are freely available at: http://www.grouplens.org/taxonomy/term/14.

The MovieLens dataset consists of 25000095 anonymous ratings of 62423 movies made by 162541 MovieLens users. The rating score is represented with an integer from the set $\{1, 2, 3, 4, 5\}$. The dataset is a 62423×162541 rating matrix where (i, j) entry is the rating score of the j-th reviewer on the i-th movie.

Inspired by the experimental set-up in (Cortes, Mohri, and Rastogi 2007; Freund et al. 2003; Zhou et al. 2016), we grouped the reviewers into 500 – 1000 and 1000 – 1500 movies according to the number of movies they have rated. Five-hundred reference reviewers were selected at random from one of the two groups. Moreover, the test reviewers are selected from those users who had rated more than 5000 movies. For a given test reviewer we only kept those rows in the rating matrix corresponding to the movies reviewed by the test reviewer. These columns and rows corresponding to 500 reference reviewers constituted a submatrix of size at least 5000×501, where the last column corresponds to the test reviewer. In fact, this rating matrix is very sparse. Then those movies rated by none of the reference reviewers and those reference reviewers who did not rate any movie viewed by the test reviewer were removed. Finally, missing review values of every left movie were replaced with the median review score of those left reference reviewers on this movie. We then obtain a smaller submatrix. Each row of it formed a data pair $(\mathbf{x}_i, y_i)$ and the last entry was the label y_i of the input features $\mathbf{x}_i$. The data we got were divided into two parts at random according to the ratio $8 : 2$. One part was used for training and the rest part for test. We repeat the experiment for 10 times and present the average results in Table 3. These results show effectiveness of the communication strategy for DLSRank.

Conclusion

This paper proposed a new distributed ranking model with communication strategy. Theoretical analysis is provided to demonstrate its promising learning rate. Empirical examples verify its competitive performance compared with the DLSRank without communication strategy.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant Nos. 11671161, 12071166,

62076041, 61702057, 61806027 and 61972188. We are grateful to the anonymous AAAI reviewers for their constructive comments.

References

Agarwal, S.; Graepel, T.; Herbrich, R.; Har-Peled, S.; and Roth, D. 2005. Generalization bounds for the area under the ROC curve. *J. Mach. Learn. Res.* 6: 393–425.

Agarwal, S.; and Niyogi, P. 2009. Generalization bounds for ranking algorithms via algorithmic stability. *J. Mach. Learn. Res.* 10: 441–474.

Aronszajn, N. 1950. Theory of reproducing kernels. *Tran. Am. Math. Soc.* 68: 337–404.

Chang, X.; Lin, S.-B.; and Zhou, D.-X. 2017. Distributed semi-supervised learning with kernel ridge regression. *J. Mach. Learn. Res.* 18: 1–22.

Chen, H. 2012. The convergence rate of a regularized ranking algorithm. *Journal of Approximation Theory* 164: 1513–1519.

Chen, H.; Li, H.; and Pan, Z. 2019. Error analysis of distributed least squares ranking. *Neurocomputing* 361: 222–228.

Chen, H.; Peng, J.; Zhou, Y.; Li, L.; and Pan, Z. 2014. Extreme learning machine for ranking: Generalization analysis and applications. *Neural Networks* 53: 119–126.

Chen, H.; Tang, Y.; Li, L.; Yuan, Y.; Li, X.; and Tang, Y.-Y. 2013. Error analysis of stochastic gradient descent ranking. *IEEE Transactions on Cybernetics* 43(3): 898–909.

Clémencon, S.; Lugosi, G.; and Vayatis, N. 2008. Ranking and empirical minimization of U-statistics. *Ann. Statist.* 36: 844–874.

Cortes, C.; Mohri, M.; and Rastogi, A. 2007. Magnitude-preserving ranking algorithms. In *Proc. 24th Int. Conf. Mach. Learn.(ICML)*, 169–176.

Cossock, D.; and Zhang, T. 2008. Statistical analysis of Bayes optimal subset ranking. *IEEE Transactions on Information Theory* 54(11): 5140–5154.

Cucker, F.; and Zhou, D. X. 2007. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge, U.K.: Cambridge Univ. Press.

Freund, Y.; Iyer, R.; Schapire, R.; and Singer, Y. 2003. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.* 4: 933–969.

Guo, X.; Hu, T.; and Wu, Q. 2020. Distributed Minimum Error Entropy Algorithms. *Journal of Machine Learning Research* 21: 1–31.

Guo, Z.-C.; Lin, S.-B.; and Shi, L. 2019. Distributed learning with multi-penalty regularization. *Appl. Comput. Harmon. Anal.* 46: 478–499.

Guo, Z.-C.; Shi, L.; and Wu, Q. 2017. Learning theory of distributed regression with bias corrected regularization kernel network. *J. Mach. Learn. Res.* 18: 1–25.

Hsieh, C.-J.; Si, S.; and Dhillon, I. S. 2014. A divide-and-conquer solver for kernel support vector machines. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*.

Hu, T.; Fan, J.; Wu, Q.; and Zhou, D. 2013. Learning theory approach to minimum error entropy criterion. *Journal of Machine Learning Research* 14(1): 377–397.

Hu, T.; Wu, Q.; and Zhou, D.-X. 2020. Distributed kernel gradient descent algorithm for minimum error entropy principle. *Appl. Comput. Harmon. Anal.* 49: 229–256.

Huang, C.; and Huo, X. 2019. A distributed one-step estimator. *Mathematical Programming* 174(1): 41–76.

Kriukova, G.; Panasiuk, O.; Pereverzyev, S. V.; and Tkachenko, P. 2016. A linear functional strategy for regularized ranking. *Neural Networks* 73: 26–35.

Kriukova, G.; Pereverzyev, S. V.; and Tkachenko, P. 2016. On the convergence rate and some applications of regularized ranking algorithms. *Journal of Complexity* 33: 14–29.

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

Lin, S.-B.; Guo, X.; and Zhou, D.-X. 2017. Distributed learning with regularized least squares. *J. Mach. Learn. Res.* 18: 1–31.

Lin, S.-B.; Wang, D.; and Zhou, D.-X. 2020. Distributed Kernel Ridge Regression with Communications. *Journal of Machine Learning Research* 21: 1–38.

Rejchel, W. 2012. On ranking and generalization bounds. *J. Mach. Learn. Res.* 13: 1373–1392.

Rosasco, L.; Belkin, M.; and Vito, E. 2010. On learning with integral operators. *Journal of Machine Learning Research* 11: 905–934.

Rudin, C. 2009. The P-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *J. Mach. Learn. Res.* 10: 2233–2271.

Rudin, C.; and Schapire, R. E. 2009. Margin-based ranking and an equivalence between AdaBoost and RankBoost. *J. Mach. Learn. Res.* 10: 2193–2232.

Smale, S.; and Zhou, D. X. 2005. Shannon sampling: Connections to learning theory. *Applied and Computational Harmonic Analysis* 19: 285–302.

Smale, S.; and Zhou, D. X. 2007. Learning theory estimates via integral operators and their approximations. *Constr. Approx.* 26: 153–172.

Sun, H.; and Wu, Q. 2009. A note on applications of integral operator in learning theory. *Applied and Computational Harmonic Analysis* 26(3): 416–421.

Xu, C.; Zhang, Y.; Li, R.; and Wu, X. 2016. On the feasibility of distributed kernel regression for big data. *IEEE Transactions on Knowledge and Data Engineering* 28: 3041–3052.

Zeng, J.; and Yin, W. 2018. On nonconvex decentralized gradient descent. *IEEE Transactions on Signal Processing* 66: 2834–2848.

Zhang, Y.; Duchi, J.; and Wainwright, M. 2015. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. *J. Mach. Learn. Res.* 1: 1–40.

Zhou, Y.; Chen, H.; Lan, R.; and Pan, Z. 2016. Generalization performance of regularized ranking with multiscale kernels. *IEEE Trans. Neural Netw. Learning Syst.* 27(5): 993–1002.