第48卷第4期 中国科学技术大学学报 Vol.48,No.4
2018年4月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Apr. 2018

文章编号:0253-2778(2018)04-0284-06

基于数据划分的核岭回归加速算法

刘恩江[1],宋云胜[1],梁吉业[1,2]

(1.山西大学计算机与信息技术学院,山西太原 030006;
2.计算智能与中文信息处理教育部重点实验室,山西太原 030006)

摘要:核岭回归(KRR)是一种重要的回归算法,具有可解释性、强泛化性能等优点,被广泛应用于模式识别、数据挖掘等领域;然而面对大规模数据时,核岭回归存在着训练效率较低的缺陷.为此,利用分而治之思想提出一种基于数据划分的核岭回归加速算法(PP-KRR).首先利用一簇平行超平面将当前数据所在的空间划分为 m 个互不相交的区域;其次在划分后的每个区域上训练 KRR 模型;最后每个 KRR 模型预测处在同一区域内的未标记实例.在真实数据集上与传统的算法进行实验比较分析,实验结果表明,提出的算法在保持一定预测精度的同时,能够获得更短的训练时间.

关键词:核岭回归;分而治之;平行分割;主成分分析

中图分类号:TP391 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2018.04.003

引用格式:刘恩江,宋云胜,梁吉业. 基于数据划分的核岭回归加速算法[J]. 中国科学技术大学学报,2018,48(4):284-289.

LIU Enjiang, SONG Yunsheng, LIANG Jiye. An accelerator for kernel ridge regression algorithms based on data partition[J]. Journal of University of Science and Technology of China, 2018,48(4):284-289.

An accelerator for kernel ridge regression algorithms based on data partition

LIU Enjiang[1], SONG Yunsheng[1], LIANG Jiye[1,2]

(1. *School of Computer and Information Technology, Shanxi University, Taiyuan* 030006, *China*;
2. *Key Laboratory of Computational Intelligence and Chinese Information Processing, Taiyuan* 030006, *China*)

Abstract: Kernel ridge regression (KRR) is an important regression algorithm widely used in pattern recognition and data mining for its interpretability and strong generalization capability. However, it has the defect of low training efficiency when faced with large-scale data. To address this problem, an accelerating algorithm is proposed which uses the concept of divide-and-conquer for kernel ridge regression based on data partition (PP-KRR). Firstly, the current training data space is divided into m mutually disjoint regions by a bunch of parallel hyperplanes. Secondly, each KRR model is trained on each region respectively. Finally, each unlabeled instance is predicted by the KRR model within the same region. Comparisons with three traditional algorithms on real datasets show that the proposed algorithm obtains similar prediction accuracy with less training time.

Key words: kernel ridge regression; divide-and-conquer; parallel partition; principal component analysis

收稿日期:2017-05-23;**修回日期**:2017-06-24

基金项目:国家自然科学基金重点项目(61432011, U1435212)资助.

作者简介:刘恩江,男,1993年生,硕士研究生,研究方向:机器学习与数据挖掘. E-mail:2510087270@qq.com

通讯作者:梁吉业,博士/教授. E-mail:ljy@sxu.edu.cn

0　引言

核岭回归[1]是一种基于核方法的回归算法，它利用核函数将原始数据映射到高维空间，并在此空间建立岭回归模型.与传统的岭回归算法相比，映射后的数据在高维空间往往呈线性关系，故在此高维空间建立岭回归模型具有较强的泛化性能.由于核岭回归算法计算复杂度为 $O(N^3)$，故难以有效地处理大规模数据，其中 N 为样本数.

为了提升核岭回归算法处理大规模数据的效率，学者提出了一系列改进型算法.现有研究主要分为两个方向：核矩阵近似与基于数据划分.

核矩阵近似可分为以下 3 类.第 1 类是基于低秩矩阵近似的方法.其中典型的有 Scholkopf 提出的核 PCA[2]，Fine 等提出的不完全 Cholesky 分解[3]，Williams 等提出的 Nyström sampling 等算法[4-7].这类方法降低计算时间复杂度至 $O(d\,n^2)$ 或 $O(n\,d^2)$，这里 $d \ll n$，d 代表矩阵的秩.第 2 类是随机特征近似的方法.Rahimi 等[8]提出了运用随机傅里叶映射一致逼近平移不变核，用相对低维的显示特征映射来逼近高维的隐示特征，从而降低复杂度.第 3 类关于核矩阵近似算法是从迭代梯度法求解核矩阵拓展的.通过提前终止迭代优化算法，包括 Yao 等[9]提出的梯度下降和 Blanchard 等[10]提出的共轭梯度下降均为利用正则化参数提前停止迭代，提高效率.对超大规模数据集（$N > 10^6$），基于核矩阵近似的算法已不再适用.

基于数据划分的方法从数据的角度考虑，可以有效地改进现有大多数机器学习算法，使之适用于并行或分布式计算环境.具体而言：首先对数据集进行分解，然后在每个子集上训练模型，最后将各模型融合.现有数据集划分的方式主要有两种：随机划分[11]与基于聚类的划分[12-14].随机划分是每个子集中的元素从原始数据集中无放回随机抽样得到的，此方法并未考虑数据之间的相关性.基于聚类划分是利用各种聚类算法将数据集分成若干个类，每个类为一个子集，与随机划分相比，它考虑了数据之间的相关性，但是并不能保证每个子集的大小一致，进而导致算法的最终运行时间变长.

本文提出了一种基于数据划分的加速算法.该算法采用一簇平行的超平面将特征空间分割成若干个互不相交的区域，并将落入每个区域内的样本作为一个子集，每个子集规模大致相等.在每个区域上分别训练核岭回归模型.对给定的未标记样本，首先判定其落入划分后的区域，然后利用此区域对应的模型进行预测.该方法不仅保证了子集大小的平衡性，同时保持了数据的局部信息.实验结果表明，本文提出的算法大幅提高了核岭回归算法学习效率，且不影响其预测精度.

1　相关工作

随着数据规模越来越大，现有的许多机器学习、数据挖掘算法难以有效应对.核岭回归作为一种基于核方法的回归学习算法，同样面临数据量很大时，训练模型需要的时间长、空间开销大的问题.

在此背景下，适用于大规模数据的 KRR 算法得到了广泛的研究，提高其求解速度和保持性能具有重要的应用价值.针对这一问题，研究人员提出了各种方法对核矩阵进行逼近.其中以低秩矩阵近似[15]为代表的一批相关算法被成功应用到各种核学习算法当中.低秩矩阵是对原始矩阵特征分解后求出的前 d 阶特征值及其相对应的特征向量所构成的近似矩阵，这里 $d \ll n$，低秩矩阵的求解有许多快速算法，因此如何构造一个低秩矩阵就是算法的核心.Nyström sampling[4]算法是一种代表性算法.该算法随机抽取核矩阵的部分列向量去近似原始核矩阵，Alaoui 等[6]则提出了首先根据数据本身信息构造抽样概率，然后依概率抽样得到近似矩阵.这样就大大减少了所需要的时间和内存空间.现实情况中，为了使近似矩阵可以达到良好的性能，抽样概率的构建将很复杂，对于超大规模数据集（$>10^6$），这类算法并不能很好适用.随机特征近似[8]则是另一类核矩阵逼近的方法.该方法随机生成一组映射特征，通过这组特征的线性组合去近似核函数.随机傅里叶映射一致逼近平移不变核就是其中一种有效的算法，利用相对低维的显示特征映射逼近高维的隐示特征，从而降低复杂度.相关研究表明，运用这种方法，KRR 算法可以达到与深度神经网络相媲美的效果[16]，但只有选择足够多的随机特征才能实现该性能，这仍然增加了计算复杂度.

近年来分而治之策略得到了学者的广泛研究[17-20].分治思想就是将大规模的数据集分成多个小规模的子集分别处理，随后再按照相关方式融合得到最终的结果.近几年，分治策略在多种算法上得以实现，并且都有相关的统计性、收敛性分析.在应用到核岭回归的研究中，Zhang 等[11]提出的分而治

之核岭回归(random-KRR)算法,采用随机抽样的方法,将训练集分割为大小相等、互不相交的若干子集,在每个子集分别训练模型,对于未标记的数据,其预测结果为该数据在所有子集模型结果的求和平均.作者对该方法做了统计性证明,在强假设条件下,算法性能可与全局性能相比.因为是随机划分,输入空间并不是互不相交的,所以限制性较强,因此Tandon等[14]提出了空间划分核岭回归(DC-KRR).将输入空间区域化,每个区域独立,每个数据只用自身所属区域预测学习,最终所得到的模型泛化能力增强,使用约束减少,并有相关理论保证.作者使用核 k-means 算法作为空间分割方法,在多个数据集上进行实验,结果均比随机划分的性能要好.由于聚类算法本身速度较慢,虽然在实验中采取了抽样聚类的方法,相比全局算法训练速度加快,但是仍然慢于随机划分方法,因此本文提出了基于平行超平面划分数据集的加速算法,在保持预测精度的同时,训练速度得到了显著提高.

2 预备知识

令 $X=\{(x_i,y_i)\}_{i=1}^n$ 为容量 n 的数据集,$x_i=\{x_{i1},x_{i2},\cdots,x_{ik}\}\in\mathbb{R}^k$ 表示一个训练实例,$y_i\in\mathbb{R}$表示响应值.核岭回归的训练过程就是求解一个如下的最小化问题:

$$\min_{\alpha\in\mathbb{R}^n}\|\boldsymbol{Y}-\boldsymbol{G\alpha}\|_2^2+\lambda(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G\alpha}) \quad (1)$$

式中,$\boldsymbol{G}$ 是一个 $n\times n$ 的矩阵,$G_{ij}=K(x_i,x_j)$,其中 $K(x_i,x_j)$ 是一个核函数,λ 是惩罚参数,$\boldsymbol{Y}=\{y_1,y_2,\cdots,y_n\}$,$\boldsymbol{\alpha}\in\mathbb{R}^n$ 为权向量.

令 $\boldsymbol{\alpha}^*=(\boldsymbol{G}+n\lambda\boldsymbol{I})^{-1}\boldsymbol{Y}$ 为求解上述最小化函数所得最优解.对于给定的为标记的样本 x,其预测值为 $\sum_{i=1}^{n}K(x,x_i)\alpha_i^*$.

为了叙述的方便,我们给出了一个定义.

定义 2.1 给定数据 $X=\{(x_1,y_1),\cdots,(x_n,y_n)\}$,在特征空间 S 上有一个划分 $\{C_1,C_2,\cdots,C_m\}$,如果 $S=\bigcup_{i=1}^m C_i$,$C_i\cap C_j=\varnothing$,$\forall i,j\in\{1,2,\cdots,m\}$,$i\neq j$,则每个样本将属于唯一的一个划分 C_i.

在给定空间划分 $\{C_1,C_2,\cdots,C_m\}$,将落在每个区域 C_i 内的所有数据作为一个子集,便可以将数据集划分为 m 个不相交的子集 $\{D_1,D_2,\cdots,D_m\}$.

3 PP-KRR 算法

PP-KRR 算法是利用分而治之的策略将一个大规模问题转化为若干个较小的子问题.首先将原始数据 D 空间划分为若干子集,并在每个子集上建立相应的核岭回归模型,对未标记的实例判定其所属的子集并利用该子集的模型进行预测.算法伪代码如下:

算法 3.1 PP-KRR 算法

输入:数据集 $D=\{(x_i,y_i)\}_{i=1}^n$,子集个数 m,未标记的样本

输出:未标记样本预测值

步骤 1 将数据集划分为 m 个子集,$\{D_1,D_2,\cdots,D_m\}$

步骤 2 对每个子集分别建立回归模型

步骤 3 判断未标记样本所属的子集,并用该子集上的模型预测

PP-KRR 算法的运行时间主要依赖于划分后的最大子集的大小,因此将数据集划分为若干个容量大小近似相等的子集.基于核岭回归算法本身的特性,数据的划分保证了数据的局部几何性质.换言之,在原始数据集中互为近邻的数据点在划分后仍然保持这种近邻关系.为了满足以上两个要求,本文提出了一种基于平行超平面的数据划分算法(PHP).

3.1 PHP 算法

我们利用若干个平行的超平面将数据的输入空间划分为若干个子区域,并将落入每个子区域的样本点作为一个子集.采用超平面划分数据集是因为这种划分方式简单易于操作,并且划分速度快.同时利用向量 $\boldsymbol{w}$ 将数据空间映射至一维空间后,原始空间中相近的点仍然在同一区域内,保持其相近关系.PHP 算法主要分为以下两个步骤:

(Ⅰ)方向向量的选择

虽然超平面划分方式可以保证数据的近邻性质,但是采用不同的超平面,得到的划分区域之间的差异性也不同.为了满足各子区域数据之间方差最大,本文利用主成分分析(principal component analysis, PCA)获取划分向量.

主成分分析是一种有效的降维算法,按照最大方差理论,对于一个数据集,其少量的主成分就可以包含该数据85%以上的信息,而第一主成分作为数据特征的线性组合,会包含数据集的大部分信息.因为第一主成分是数据方差最大的方向,按照第一主成分划分,会使得划分后的区域差异性最大,对于每个数据在各自区域内的近似是最佳的,因此采用第一主成分作为划分向量在保证信息损耗最小的前提

下，最大限度地使区域相差最大.

$$\boldsymbol{w}=\arg\max_{\boldsymbol{w}}\boldsymbol{w}^{\mathrm{T}}\hat{\sum}\boldsymbol{w}\quad \text{s.t.}\|\boldsymbol{w}\|_2=1 \tag{2}$$

式中，$\hat{\sum}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 是数据的协方差矩阵，$\boldsymbol{w}$ 是属于 $\hat{\sum}$ 的特征向量.

(Ⅱ)划分数据集

按照上述求得向量 w，对于数据集 X，我们可以得到一个有序序列，$w\cdot x_{1'}\leqslant w\cdot x_{2'}\leqslant\cdots\leqslant w\cdot x_{n'}$，定义 b_p 点能分割区间 $[w\cdot x_{1'},w\cdot x_{n'}]$ 到 m 个子区间 $[b_{p-1},b_p]$，$p=1,\cdots,m$，则

$$b_p=\begin{cases}w\cdot x_{1'},p=0\\ \dfrac{1}{2}\{w\cdot x_{P*\lceil\frac{n}{m}\rceil}+w\cdot x_{P*\lceil\frac{n}{m}\rceil+1}\},1\leqslant p\leqslant m\\ w\cdot x_{n'},p=m\end{cases} \tag{3}$$

式中，$\lceil x\rceil$ 为靠近 x 的最大整数，因此数据集所在的特征空间可以被划分为 m 个子区域 $\{D_1,D_2,\cdots,D_m\}$，其中 $|D_p|=n_p$，$\sum_{p=1}n_p=n$，且

$$D_p=\begin{cases}\{x\mid w\cdot x\leqslant b_1,x\in X\},p=1\\ \{x\mid b_{p-1}<w\cdot x\leqslant b_p,x\in X\},2\leqslant p<m\\ \{x\mid w\cdot x>b_m,x\in X\},p=m\end{cases} \tag{4}$$

3.2　时间复杂度分析

PP-KRR 算法主要由数据划分和在每个子集上的训练模型构成.数据划分算法主要分为两个阶段，第一阶段是求方向向量 $\boldsymbol{w}$.因为方向向量是基于 PCA 算法所得，而 PCA 算法的时间复杂度是 $O(nk^2)$，第二阶段数据集划分的时间复杂度是 $O(n)$.根据核岭回归算法的时间复杂度可知，在每个大小大致为 n/m 子集上的模型训练复杂度为 $O(n^3/m^3)$，因此总的算法训练过程为在每个子集上训练的 m 倍，即 $O(n^3/m^2)$.

4　实验及结果分析

4.1　实验设置

为了检验本文算法的性能，在真实数据集上进行实验，并与 random-KRR，DC-KRR 以及全局 whole-KRR 算法进行比较，其中 DC-KRR 采用核 k-means 聚类算法.

4.1.1　数据集

实验选取的是 UCI 数据库中 4 组规模较大的数据集，其详细信息见表 1.不失一般性，我们对数据的每个特征进行了归一化处理.

表 1　数据集描述

Tab.1　Summary of UCI datasets

datasets	# samples	# features	σ
Puma32h	8 192	32	10^{-4}
Cpusmall	8 192	12	10^{-1}
Ccpp	9 568	4	10^{-3}
CT Slice	53 500	385	10^{-2}

4.1.2　超参数设置

实验中我们使用高斯径向基作为核函数

$$K(x,x')=\exp\left(-\frac{\|x-x'\|_2^2}{2\sigma^2}\right) \tag{5}$$

式中，核参数 σ 通过交叉验证选取见表 1.对于正则项参数 λ 的选取，文献[11]进行过实验，对于每个子集如果分别设置 λ，则最终的泛化误差比单个子集采用同样的 λ 误差高，这是因为该方法是基于随机抽样的，因此每个子集是全局的抽样近似.用本文算法做同样的实验，对于 λ 不变的情况，我们设置 $\lambda=1/n$；对于分别设置 λ 的情况，我们令 $\lambda=1/n_{\mathrm{sub}}$，其中 n_{sub} 是每个子集的元素个数.由图 1 可知，相比于设置为相同大小的 λ，分别设置 λ 的泛化误差会更小，这是因为我们的每个子训练器是基于单个子集的，因此与全局相关性低.讨论子集个数 m 的设置，对于 3 种算法，我们均设置 $m=2^r$，$r=1,2,\cdots,6$，便于划分数据集.同时在训练过程中改变训练集的样本个数，以便观察数据集大小与划分子集个数多少对测试误差的影响.实验结果如图 2 和表 2，3 所示.

表 2　真实数据集上测试误差

Tab.2　Test error on real datasets

datasets	# m	KRR	PP-KRR	random-KRR	DC-KRR
Puma32h	16	4.06	3.60	5.05	3.32
Cpusmall	16	3.21	3.22	4.37	3.15
Ccpp	32	3.914	3.945	3.922	3.852
CT Slice	64	3.72	6.42	13.42	6.51

表 3　真实数据集上训练时间

Tab.3　Training time on real datasets

datasets	# m	whole-KRR	PP-KRR	random-KRR	DC-KRR
Puma32h	16	19.8	0.3	0.4	5.7
Cpusmall	16	19.7	0.3	0.3	4.37
Ccpp	32	43.8	0.4	0.5	8.3
CT Slice	64	393.5	7.3	7.6	133.2

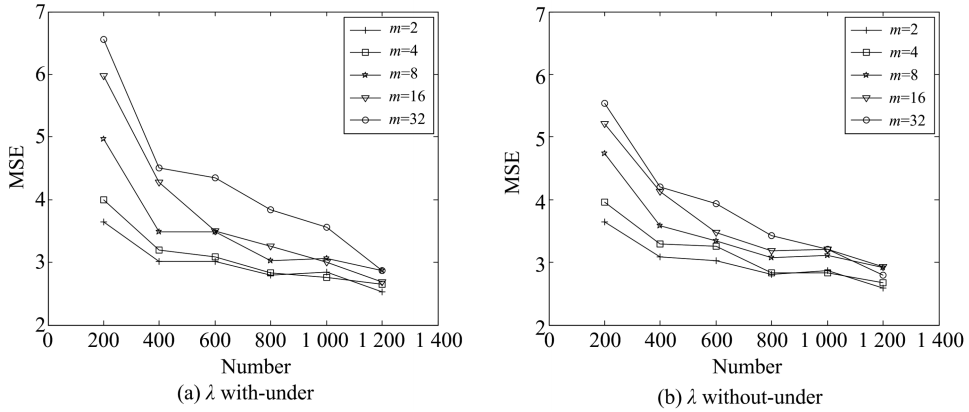

图 1 λ 值对误差的影响

Fig.1 The effect of λ on the error

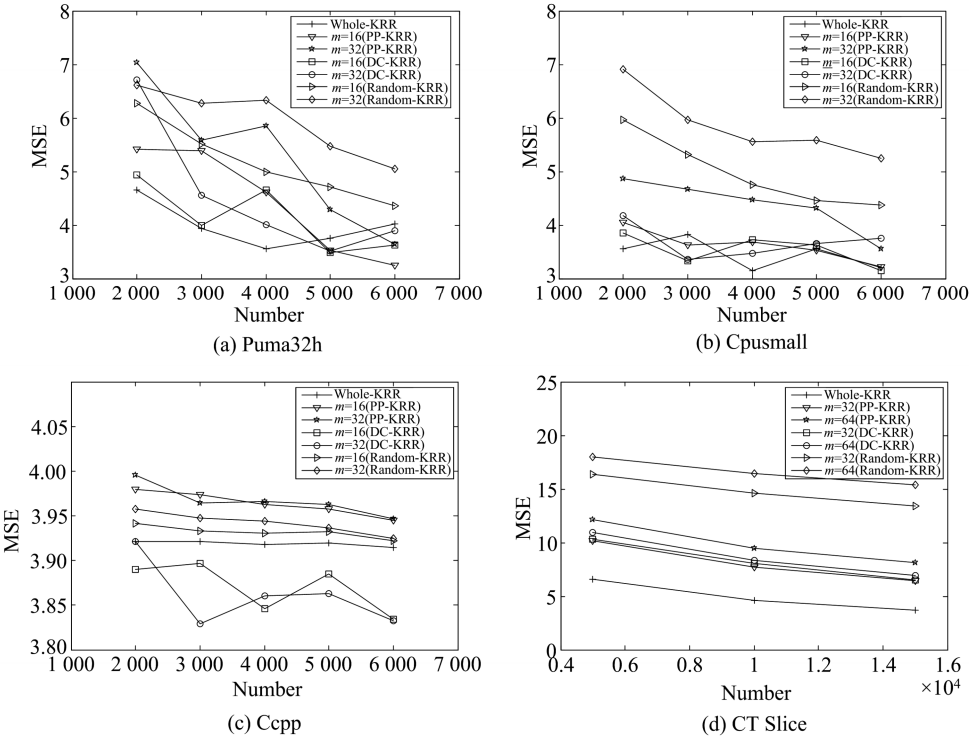

图 2 在四个数据集上的测试误差曲线

Fig.2 Test error curves on four data sets

为了验证算法的有效性,我们选择均方根误差(MSE)作为评价指标,同时比较了全部方法的训练时间.

4.2 实验结果

由表 2 和图 2 可知本文所提出的算法在全部数据集上都可以得到与 whole-KRR 算法相接近的测试误差,在测试精度上优于 random-KRR 算法,仅略逊于 DC-KRR 算法.本文所提算法在 Puma32h 数据集上超过了 whole-KRR 算法,这可能是因为该数据集是一种分段数据集,数据集有较强的近邻特性,DC-KRR 算法也显示了该特征.在 Cpusmall 数据集上本文所提出的算法和 DC-KRR 算法均达到了 whole-KRR 算法的精度,而 random-KRR 则明显误差增大.在 Ccpp 数据集上,全部算法并无太大差异,在最大的 CT Slice 数据集上,本文的算法性能也是最接近 whole-KRR 算法误差的.

同时由表 3 可知,本文算法的训练时间与

random-KRR 几乎一致，均对 KRR 算法实现了较大的加速.相较于 DC-KRR 算法也提升明显，尤其是随着数据集的增大更为显著.由于本文所采用的是基于超平面的数据划分方法，在区域划分上与可同时适用于低维和高维数据，划分速度近似于随机划分，大大提高了算法效率.

通过图 2，我们发现 PP-KRR 算法可能在前后相邻划分时出现第 1 次划分性能良好、第 2 次性能波动较大的情况，这说明单一主方向的选取可能会丢失部分信息.

5　结论

本文针对核岭回归算法训练效率低的问题，以分而治之的策略，利用多个平行的超平面，将大规模数据集划分为若干个容量近似相等的子集，并在每个子集上训练核岭回归模型.该数据划分方法不仅保证了数据的局部性质，而且有效地缩短了训练时间.实验结果表明，该算法在真实数据上获得了较好的预测精度.

参考文献(References)

[1] ROSIPAL R. Kernel-based regression and objective nonlinear measures to assess brain functioning[D]. Scotland: University of Paisley, 2001.

[2] SCHÖLKOPF B, SMOLA A, MÜLLER K R. Nonlinear component analysis as a kernel eigenvalue problem[J]. Neural computation, 1998, 10(5): 1299-1319.

[3] FINE S, SCHEINBERG K. Efficient SVM training using low-rank kernel representations[J]. Journal of Machine Learning Research, 2001, 2: 243-264.

[4] WILLIAMS C K I, SEEGER M. Using the Nyström method to speed up kernel machines[C]// Proceedings of the 13th Conference on Neural Information Processing Systems. Cambridge: MIT press, 2000: 661-667.

[5] BACH F. Sharp analysis of low-rank kernel matrix approximations[J]. Journal of Machine Learning Research, 2012, 30: 185-209.

[6] ALAOUI A E, MAHONEY M W. Fast randomized kernel ridge regression with statistical guarantees[C]// Proceedings of the 28th Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2015: 775-783.

[7] RUDI A, CAMORIANO R, ROSASCO L. Less is more: Nyström computational regularization[C]// Proceedings of the 28th Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2015: 1657-1665.

[8] RAHIMI A, RECHT B. Random features for large-scale kernel machines[C]// Proceedings of the 20th Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2007: 1177-1184.

[9] YAO Y, ROSASCO L, CAPONNETTO A. On early stopping in gradient descent learning[J]. Constructive Approximation, 2007, 26(2): 289-315.

[10] BLANCHARD G, KRÄMER N. Optimal learning rates for kernel conjugate gradient regression[C]// Proceedings of the 20th Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2010: 226-234.

[11] ZHANG Y, DUCHI J, WAINWRIGHT M. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates[J]. Journal of Machine Learning Research, 2015, 16: 3299-3340.

[12] GU Q, HAN J. Clustered support vector machines[C]// Proceedings of the 16th International Conference on Artificial Intelligence and Statistics. Scottsdale: PMLR, 2013, 31: 307-315.

[13] HSIEH C J, SI S, DHILLON I S. A divide-and-conquer solver for kernel support vector machines[C]// Proceedings of the 31th International Conference on Machine Learning. Beijing: PMLR, 2014, 32(1): 566-574.

[14] TANDON R, SI S, RAVIKUMAR P, et al. Kernel ridge regression via partitioning[J/OL]. (2016.8.5)[2017.5.24]. https://arxix.org/pdf/1608.01976.pdf

[15] GITTENS A, MAHONEY M W. Revisiting the Nyström method for improved large scale machine learning[J]. Journal of Machine Learning Research, 2016, 17(1): 3977-4041.

[16] HUANG P S, AVRON H, SAINATH T N, et al. Kernel methods match deep neural networks on TIMIT[C]// Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE press, 2014: 205-209.

[17] BOTTOU L, VAPNIK V. Local learning algorithms[J]. Neural computation, 1992, 4(6): 888-900.

[18] ZHANG Y, DUCHI J C, WAINWRIGHT M J. Communication-efficient algorithms for statistical optimization[J]. Journal of Machine Learning Research, 2012, 14(1): 3321-3363.

[19] MACKEY L, TALWALKAR A, JORDAN M I. Divide-and-Conquer matrix factorization[C]// Proceedings of the 25th Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2012: 1134-1142.

[20] PAN Y, XIA R, YIN J, et al. A divide-and-conquer method for scalable robust multitask learning[J]. IEEE transactions on neural networks and learning systems, 2015, 26(12): 3163-3175.