文章编号:1672-3961(2024)04-0013-08 DOI:10.6040/j.issn.1672-3961.0.2023.163

基于神经正切核草图的多核学习方法

王梅[1],许传海[2*],王伟东[1],韩非[3]
(1.东北石油大学计算机与信息技术学院, 黑龙江 大庆 163318; 2.新疆理工学院信息工程学院, 新疆 阿克苏 843100; 3.东北石油大学人工智能能源研究院, 黑龙江 大庆 163318)

摘要:为提高多核学习对大规模及分布不均衡问题的处理能力,提出一种基于神经正切核草图的多核学习方法(neural tangent kernel sketch multiple kernel learning, NS-MKL)。应用神经正切核代替单层核函数作为多核学习基核函数,提高多核学习方法表示能力;使用神经正切核草图算法对神经正切核进行近似,减少神经正切核的特征数量和特征维度,提高多核学习方法计算效率;使用核目标对齐计算每个近似神经正切核的基核权重,根据权重进行多核线性组合,得到多核决策函数。在 3 个 UCI 数据集上对神经正线核(neural tangent kernel, NTK)核支持向量机(support vector machine, SVM)与传统核 SVM 进行比较分析,NTK 核 SVM 比传统核 SVM 预测准确率最低提高 1.9%,精度最低提高 2.0%,召回率最低提高 2.0%。在 3 个 UCI 数据集上对 NS-MKL 与传统核 MKL 进行比较分析,NS-MKL 比应用传统核 MKL 预测准确率最低提高 2.0%,运行时间最低减少 9 s。NS-MKL 能提高预测准确率,降低计算速度。

关键词:多核学习;神经正切核;核目标对齐;反余弦核;草图算法
中图分类号:TP391 **文献标志码**:A
引用格式:王梅,许传海,王伟东,等.基于神经正切核草图的多核学习方法[J].山东大学学报(工学版),2024,54(4):13-20.
WANG Mei, XU Chuanhai, WANG Weidong, et al. Multi-kernel learning method based on neural tangent kernel sketch[J]. Journal of Shandong University (Engineering Science), 2024, 54(4):13-20.

Multi-kernel learning method based on neural tangent kernel sketch

WANG Mei[1], XU Chuanhai[2*], WANG Weidong[1], HAN Fei[3]
(1. School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, Heilongjiang, China; 2. College of Information Engineering, Xinjiang Institute of Technology, Aksu 843100, Xinjiang, China; 3. Artificial Intelligence Energy Research Institute, Northeast Petroleum University, Daqing 163318, Heilongjiang, China)

Abstract: To address large-scale and unbalanced distribution problems, a neural tangent kernel sketch-based multiple kernel learning method (NS-MKL) was proposed. The neural tangent kernel was applied instead of the single-layer kernel function as the base kernel function of the multiple kernel learning to enhance the representation capability of the multiple kernel learning method. The neural tangent kernel was approximated using the neural tangent kernel sketch algorithm, which reduced the number of features and the feature dimension of the neural tangent kernel, thus improved the computational efficiency of the multiple kernel learning method. The kernel target alignment was applied to compute the base kernel weight for each approximate neural tangent kernel, and a linear combination of multiple kernels was performed based on the weight to obtain the final multiple kernel decision function. By compareing the single-layer kernel with the NTK-based SVM in 3 UCI datasets, the NTK-based SVM improved the accuracy at least 1.9%, precision at least 2.0%, and recall rate at least 2.0%. By compareing the NS-MKL with other MKL methods in 3 UCI datasets, the NS-MKL improved the accuracy at least 2.0%, and runtime reduced at least 9 s. The proposed algorithm had higher predictive accuracy and faster computation speed.

Keywords: multi-kernel learning; neural tangent kernel; kernel-target alignment; arc-cosine kernel; sketch algorithm

收稿日期:2023-07-06
基金项目:国家自然科学基金资助项目(51774090,62073070);黑龙江省博士后科研启动金资助项目(LBH-Q20080);黑龙江省研究生精品课程建设项目(15141220103)
第一作者简介:王梅(1975—),女,河北安国人,教授,硕士生导师,博士,主要研究方向为机器学习、核方法、模型选择等。E-mail: wangmei@nepu.edu.cn
*通信作者简介:许传海(1998—),男,黑龙江鸡西人,助教,硕士,主要研究方向为机器学习、深度核学习。E-mail: 2023266@xjit.edu.cn

0 引言

核方法，如支持向量机（support vector machine，SVM）是一类重要机器学习方法[1]，成功应用于各种机器学习问题中。这些方法通过一个核函数隐式将数据点从输入空间映射到特征空间，在特征空间中学习线性学习器。常用的核函数包括高斯核、径向基核、多项式核等。核方法性能与核函数关系较大，核函数选择成为核方法研究关键问题。用户需先确定核函数类型，再通过优化核函数质量函数来确定核参数。常用的质量函数包括交叉验证、泛化误差界和核对齐等[2-4]。

神经网络与核方法的关联研究工作开始于十几年前。文献[5]指出单隐层无限宽神经网络与高斯过程是等价的；文献[6]进一步把结果推广至深度完全连接神经网络，它仅训练最后一层，剩余层均为初值；文献[7]提出过参数化深层神经网络训练可由神经正线核（neural tangent kernel，NTK）的核回归训练动力学表征。NTK 在无限宽条件下趋于确定的核，在梯度下降训练过程中基本不变；文献[8]通过试验验证，原始有限宽网络的预测与线性化版本预测有较好一致性。文献[9]发现，当数据点位于超球面时，NTK 在光谱信息方面类似于 Laplace 内核；文献[10]通过分析 NTK 的特征值分布，表明 NTK 收敛于确定性分布；文献[11-12]分别将 NTK 核应用于核分类和核聚类，提高模型性能。

当样本特征包含异构信息、数据规模大、数据分布不平衡或者数据不规则时，应用单个核函数对所有样本进行映射并不可靠[13]。近年来研究人员对多核学习方法（multiple kernel learning，MKL）展开了大量的研究[14]。研究表明，多核学习使用多个基核函数凸组合实现多核模型，能够增强决策函数可解释性，获得比单核模型更优性能[15]。核选择、权重选择和模型优化求解等为多核学习研究重点问题。

对于核函数选择，文献[16]将直方图交叉核 SVM、高斯核 SVM 和多项式核 SVM 进行线性组合合成一个新的核函数；文献[17]使用 Gating 模型进行局部核函数选择；文献[18]将多个基核函数线性组合构建 MKL，处理货币特征融合问题。

对于基核函数权重计算问题，文献[19]用核目标对齐计算单核权重得到合成核；文献[20]用信息增益对数据集特征向量进行加权处理，使每个特征拥有不同的权重，提高分类准确率；文献[21]用模糊约束理论求解基核函数权重，获得组合核函数；文献[22]用解秩空间差异性方法实现核函数合并；文献[23]用加权相加与相乘方法分别对所提取基础特征进行合并，在 SVM 模型内学习每个特征代表的基本核权重。

大规模数据集上应用多核学习方法计算核矩阵所需成本较大[24]。文献[25]按各样本观测通道划分样本，不需任何插补可快速求解缺失多核学习模型；文献[26]提出一种随机方差缩减梯度方法，避免大量矩阵运算，减少内存分配；在类内散布矩阵基础上，文献[27]提出迹约束多核学习算法，能够在学习基核权重时同步调整正则化参数，加快训练速度。

近年出现许多核近似构造方法，文献[28]提出的随机特征方法是最受欢迎方法之一，可用于构造反余弦核、多项式核和一般点积核的随机特征[29]。这些低维特征能够应用于快速线性方法，节省时间和空间复杂度。

本研究综合考虑基核函数、基核权重计算以及模型快速求解，提出一种基于神经正切核草图的多核学习方法。

1 基于反余弦核随机特征和张量草图的 NTK 近似算法

1.1 符号定义

令$[n]:=\{1,2,\cdots,n\}$，$\otimes$表示张量乘积，$\odot$表示两个矩阵的元素乘数。对于方阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$，如果 $\boldsymbol{B}-\boldsymbol{A}$ 是半正定的，则有 $\boldsymbol{A}\leqslant\boldsymbol{B}$。将 $\boldsymbol{a}_{i,j}$ 定义为方阵 $\boldsymbol{A}$ 的第 i 行第 j 列元素，v_i 定义为向量 $\boldsymbol{v}$ 的第 i 项。

1.2 全连接神经网络 NTK

给定输入 $\boldsymbol{x}\in\mathbf{R}^d$，考虑一个输入维数为 d，线性整流函数（linear rectification function，ReLU）为激活函数，隐层维数为 $d_1,d_2,\cdots,d_L$ 的 L 层全连接神经网络为：

$$\begin{aligned}&\boldsymbol{h}_0=\boldsymbol{x},\\&\boldsymbol{h}_l=\sqrt{2/d_l}\,\mathrm{ReLU}(\boldsymbol{h}_{l-1}^{\mathrm{T}}\boldsymbol{W}_l),\\&f(\boldsymbol{x},\boldsymbol{\theta})=\boldsymbol{h}_L^{\mathrm{T}}\boldsymbol{w},\end{aligned}\tag{1}$$

式中，$\boldsymbol{W}_l\in\mathbf{R}^{d_{l-1}\times d_l}$，$\boldsymbol{\theta}:=(\boldsymbol{W}_1\ \boldsymbol{W}_2\cdots\ \boldsymbol{W}_L\boldsymbol{w})$表示可训练参数，$\boldsymbol{w}\in\mathbf{R}^{d_L}$，$\mathrm{ReLu}(\cdot)$为 ReLU 激活函数，$l\in\{1,2,\cdots,L\}$。神经正切核的梯度形式定义为：

$$K_{\mathrm{NTK}}^{(L)}(\boldsymbol{x},\boldsymbol{x}')=\ {}_{\theta}[\langle\frac{\partial f(\boldsymbol{x},\boldsymbol{\theta})}{\partial\boldsymbol{\theta}},\frac{\partial f(\boldsymbol{x}',\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\rangle],\tag{2}$$

式中 $\boldsymbol{\theta}$ 来自标准高斯分布。给定 n 个数据点 $\boldsymbol{X}=$

$[\boldsymbol{x}_1\ \boldsymbol{x}_2\cdots\ \boldsymbol{x}_n]^{\mathrm{T}}\in\mathbf{R}^{n\times d}$,令

$\boldsymbol{f}(\boldsymbol{X},\boldsymbol{\theta}):=[f(\boldsymbol{x}_1,\boldsymbol{\theta})\ f(\boldsymbol{x}_2,\boldsymbol{\theta})\ \cdots f(\boldsymbol{x}_n,\boldsymbol{\theta})]^{\mathrm{T}}\in\mathbf{R}^n$,则 NTK 矩阵为$\boldsymbol{K}_{\mathrm{NTK}}^{(L)}\in\mathbf{R}^{n\times n}$,其第$(i,j)$项为$\boldsymbol{K}_{\mathrm{NTK}}^{(L)}(\boldsymbol{x}_i,\boldsymbol{x}_j)$, $i,j\in[n]$。

无限宽度限制下,每个隐藏层预激活$f^{(l)}(\boldsymbol{x},\boldsymbol{\theta})$的所有坐标都趋于独立同分布,$l\in[L]$。中心高斯过程协方差$\sum^{(l-1)}:\mathbf{R}^d\times\mathbf{R}^d\to\mathbf{R}$递归定义为:

$$\sum^{(0)}(\boldsymbol{x},\boldsymbol{x}')=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}',$$

$$\boldsymbol{\Lambda}^{(l)}(\boldsymbol{x},\boldsymbol{x}')=\begin{pmatrix}\sum^{(l-1)}(\boldsymbol{x},\boldsymbol{x}) & \sum^{(l-1)}(\boldsymbol{x},\boldsymbol{x}')\\ \sum^{(l-1)}(\boldsymbol{x}',\boldsymbol{x}) & \sum^{(l-1)}(\boldsymbol{x}',\boldsymbol{x}')\end{pmatrix}\in\mathbf{R}^{2\times2},$$

$$\sum^{(l)}(\boldsymbol{x},\boldsymbol{x}')=c_\sigma\underset{(\boldsymbol{u},\boldsymbol{v})\sim N(\boldsymbol{0},\boldsymbol{\Lambda}^{(l)})}{}[\sigma(\boldsymbol{u})\sigma(\boldsymbol{v})],\quad(3)$$

式中$\boldsymbol{u},\boldsymbol{v}$为向量。

现定义导数协方差为:

$$\dot{\sum}^{(l)}(\boldsymbol{x},\boldsymbol{x}')=c_\sigma\underset{(\boldsymbol{u},\boldsymbol{v})\sim N(\boldsymbol{0},\boldsymbol{\Lambda}^{(l)})}{}[\dot\sigma(\boldsymbol{u})\dot\sigma(\boldsymbol{v})]。\quad(4)$$

全连接神经网络的 NTK 表达式为:

$$K^{(L)}(\boldsymbol{x},\boldsymbol{x}')=\sum_{l=1}^{L+1}\Big(\sum^{(l-1)}(\boldsymbol{x},\boldsymbol{x}')\cdot\prod_{l'=l}^{L+1}\dot\sum^{(l')}(\boldsymbol{x},\boldsymbol{x}')\Big),\quad(5)$$

式中$\sum^{(l-1)}(\boldsymbol{x},\boldsymbol{x}')=1$。

1.3　反余弦核随机特征

随机特征是一种缩放核的方法,可节省时间和存储空间。在大多数情况下,随机特征模型的目标是内核$K:\mathbf{R}^d\times\mathbf{R}^d\to\mathbf{R}$,对于某些分布$p$可以写为:

$$K(\boldsymbol{x},\boldsymbol{x}')=\underset{\boldsymbol{v}\sim p}{}[\Phi(\boldsymbol{x},\boldsymbol{v})\cdot\Phi(\boldsymbol{x}',\boldsymbol{v})]$$

且函数$\Phi:\mathbf{R}^d\times\mathbf{R}^d\to\mathbf{R}$。根据分布$p$采样生成$m$个向量$\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m$,随机特征近似映射定义为:

$$\boldsymbol{\Phi}_m(\boldsymbol{x}):=\frac{1}{\sqrt{m}}[\Phi(\boldsymbol{x},\boldsymbol{v}_1)\ \Phi(\boldsymbol{x},\boldsymbol{v}_2)\ \cdots\Phi(\boldsymbol{x},\boldsymbol{v}_m)]^{\mathrm{T}}\in\mathbf{R}^m,$$

近似核$K'(\boldsymbol{x},\boldsymbol{x}')=\langle\boldsymbol{\Phi}_m(\boldsymbol{x}),\boldsymbol{\Phi}_m(\boldsymbol{x}')\rangle$。与$K'$关联的内核矩阵$\boldsymbol{K}'$是一个已知分解的低秩矩阵。令

$$\boldsymbol{\Phi}:=[\boldsymbol{\Phi}_m(\boldsymbol{x}_1)\ \boldsymbol{\Phi}_m(\boldsymbol{x}_2)\ \cdots\boldsymbol{\Phi}_m(\boldsymbol{x}_n)]^{\mathrm{T}}\in\mathbf{R}^{n\times m},$$

则$\boldsymbol{K}'=\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\approx\boldsymbol{K}$。近似核矩阵的秩为$m$,参数$m$在计算复杂度和近似质量之间进行权衡,较小的$m$会加快计算速度,降低内核近似的准确性。

文献[30]提出反余弦核A_0的随机特征为:

$$a_0(\boldsymbol{x})=\sqrt{2/m_0}\,\mathrm{Step}([\boldsymbol{w}_1\ \boldsymbol{w}_2\cdots\ \boldsymbol{w}_{m_0}]^{\mathrm{T}}\boldsymbol{x})。\quad(6)$$

A_1的随机特征为:

$$a_1(\boldsymbol{x})=\sqrt{2/m_1}\,\mathrm{ReLU}([\boldsymbol{w}_1'\ \boldsymbol{w}_2'\cdots\ \boldsymbol{w}_{m_1}']^{\mathrm{T}}\boldsymbol{x}),\quad(7)$$

式中,$\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_{m_0}$, $\boldsymbol{w}_1',\boldsymbol{w}_2',\cdots,\boldsymbol{w}_{m_1}'\in\mathbf{R}^d$采样于$N(0,I_d)$,step$(\cdot)$为阶跃函数,$l\in\{1,2,\cdots,L\}$。对于$\boldsymbol{x},\boldsymbol{x}'\in\mathbf{R}^d$,

$$[\langle a_0(\boldsymbol{x}),a_0(\boldsymbol{x}')\rangle]=A_0(\boldsymbol{x},\boldsymbol{x}'),$$

$$[\langle a_1(\boldsymbol{x}),a_1(\boldsymbol{x}')\rangle]=A_1(\boldsymbol{x},\boldsymbol{x}'),$$

其反余弦核A_0和A_1为:

$$A_0(\boldsymbol{x},\boldsymbol{x}'):=1-\frac{1}{\pi}\cos^{-1}\Big(\frac{\langle\boldsymbol{x},\boldsymbol{x}'\rangle}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}'\|_2}\Big),$$

$$A_1(\boldsymbol{x},\boldsymbol{x}'):=\|\boldsymbol{x}\|_2\|\boldsymbol{x}'\|_2 f\Big(\frac{\langle\boldsymbol{x},\boldsymbol{x}'\rangle}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}'\|_2}\Big)。$$

文献[31]利用反余弦核显式特征映射的递归张量,提出了全连接神网络 *NTK* 的显示无限维特征映射。将每个显示特征映射替换为相应内核的随机特征映射,可得到 *NTK* 的随机特征映射,具体结构为:

$$\boldsymbol{\Psi}^{(l+1)}(\boldsymbol{x})=a_1(\boldsymbol{\Psi}^{(l)}(\boldsymbol{x})),$$

$$\boldsymbol{\Phi}^{(0)}(\boldsymbol{x})=\boldsymbol{\Psi}^{(0)}(\boldsymbol{x})=\boldsymbol{x},$$

$$\boldsymbol{\Phi}^{(l+1)}(\boldsymbol{x})=[\boldsymbol{\Psi}^{(l+1)}(\boldsymbol{x}),a_0(\boldsymbol{\Psi}^{(l)}(\boldsymbol{x}))\otimes\boldsymbol{\Phi}^{(l)}(\boldsymbol{x})],\quad(8)$$

式中$l=0,1,\cdots,L-1$。这些特征可用于 NTK 近似,形式为:

$$K_{\mathrm{NTK}}^{(l)}(\boldsymbol{x},\boldsymbol{x}')\approx\langle\boldsymbol{\Phi}^{(l)}(\boldsymbol{x}),\boldsymbol{\Phi}^{(l)}(\boldsymbol{x}')\rangle。\quad(9)$$

由式(8)可以看出,NTK 核的特征数量在深度L上为指数级,输出特征的$\boldsymbol{\Phi}^{(L)}(\boldsymbol{x})$的维度为:

$$(\sum_{k=0}^{L-1}m_0^k)m_1+m_0^Ld=O(m_0^L(m_1+d))。$$

由此可得$O(m_0^L(m_1+d))$时间复杂度。深度L的指数增长来源于张量积⊗。当L足够大时,特征数极易大于数据数。为减少特征映射维度,文献[32]在 NTK 反余弦核随机特征基础上加入张量草图算法,提出神经正切核草图 NTK 近似算法 NTKSketch。

1.4　TensorSketch 变换

TensorSketch 基于 CountSketch 变换。CountSketch 变换是一种保留范数的降维技术。令$h:[d]\to[m]$是一个成对独立的哈希函数,其箱子是随机均匀选择;$s:[d]\to\{+1,-1\}$是一个成对独立的符号函数,其中符号是均匀选择。给定$\boldsymbol{x}\in\mathbf{R}^d$和$m\in\mathbf{N}$,定义$\boldsymbol{C}:\mathbf{R}^d\to\mathbf{R}^m$, $i\in[m]$则有

$$[\boldsymbol{C}(\boldsymbol{x})]_i=\sum_{j:h(j)=i}s(j)[\boldsymbol{x}]_j,\quad(10)$$

且$\mathrm{E}[\langle\boldsymbol{C}(\boldsymbol{x}),\boldsymbol{C}(\boldsymbol{y})\rangle]=\langle\boldsymbol{x},\boldsymbol{y}\rangle$。

文献[33]提出将 CountSketch 应用于向量张量

积算法 TensorSketch。令 $h_1:[d_1]\to[m]$ 和 $h_2:[d_2]\to[m]$ 是成对独立的随机哈希函数；$s_1:[d_1]\to\{-1,1\}$ 和 $s_2:[d_2]\to\{-1,1\}$ 是成对独立符号函数。分别用 $\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$ 表示对应的 CountSketch。考虑一个新的变换 $\boldsymbol{C}:\mathrm{R}^{d_1,d_2}\to\mathbf{R}^m$，其哈希函数定义为：

$$H(j_1,j_2)\equiv h_1(j_1)+h_2(j_2)\ (\mathrm{mod}\ m),$$

符号函数定义为：

$$S(j_1,j_2)=s_1(j_1)\cdot s_2(j_2),\ j_1\in[d_1],\ j_2\in[d_2]。$$

给定 $\boldsymbol{x}\in\mathbf{R}^{d_1}$，$\boldsymbol{y}\in\mathbf{R}^{d_2}$，$\boldsymbol{C}(\boldsymbol{x}\otimes\boldsymbol{y})$ 等于 $\boldsymbol{C}_1(\boldsymbol{x})$ 和 $\boldsymbol{C}_2(\boldsymbol{y})$ 的卷积，具体表示为：

$$\boldsymbol{C}(\boldsymbol{x}\otimes\boldsymbol{y})=F^{-1}(F(\boldsymbol{C}_1(\boldsymbol{x}))\odot F(\boldsymbol{C}_2(\boldsymbol{y}))),\tag{11}$$

式中 F 和 F^{-1} 为快速傅里叶变换及其逆变换。

1.5 NTKSketch 算法

NTKSketch 算法使用 TensorSketch 减少 $a_0(\boldsymbol{\Psi}^{(l)}(\boldsymbol{x}))\otimes\boldsymbol{\Phi}^{(l)}(\boldsymbol{x})$ 维数，将其替换为

$$\boldsymbol{\Gamma}^{(l)}(\boldsymbol{x}):=F^{-1}(F(\boldsymbol{C}_1(a_0(\boldsymbol{\Psi}^{(l)}(\boldsymbol{x}))))\odot F(\boldsymbol{C}_2(\boldsymbol{\Phi}^{(l)}(\boldsymbol{x})))),\tag{12}$$

式中，$\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$ 是映射到 $\mathbf{R}^{m_{cs}}$ 的 CountSketch 转换，对全连接神经网络每一层重复该过程。基于 NTKSketch 的特征构造如算法 1 所示。

算法 1 NTKSketch 算法[32]。

输入 $\boldsymbol{x}\in\mathbf{R}^d$，网络深度 L，特征维度 m_0、m_1 和 m_{cs}。

初始化 $\boldsymbol{\Phi}^{(0)}(\boldsymbol{x})=\boldsymbol{x}$，$\boldsymbol{\Psi}^{(0)}(\boldsymbol{x})=\boldsymbol{x}$，$m=d$，$l=1$；

循环

依据 $N(0,\boldsymbol{I}_m)$ 对 $\boldsymbol{w}_i$ 进行采样，$i\in[m_0]$，根据式(6)计算反余弦核随机特征

$$\boldsymbol{\Lambda}^{(l)}(\boldsymbol{x})\leftarrow\sqrt{2/m_0}\,\mathrm{Step}([\boldsymbol{w}_1\ \boldsymbol{w}_2\cdots\boldsymbol{w}_{m_0}]^{\mathrm{T}}\boldsymbol{\Psi}^{(l-1)}(\boldsymbol{x}));$$

依据 $N(0,\boldsymbol{I}_m)$ 对 $\boldsymbol{w}'_j$ 进行采样，$j\in[m_1]$，根据式(7)计算反余弦核随机特征

$$\boldsymbol{\Psi}^{(l)}(\boldsymbol{x})\leftarrow\sqrt{2/m_1}\,\mathrm{ReLU}([\boldsymbol{w}'_1\ \boldsymbol{w}'_2\cdots\boldsymbol{w}'_{m_1}]^{\mathrm{T}}\boldsymbol{\Psi}^{(l-1)}(\boldsymbol{x}));$$

设置两个独立的 CountSketch 变换 $C_0^{(l)}$ 和 $C_1^{(l)}$，根据式(12)计算 TensorSketch 变换

$$\boldsymbol{\Gamma}^{(l)}(\boldsymbol{x})\leftarrow F^{-1}(F(\boldsymbol{C}_0^{(l)}(\boldsymbol{\Lambda}^{(l)}(\boldsymbol{x})))\odot F(\boldsymbol{C}_1^{(l)}(\boldsymbol{\Phi}^{(l-1)}(\boldsymbol{x}))));$$

$$\boldsymbol{\Phi}^{(l)}(\boldsymbol{x})\leftarrow[\boldsymbol{\Psi}^{(l)}(\boldsymbol{x}),\boldsymbol{\Gamma}^{(l)}(\boldsymbol{x})];$$

$$l=l+1;$$

直到 $l\geqslant L+1$，算法收敛跳出循环；

输出 NTK 随机特征 $\boldsymbol{\Phi}^{(L)}(\boldsymbol{x})$。

2 基于 NTKSketch 的多核学习算法

2.1 多核学习方法

令训练数据集 $D=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^n$，样本规模为 n，其中 $\boldsymbol{x}_i\in\boldsymbol{X}$ 为特征向量，$y_i\in\{-1,+1\}$ 为标签。核方法中，输入空间为 $\boldsymbol{X}$，特征空间为 $\boldsymbol{H}$，若存在一个从 $\boldsymbol{X}$ 到 $\boldsymbol{H}$ 的特征映射函数 ϕ，对于任意 $\boldsymbol{x}_i,\boldsymbol{x}_j\in\boldsymbol{X}$，核函数 $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 定义为：

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)。\tag{13}$$

SVM 优化问题可表示为：

$$\begin{aligned}&\min_{s,b,\xi_i}\quad\|\boldsymbol{s}\|^2/2+C\sum_{i=1}^n\xi_i\\&\text{s.t.}\quad y_i(\boldsymbol{s}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b)\geqslant1-\xi_i\\&\qquad\xi_i\geqslant0,\quad i=1,\cdots,n\end{aligned}\tag{14}$$

式中，ξ_i 是松弛变量，C 是惩罚系数，$\boldsymbol{s}$ 为超平面的法向量，b 为截距。应用 KKT 条件及拉格朗日对偶性，式(14)的对偶问题可以表示为：

$$\begin{aligned}&\max_{\alpha}\sum_{i=1}^n\alpha_i-\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\alpha_i\alpha_jy_iy_jk(\boldsymbol{x}_i,\boldsymbol{x}_j),\\&\text{s.t.}\sum_{i=1}^n a_iy_i=0,\\&\qquad0\leqslant a_i\leqslant C,\quad i=1,\cdots,n,\end{aligned}\tag{15}$$

式中 α_i 为拉格朗日乘子。对式(15)求解后得到 SVM 决策函数为：

$$f(\boldsymbol{x})=\mathrm{sgn}(\sum_{i=1}^n\alpha_iy_ik(\boldsymbol{x}_i,\boldsymbol{x})+b)。\tag{16}$$

多核 SVM 是典型多核学习方法，其将若干个基核函数进行凸组合构造组合核函数：

$$\begin{aligned}&k(\cdot,\cdot)=\sum_{i=1}^m d_ik_i(\cdot,\cdot),\\&\text{s.t.}\quad d_i\geqslant0,\ \sum_{i=1}^m d_i=1,\end{aligned}\tag{17}$$

式中，d_i 为基核函数权重系数，m 为基核函数个数，$k_i(\cdot,\cdot)$ 为基核函数。由式(16)，多核 SVM 决策函数为：

$$f(\boldsymbol{x})=\mathrm{sgn}(\sum_{j=1}^n a_jy_j\sum_{i=1}^m d_ik_i(\boldsymbol{x}_i,\boldsymbol{x})+b)。\tag{18}$$

2.2 核目标对齐

文献[3]提出核目标对齐度量两个核函数相似性程度。核函数 k 的核矩阵 $\boldsymbol{K}$ 定义为 $[\boldsymbol{K}]_{i,j}=k(\boldsymbol{x}_i,\boldsymbol{x}_j)$，核函数 k_1 和核函数 k_2 的经验核目标对齐被定义为：

$$A(\boldsymbol{K}_1,\boldsymbol{K}_2)=\frac{\langle\boldsymbol{K}_1,\boldsymbol{K}_2\rangle_F}{\sqrt{\langle\boldsymbol{K}_1,\boldsymbol{K}_1\rangle_F\langle\boldsymbol{K}_2,\boldsymbol{K}_2\rangle_F}}。\tag{19}$$

为便于分类,定义一个理想核矩阵 $\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$,其中 $\boldsymbol{y}=(y_1\ y_2\cdots y_n)^{\mathrm{T}}$ 是数据集 D 的标签向量。核矩阵 $\boldsymbol{K}$ 和理想核矩阵 $\boldsymbol{K}^*$ 之间的经验核目标对齐为:

$$\hat{A}(\boldsymbol{K},\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})=\frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{y}}{n\sqrt{\langle \boldsymbol{K},\boldsymbol{K}\rangle_{\mathrm{F}}}}。\tag{20}$$

$\hat{A}$ 可表示数据集样本与标签分布在特征空间中的一致程度。当 $\hat{A}$ 越大,数据分布与标签分布在特征空间中一致性越高,则该核函数对原始数据的特征选择与表达能力更好。可通过计算不同基核函数的 $\hat{A}$ 得到不同核函数的权重。

2.3　基于 NTKSketch 的多核学习算法

NTK 相较于传统核函数有更深结构,有更好表示能力和性能;应用 NTKSketch 算法实现 NTK 近似,加快其运行效率。现将近似 NTK 作为多核学习基核函数,式(17)改写为:

$$\hat{K}_{\mathrm{NTK}}=\sum_{i=1}^{m}\tilde{a}_i\hat{K}^i_{\mathrm{NTK}},\tag{21}$$

式中 $\tilde{a}_i=\hat{A}(\hat{K}_i)/\sum_{j=1}^{m}\hat{A}(\hat{K}_j)$。由式(18)和式(21),多核 SVM 决策函数可表示为:

$$f(\boldsymbol{x})=\mathrm{sgn}\Big(\sum_{j=1}^{n}a_jy_j\sum_{i=1}^{m}\tilde{a}_i\hat{K}^i_{\mathrm{NTK}}(\boldsymbol{x}_i,\boldsymbol{x})+b\Big)。\tag{22}$$

将 NTKSketch 算法引入到多核学习方法中,实现一种多核学习算法(NTKSketch-Multiple kernel learning, NS-MKL)。将原始结构简单基核函数替换为深层结构 NTK 核函数,使用 NTKSketch 算法对 NTK 近似;应用近似 NTK 核目标对齐值得到每个近似 NTK 基核权重;根据权重进行加权组合得到多核学习决策函数。NS-MKL 算法流程如算法 2 所示。

算法 2　NS-MKL 算法

输入　训练数据 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\}$,基核函数个数 m,特征维度 m_0、m_1 和 m_{cs}。

$i=1$;

循环

随机初始化层数 L;

根据层数 L 和特征维度 m_0、m_1 和 m_{cs},按照算法 1 的流程计算近似神经正切核

$\hat{K}_{\mathrm{NTK}}(\boldsymbol{x},\boldsymbol{x}')\approx\langle\boldsymbol{\Phi}^{(L)}(\boldsymbol{x}),\boldsymbol{\Phi}^{(L)}(\boldsymbol{x}')\rangle$;

根据式(20)计算第 i 个近似 NTK 的核目标对齐值

$$\hat{A}(\boldsymbol{K}^i_{\mathrm{NTK}},\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})=\frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{K}^i_{\mathrm{NTK}}\boldsymbol{y}}{n\sqrt{\langle\boldsymbol{K}^i_{\mathrm{NTK}},\boldsymbol{K}^i_{\mathrm{NTK}}\rangle_F}};$$

$i=i+1$;

直到 $i\geqslant m+1$,算法收敛跳出循环;

根据式(21)求出最终的线性加权组合的多核函数 $\hat{K}_{\mathrm{NTK}}$;

将 $\hat{K}_{\mathrm{NTK}}$ 带入多核 SVM 中求解优化问题,求出决策函数 $f(\boldsymbol{x})$;

输出　决策函数 $f(\boldsymbol{x})$。

3　试验结果与分析

本章共进行两组试验对提出的模型性能进行验证,一个是 NTK 与传统核函数的性能对比试验,另一个是所提 NS-MKL 算法与其他常用多核学习算法的性能对比试验。

3.1　NTK 核性能试验及结果

本组试验应用 Abalone 数据集、Car 数据集和 Avila 数据集 3 个标准数据集。表 1 为 3 个数据集的基本信息。所有试验均随机选取 60%数据作为训练数据,40%作为测试数据。

表 1　NTK 性能验证数据集

Table 1　Datasets for NTK Performance testing

数据集	样本数	类别数	维数
Car	1 728	4	6
Abalone	4 177	3	8
Avila	10 430	3	10

由表 1 看出,Car 数据集、Abalone 数据集和 Avila 数据集样本规模不同,可对比 NTK 与传统核函数在小样本集、中样本集和较大样本集上性能表现。

利用 Scikit-Ntk 软件包随机初始化 ntk_1、ntk_2 和 ntk_3 3 个神经正切核函数,初始化层数 L 为 $\{1,2,3,4,5\}$,缩放因子 c_σ 为 $\{10^{-5},10^{-4},\cdots,10^{3}\}$。将 3 个神经正切核函数与高斯核函数、线性核函数和多项式核函数应用于 SVM 模型。线性核函数参数为 $C=1.0$;高斯核函数参数为 $C=1.0,\gamma=0.1$;多项式核函数参数为 $C=1.0,d=3$。

采用准确率 A_{cc}、精确率 P_{re} 和召回率 R_{ec} 3 个常用评价标准对各分类器在多个数据集上进行性能评估,其中精确率和召回率分别为宏精确率和宏召回率。

各模型在 3 个数据集上的准确率、精确率和召回率统计结果分别如表 2 所示。由表 2 试验结果可得出,NTK 核函数相比于传统核函数在小型、中型和较大规模数据集上都有更高准确率;传统核函数准确率虽然与神经正切核相差不大,精确率和召回率较低。

表2 NTK核与其他核的性能对比

Table 2 The comparison between NTK and other kernels

核	Car			Abalone			Avila		
	A_{cc}	P_{re}	R_{ec}	A_{cc}	P_{re}	R_{ec}	A_{cc}	P_{re}	R_{ec}
linear	0.609	0.180	0.250	0.644	0.630	0.570	0.616	0.660	0.600
rbf	0.709	0.360	0.380	0.498	0.250	0.330	0.744	0.760	0.740
poly	0.751	0.550	0.510	0.648	0.600	0.600	0.742	0.750	0.740
ntk_1	**0.887**	0.640	0.620	**0.667**	0.630	0.600	**0.805**	**0.820**	**0.800**
ntk_2	0.861	**0.750**	**0.700**	0.648	**0.650**	**0.620**	0.797	0.790	0.792
ntk_3	0.873	0.740	0.670	0.654	0.630	0.590	0.801	0.800	0.794

注:黑体数字为最佳性能指标值。

为验证神经正切核与传统核函数在精确率和召回率上的差异,将 ntk_1 与其他3个传统核在Car数据集上的每一类的精确率和召回率进行对比,具体结果如表3所示。

表3 Car数据集上各类别预测试验结果

Table 3 Experimental results for each category on the car dataset

类别	样本个数	ntk_1		linear		rbf		poly	
		P_{re}	R_{ec}	P_{re}	R_{ec}	P_{re}	R_{ec}	P_{re}	R_{ec}
−1	499	0.950	0.950	0.720	1.000	0.900	0.950	0.920	0.950
0	144	0.770	0.830	0.000	0.000	0.550	0.630	0.470	0.520
1	20	0.480	0.550	0.000	0.000	0.000	0.000	0.170	0.350
2	29	1.000	0.520	0.000	0.000	0.000	0.000	0.200	0.240

由表3可以看出,4个类别上不同核的精确率和召回率差别较大,传统核函数在进行多分类且类别分布不均衡时只能识别出其中一类或两类,另外几类无法识别。由此也可验证神经正切核在处理类别不均衡数据集时优于传统核函数。

3.2 NS-MKL算法性能试验及结果

本节对比NS-MKL算法与AverageMKL算法,EasyMKL算法和比例加权多核学习方法(proportionally weighted multiple kernels learning,PWMKL)[34-35]算法的性能。本组试验应用Bean数据集、Occupancy数据集和Bank数据集3个UCI数据集。表4为3个数据集的基本信息。

表4 NS-MKL性能测试数据集

Table 4 Datasets for NS-MKL Performance testing

数据集	样本数	类别数	维数
Bean	13 611	7	16
Occupancy	20 560	2	10
Bank	41 118	2	20

AverageMKL算法求解一个简单的线性核组合,将组合定义为基本内核的平均值,内核组合定义为:

$$k(\boldsymbol{x},\boldsymbol{z})=\sum_{r=1}^{P}u_r k_r(\boldsymbol{x},\boldsymbol{z}),\ u_r=\frac{1}{P},$$

式中μ为组合系数。

EasyMKL算法求解得到使类之间的边距最大化的内核组合,定义为:

$$k(\boldsymbol{x},\boldsymbol{z})=\sum_{r=1}^{P}u_r k_r(\boldsymbol{x},\boldsymbol{z}),\ u_r\geqslant 0\wedge\|\boldsymbol{u}\|_1=1。$$

PWMKL算法是一种启发式MKL算法,根据各个内核的性能分配权重,定义为:

$$k(\boldsymbol{x},\boldsymbol{z})=\sum_{r=1}^{P}u_r k_r(\boldsymbol{x},\boldsymbol{z}),$$

式中,m是达到的最小精度,δ是超参数。

采用与3.1节相同的方法随机初始化ntk_1和ntk_2两个NTK核函数;使用NTK草图生成两个与ntk_1和ntk_2结构相似的ntk_3和ntk_4两个近似NTK核函数;将两个原始神经正切核函数与两个近似神经正切核核函数分别应用于2.3节所描述的NS-MKL算法,在Occupancy、Bean和Bank数据集上与使用传统核函数作为基核的AverageMKL算法、EasyMKL算法和PWMKL算法进行对比试验。

对于Occupancy和Bank数据集,ntk_1和ntk_2随机初始化层数L分别为1和2,缩放因子c_σ分别为10^{-5}和100;对于bean数据集,ntk_1和ntk_2随机初始化层数L分别为1和3,缩放因子c_σ均为10^{-5};线性核函数参数为$C=1.0$,高斯核函数参数为$C=1.0$,$\gamma=0.1$。

AverageMKL算法、EasyMKL算法和PWMKL算法应用MKLpy包实现。本组试验采用准确率A_{cc}和算法运行挂钟时间t作为多核学习算法评价准则。不同多核学习算法在数据集上的准确率和运

行时间统计结果如表 5 所示。表 5 中 AverageMKL(1+r),EasyMKL(1+r)和 PWMKL(1+r)表示使用线性核和高斯核作为基核函数;NS-MKL (ntk_1+ntk_2)表示使用两个原始的 NTK 核函数作为基核函数;NS-MKL(ntk_3+ntk_4)表示使用 2 个近似 NTK 核函数作为基核函数。

表 5　NS-MKL 与其他多核学习算法的性能对比

Table 5　The comparison between NS-MKL and other multi-core learning algorithms

核	Occupancy		Bean		Bank	
	A_{cc}	t/s	A_{cc}	t/s	A_{cc}	t /s
AverageMKL(1+r)	0.976	172	0.891	55	0.896	378
EasyMKL(1+r)	0.982	294	0.897	104	0.902	726
PWMKL(1+r)	0.987	535	0.902	263	0.913	1 351
NS-MKL (ntk1+ntk2)	**0.992**	337	**0.915**	128	**0.925**	860
NS-MKL (ntk3+ntk4)	0.989	**150**	0.907	**46**	0.915	**352**

注:黑体数字为最佳性能指标值。

由表 5 可看出:在 Occupancy、Bean 和 Bank 数据集上,应用原始 NTK 的 NS-MKL 算法准确率均比其他多核学习算法要高,使用近似 NTK 的 NS-MKL 算法运行时间均比另外几个多核学习算法要快;在 Occupancy 数据集上近似 NTK 的 NS-MKL 算法相较于使用原始 NTK 的 NS-MKL 算法速度提高2 倍多,准确率仅低 0.3%;在 Bean 数据集上相较于使用原始 NTK 的 NS-MKL 算法速度提高 2.5 倍多,准确率仅低 0.8%;在 Bank 数据集中相较于使用原始 NTK 的 NS-MKL 算法速度提高2.4倍多,准确率仅低1.0%。

综上,NS-MKL 算法在 3 个数据集上的挂钟时间均比其他几个算法要短,这也就验证了将 NTK 草图算法引入多核学习方法可有效提高在大规模数据集上的计算效率。通过对比使用原始 NTK 和近似 NTK 的 NS-MKL 算法在不同数据集上的准确率可以发现,使用近似 NTK 相较于原始 NTK 仅有少量性能损失。NS-MKL 算法在 3 个数据集上的准确率均比另外 3 个多核学习算法要高,同时挂钟时间更短,验证了 NS-MKL 算法相较于另外 3 个多核学习算法有较快计算速度和较高预测精度。

4　结论

针对现有多核学习方法存在的基核函数表示能力不足和在大规模数据中计算缓慢等问题,基于 NTKSketch 算法提出了一种名为 NS-MKL 的多核学习方法。该方法使用 NTK 核代替传统核函数作为多核学习基核,提高多核学习方法表示能力;使用 NTKSketch 算法计算 NTK 的随机特征,减少 NTK 在计算时产生的特征数和特征的维度,加快多核学习方法计算效率;根据核目标对齐值将得到的近似 NTK 进行线性凸组合,基于 SVM 建立多核学习分类器模型 NS-MKL。通过在多个 UCI 标准数据集上进行对比试验,验证了所提算法可有效提升多核学习方法运算效率和分类准确率。

由于核矩阵在进行大规模计算时不仅慢还会占用较大内存,如何减少其内存占用将是下一步需要解决问题。

参考文献:

[1] ZHANG T. An introduction to support vector machines and other kernel-based learning methods [J]. AI Magazine, 2001, 22(2): 103-104.

[2] SOLICH P. Bayesian methods for support vector machines: Evidence and predictive class probabilities [J]. Machine Learning, 2002, 46(1): 21-52.

[3] KLOFT M, BLANCHARD G. On the convergence rate of ℓ p-norm multiple kernel learning [J]. Journal of Machine Learning Research, 2012(1): 2465-2502.

[4] CRISTIANINI N, SHAWE-TAYLOR J, ELISSEEFF A, et al. On kernel-target alignment[C]// Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. Vancouver, Canada: MIT Press, 2001: 367-373.

[5] WILLIAMS C. Computing with infinite networks[C]// Proceedings of the 9th International Conference on Neural Information Processing Systems. Denver, USA: MIT Press, 1996: 295-301.

[6] LEE J, BAHRI Y, NOVAK R, et al. Deep neural networks as gaussian processes[C]// Proceedings of the 6th International Conference on Learning Representations. Vancouver, Canada: MIT Press, 2018: 1-17.

[7] JACOT A, GABRIEL F, HONGLER C. Neural tangent kernel: convergence and generalization in neural networks [C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montreal, Canada: MIT Press, 2018: 8580-8589.

[8] LEE J, XIAO L, SCHOENHOLZ S, et al. Wide neural networks of any depth evolve as linear models under gradient descent [EB/OL]. (2019-08-18)[2019-12-08]. https://arxiv.org/abs/1902.06720.

[9] 王梅,许传海,刘勇. 基于神经正切核的多核学习方法[J].计算机应用,2021,41(12): 3462-3467.
WANG Mei, XU Chuanhai, LIU Yong. Multi-kernel learning method based on neural tangent kernel [J]. Journal of Computer Applications, 2021, 41(12): 3462-3467.

[10] 王梅, 宋晓晖, 刘勇, 等.神经正切核 K-Means 聚类[J].计算机应用, 2022, 42(11): 3330-3336.
WAND Mei, SONG Xiaohui, LIU Yong, et al. Neural tangent kernel K-Means clustering [J]. Journal of Computer Applications, 2022, 42(11): 3330-3336.

[11] ARORA S, DU S S, HU W, et al. On exact computation with an infinitely wide neural net [EB/OL]. (2019-04-26)[2019-11-04]. https://arxiv. org/abs/1904.11955.

[12] CHEN L, XU S. Deep neural tangent kernel and laplace kernel have the same RKHS[EB/OL]. (2020-09-22)[2021-03-18]. https://doi. org/10. 48550/ arXiv. 2009.10683.

[13] 张琳, 汪廷华, 周慧颖.基于群智能算法的 SVR 参数优化研究进展[J].计算机工程与应用, 2021, 57(16): 50-64.
ZHANG Lin, WANG Tinghua, ZHOU Huiying. Research progress on parameter optimization of SVR based on swarm intelligence algorithm [J]. Computer Engineering and Applications, 2021, 57(16): 50-64.

[14] 祁祥洲, 邢红杰. 基于中心核对齐的多核单类支持向量机[J]. 计算机应用, 2022, 42(2): 349-356.
QI Xiangzhou, XING Hongjie. Centered kernel alignment based multiple kernel one-class support vector machine[J]. Journal of Computer Applications, 2022, 42(2): 349-356.

[15] LANCKRIET G, CRISTIANINI N, BARTLETT P L, et al. Learning the kernel matrix with semidefinite programming[J]. Journal of Machine Learning Research, 2002, 5(1): 27-72.

[16] 侯能干. 基于特征融合和多核学习的行人检测方法研究[D]. 合肥:合肥工业大学, 2014.
HOU Nenggan. Research on pedestrian detection methods based on feature fusion and multi-core learning [D]. Hefei: Hefei University of Technology, 2014.

[17] GONEN M, ALPAYDIN E. Localized multiple kernel learning [C]//Proceedings of the 25th International conference on Machine learning, Helsinki, Finland: MIT Press, 2008: 352-359.

[18] 梁俊. 基于多核学习支持向量机的货币识别[D]. 长沙:中南大学, 2014.
LIANG Jun. Currency Recognition Based on Multikernel Learning Support Vector Machines [D]. Changsha: Central South University, 2014.

[19] HE Q, ZHANG Q, WANG H. Kernel-target alignment based multiple kernel one-class support vector machine [C]//Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics. Bari, Italy: IEEE, 2019: 2083-2088.

[20] 邵朝, 李强. 基于特征加权的多核支持向量机[J]. 西安邮电大学学报, 2017, 22(2): 84-88.
SHAO Chao, LI Qiang. Multi-kernel support vector machines based on feature weighting[J]. Journal of Xi′ an University of Posts and Telecommunications, 2017, 22(2): 84-88.

[21] 贾涵, 连晓峰, 潘兵. 基于模糊松弛约束的外观缺陷多核学习技术[J]. 测控技术, 2019, 38(8): 43-47.
JIA Han, LIAN Xiaofeng, PAN Bing. Appearance defects multiple kernel learning technology based on fuzzy relaxation constraints [J]. Measurement and Control Technology, 2019, 38(8): 43-47.

[22] 王梅, 薛成龙, 张强. 基于秩空间差异的多核组合方法[J]. 山东大学学报(工学版), 2021, 51(1): 108-113.
WANG Mei, XUE Chenglong, ZHANG Qiang. Multi-kernel combination method based on rank spatial difference[J]. Journal of Shandong University (Engineering Science), 2021, 51(1): 108-113.

[23] 李湘眷, 孙显, 王宏琦. 基于多核学习的高分辨率遥感图像目标检测方法[J]. 测绘科学, 2013, 38(5): 84-87.
LI Xiangjuan, SUN Xian, WANG Hongqi. Target detection method for high-resolution remote sensing images based on multi-core learning [J]. Science of Surveying and Mapping, 2013, 38(5): 84-87.

[24] NOVAK R, XIAO L, HRON J, et al. Neural tangents: fast and easy infinite neural networks in Python [EB/OL]. (2019-12-05)[2019-12-05]. https://doi.org/10.48550/arXiv.1912.02803.

[25] LIU X, LEI W, ZHU X, et al. Absent multiple kernel learning algorithms [J]. IEEE Transactions on Pattern Analysis and Machine Intelligencev, 2019, 42(6): 1303-1316.

[26] MITCHEL A P, OVENEKE M C, H SAHLI. SVRG-MKL: a fast and scalable multiple kernel learning solution for features combination in multi-class classification problems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(5): 1710-1723.

（下转第34页）