

基于 AdaBoost 的弹性网型正则化多核学习算法

任胜兵, 谢如良

(中南大学 软件学院, 长沙 410075)

摘要: 在正则化多核学习中, 稀疏的核函数权值会导致有用信息丢失和泛化性能退化, 而通过非稀疏模型选取所有核函数则会产生较多的冗余信息并对噪声敏感。针对上述问题, 基于 AdaBoost 框架提出一种弹性网型正则化多核学习算法。在迭代选取基本分类器时对核函数的权值进行弹性网型正则化约束, 即混合 L_1 范数和 L_p 范数约束, 构造基于多个基本核最优凸组合的基本分类器, 并将其集成到最终的强分类器中。实验结果表明, 该算法在保留集成算法优势的同时, 能够实现核函数权值稀疏性和非稀疏性的平衡, 与 L_1 -MKL 和 L_p -MKL 算法相比, 能够以较少的迭代次数获得分类精度较高的分类器。

关键词: 集成学习; 多核学习; 弹性网型正则化; 弱分类器; 稀疏性

中文引用格式: 任胜兵, 谢如良. 基于 AdaBoost 的弹性网型正则化多核学习算法[J]. 计算机工程, 2019, 45(10): 189-195.

英文引用格式: REN Shengbing, XIE Ruliang. Elastic-net regularization multi kernel learning algorithm based on AdaBoost[J]. Computer Engineering, 2019, 45(10): 189-195.

Elastic-net Regularization Multi Kernel Learning Algorithm Based on AdaBoost

REN Shengbing, XIE Ruliang

(School of Software, Central South University, Changsha 410075, China)

[Abstract] In regularization multi kernel learning, the sparse kernel function weight leads to the loss of useful information and the degradation of generalization performance, while selecting all kernel functions through non-sparse models generates more redundant information and is sensitivity to noise. Aiming at these problems, an elastic-net regularization multi kernel learning algorithm based on AdaBoost architecture is proposed. When the basic classifier is selected at each iteration, the weight of the kernel function is added with the elastic-net regularization, that is, mixed L_1 norm and L_p norm constraints. The basic classifier are constructed based on multi basic kernel optimal convex combinations, which are integrated into the final strong classifier. Experimental results show that the proposed algorithm can balance the sparsity and non-sparsity of the weight in kernel function while preserving the advantages of the integrated algorithm. Compared with L_1 -MKL and L_p -MKL algorithms, it can obtain the classifier with higher classification accuracy in fewer iterations.

[Key words] ensemble learning; multi kernel learning; elastic-net regularization; weak classifier; sparsity

DOI: 10.19678/j.issn.1000-3428.0050909

0 概述

目前多核学习^[1]已成为机器学习领域的热门研究方向, 各种基于支持向量机(Support Vector Machine, SVM)的多核学习方法相继出现。文献[2]在核权值 L_2 范数约束的基础上, 提出 L_p ($p > 1$) 范数约束的多核学习(L_p -norm Multiple Kernel Learning, L_p -MKL)方法。文献[3]提出基于集成策略的多核选择方法, 优化了核

函数的选择过程。在上述基于正则化的多核学习中, 当采用 L_1 范数约束核权值时, 得到的核权值是稀疏解, 会导致信息丢失和模型泛化能力减弱^[4]; 而采用 L_p ($p > 1$) 范数约束时, 得到的核权值为非稀疏解, 核函数将被全部选择, 导致模型对噪声敏感, 会出现冗余信息并且降低可解释性^[5], 同时也会增加时间和空间复杂度。文献[6]提出 MKBoost-D2 方法。该方法每次迭代得到的最优基本分类器为所有核函数

基金项目: 中南大学研究生自主探索创新项目(1053320170432)。

作者简介: 任胜兵(1969—), 男, 副教授、博士, 主研方向为模式识别、图像处理、可信软件; 谢如良, 硕士研究生。

收稿日期: 2018-03-22 **修回日期:** 2018-06-27 **E-mail:** rsb@csu.edu.cn

的加权组合,其中也包含性能较差的核函数,因此与采用 L_p 范数约束存在同样的问题。

文献[7-8]提出采用弹性网络对核函数权值进行约束的多核学习方法,文献[9]采用一种变稀疏的方式选取核权值。上述方法基于弹性正则化选择核权值,平衡了核权值的稀疏性与非稀疏性,提高了分类精度。文献[10-11]对弹性多核学习算法进行优化,文献[12]提出基于半无限规划的弹性多核学习算法。但上述弹性正则化多核学习没有对最终的分器进行加权集成,分类性能不稳定。

本文基于 AdaBoost^[13] 框架提出一种弹性网型正则化多核学习算法 (Elastic-net Regularization Multi Kernel Learning Algorithm Based on AdaBoost, AdaBoost-EMKL)。在获取基本分类器时,采用弹性网类型正则化进行约束,即对核权值同时加入 L_1 和 L_p 范数约束,以保留性能优异的核函数和大量有用的特征,通过自适应多核学习动态平衡权值的稀疏性和非稀疏性。

1 多核学习相关理论

1.1 相关符号和概念

假设数据集为 $D = \{(x_i, y_i) \mid i=1, 2, \dots, m\}$, $x_i \in \mathbb{R}^d$ 。如果学习算法求解的是二分类问题,则 $y_i = \pm 1$;如果学习算法求解的是回归问题,则 $y_i \in \mathbb{R}$ 。机器学习的目标是找到一个函数 f ,该函数能够对未知数据进行准确预测。设 f 是通过最小正则化风险得到的函数,如式(1)所示。

$$f^* = \operatorname{argmin}_f CR_{\text{emp}}(f) + \Omega(f) \quad (1)$$

其中, $R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n G(f(x_i), y_i)$ 为 f 关于损失的经验风险, $\Omega: \mathbf{H} \rightarrow \mathbb{R}$ 为正则化项,正参数 C 用于平衡正则化项和经验风险项。本文使用的正则化项为 $\Omega(f) = \frac{1}{2} \|\mathbf{w}\|_p^p$, 函数 f 采用线性模型^[14], 如式(2)所示。

$$f_{w,b}(x) = \langle \mathbf{w}, \phi(x) \rangle + b \quad (2)$$

其中, $\phi(x): \mathbf{X} \rightarrow \mathbf{H}$ 为样本从输入空间 \mathbb{R}^n 到特征空间 \mathbf{H} (希尔伯特空间) 的映射,其完成映射过程的核函数 $K(x, z) = \phi(x)\phi(z)$ 并不需要定义映射函数 ϕ 的具体形式。因此,在输入空间 \mathbb{R}^n 中的超曲面模型对应于特征空间 \mathbf{H} 中超平面模型,则原空间中的非线性可分问题转变成新空间中的线性可分问题。

1.2 正则化多核学习框架

在多核学习框架中,假设最优核 K 是基核 $\{K_1, K_2, \dots, K_M\}$ 的线性组合,其核权值为 $\{\theta_1, \theta_2, \dots, \theta_M\}$, 通常可以通过正则化风险最小来求解最优核组合,如式(3)和式(4)所示。

$$\min_{w,b,\theta \geq 0} C \sum_{i=1}^n G(f_{w,b,\theta}(x_i), y_i) + \frac{1}{2} \|\mathbf{w}\|_p^p \quad (3)$$

$$\text{s. t. } \phi(\boldsymbol{\theta}) \leq 1 \quad (4)$$

其中, $G: \mathbb{R} \times y_i \rightarrow \mathbb{R}$ 为损失函数, $\frac{1}{2} \|\mathbf{w}\|_p^p$ 为正则化项, $\phi(\boldsymbol{\theta})$ 为核权值 $\boldsymbol{\theta}$ 的正则化定义。由于式(3)为非凸函数,因此根据文献[2]进行变换可得到式(5)。

$$\min_{w,b,\theta \geq 0} C \sum_{i=1}^n G(f_{w,b,\theta}(x_i), y_i) + \frac{1}{2} \sum_{m=1}^M \frac{\|\mathbf{w}\|_p^p}{\theta_m} \quad (5)$$

如果式(5)中的损失函数 G 与式(4)都为凸函数,则式(5)也为凸函数,因此,求解的最优值为全局最优值。本文分类问题中采用的损失函数为合页损失函数 $L[y(\mathbf{w}^T \cdot \mathbf{x} + b)] = [1 - y(\mathbf{w}^T \cdot \mathbf{x} + b)]_+$, 同时在公式中引入松弛变量 ξ_i , 则式(5)可改写成式(6)。

$$\begin{aligned} \min_{w,b,\theta,\xi} C \sum_{i=1}^n \xi_i + \frac{1}{2} \sum_{m=1}^M \frac{\|\mathbf{w}\|_p^p}{\theta_m} \\ \text{s. t. } y_i (\sum_{m=1}^M \mathbf{w}_m^T \phi_m(x_i) + b) \geq 1 - \xi_i, \\ \xi_i \geq 0, \phi(\boldsymbol{\theta}) \leq 1 \end{aligned} \quad (6)$$

在多核学习过程中,如果在选择核权值时采用 L_1 范数进行约束,即在式(6)中 $\phi(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$, 则该方法被称为 L_1 -MKL。目前有大量关于 L_1 -MKL 加速的算法,例如半定规划 (Semi-Definite Programming, SDP)、半无限线性规划 (Semi-Infinite Linear Programming, SILP) 以及梯度下降等方法都可以减少求取最优核权值的时间复杂度,但是使用 L_1 -MKL 可能会丢失信息,导致模型的泛化性能下降。

如果在选择核权值时采用 L_p 范数进行约束,即在式(6)中 $\phi(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_p^p$, 则该方法被称为 L_p -MKL。在通用性上 L_p -MKL 较 L_1 -MKL 有明显提高。但是 L_p -MKL 在核权值的选择上会产生非稀疏解,即所有的核都会参与到模型中,而非稀疏模型可解释性较差,并且对噪声敏感,同时也会产生冗余信息。因此,本文在选择核权值时,采用的是混合 L_1 范数和 L_p 范数进行约束,即弹性正则化约束。基于弹性正则化约束的核权值选择是通过调节参数 ν 平衡其稀疏性与非稀疏性,该方法能够增强模型的泛化能力,同时也能降低其经验风险,即降低模型对噪声的敏感性,使模型更具鲁棒性。

2 算法设计

2.1 弹性网型正则化

针对传统多核学习中核函数权值稀疏性或非稀疏性的问题,本文在选择基本分类器时,基于 L_1 范数和 L_p 范数 ($p > 1$) 混合网型正则化对基本分类器的核函数进行约束选择,以平衡核权值的稀疏性与非稀疏性,减少因稀疏性造成模型泛化能力退化与信息丢失的情况,避免非稀疏模型的解释性差、噪声敏感以及产生冗余信息的缺点。优化目标如式(7)和式(8)所示。

$$\min_{\theta \in \theta, w, b} C \sum_{i=1}^n G(f_{w,b,\theta}(x_i), y_i) + \frac{1}{2} \sum_{m=1}^M \frac{\|\mathbf{w}\|_p^p}{\theta_m} \quad (7)$$

$$\Theta = \{\theta \in \mathbb{R}_+^m : (v\|\theta\|_1 + (1-v)\|\theta\|_p^p) \leq 1\} \quad (8)$$

其中, 参数 $v(0 \leq v \leq 1)$ 的主要作用是平衡 2 项约束, 即平衡核权值的稀疏性与非稀疏性。当 $v=0$ 时, 约束简化为对核权值的 L_p 范数约束, 即为 L_p -MKL 方法; 当 $v=1$ 时, 约束简化为 L_1 范数约束, 即为 L_1 -MKL 方法。当 $v \in (0, 1)$ 时, 式(8)关于 θ 是严格的凸函数, 并且同时具有 L_1 范数约束与 L_p 范数约束的相关特性。当 v 的值较大时, $\|\theta\|_1$ 占比更大, 因此核权值更趋向于稀疏解; 反之, 会降低核权值的稀疏性。关于优化目标式(7), 如果使用凸损失函数, 则优化目标为凸优化问题, 求得的最优值为全局最优值。与式(6)的处理方式相同, 如果采用合页损失函数, 则优化目标式(7)的原始问题等价于式(9)。

$$\min_{\theta \in \Theta, w, b} C \sum_{i=1}^n \xi_i + \frac{1}{2} \sum_{m=1}^M \frac{\|w_m\|_p^p}{\theta_m}$$

$$\text{s. t. } y_i (\sum_{m=1}^M w_m^T \varphi_m(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (9)$$

采用拉格朗日乘法, 可得到 $L(w, b, \xi, \alpha, \beta)$ 计算公式如下:

$$L(w, b, \xi, \alpha, \beta) = C \sum_{i=1}^n \xi_i + \frac{1}{2} \sum_{m=1}^M \frac{\|w_m\|_p^p}{\theta_m} - \sum_{i=1}^n \beta_i \xi_i - \sum_{i=1}^n \alpha_i (y_i (\sum_{m=1}^M w_m^T \varphi_m(x_i) + b) - 1 + \xi_i) \quad (10)$$

其中, $\alpha > 0, \beta > 0$ 为拉格朗日乘子。对拉格朗日函数对应的原始变量 w, b, ξ 求偏导数, 得到式(9)的对偶形式:

$$\min_{\theta \in \Theta} \max_{\alpha} D(\theta, \alpha), \text{ s. t. } 0 \leq \alpha \leq C, y^T \alpha = 0 \quad (11)$$

$$D(\theta, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \sum_{m=1}^M \theta_m K_m(x_i, y_j) = \sum_{i=1}^n \alpha_i - \frac{1}{2} (\alpha \circ y)^T (\sum_{m=1}^M \theta_m K_m) (\alpha \circ y) \quad (12)$$

其中, \circ 表示 2 个向量对应位置元素的乘积。式(11)为凸问题, 其最优解能够保证为全局最优。并且式(11)为 SVM 的标准对偶形式, 通常在算法中直接使用 SVM 求解器^[15]进行求解。优化式(11)的鞍点问题, 通常采用交替优化 α 和 θ 进行求解。当 θ 固定时, 可以执行 SVM 求解变量 α ; 当求取核权值 θ 时, 可以将式(11)转化成半无限规划问题再求解。假设 $u(\alpha, \theta)$ 为目标函数值, 并且 α^* 为最优解, 则对于任何 α 和 θ , 有 $u(\alpha^*, \theta) \geq u(\alpha, \theta)$ 。因此, 式(11)等价于最小化最优值的上界 η , 式(13)为半无限规划问题。

$$\min_{\eta} \eta$$

$$\text{s. t. } \forall \alpha \in A: \eta \geq \sum_{i=1}^n \alpha_i - \frac{1}{2} (\alpha \circ y)^T \cdot (\sum_{m=1}^M \theta_m K_m) (\alpha \circ y), \theta \geq 0$$

$$v\|\theta\|_1 + (1-v)\|\theta\|_p^p \leq 1 \quad (13)$$

其中, $A = \{\alpha \in \mathbb{R}^n | 0 \leq \alpha \leq C, y^T \alpha = 0\}$, 当 $p > 1$ 时,

$v\|\theta\|_1 + (1-v)\|\theta\|_p^p \leq 1$ 为非线性约束, 可以使用二阶泰勒展开式进行近似处理。在二阶泰勒展开式中, 二次项的 Hessian 矩阵为正对角阵, 因此能有效地求解二次约束问题。为保证算法的有效运行, 首先赋予 θ 一个初始值, 假设所有的核函数 K_1, K_2, \dots, K_M 都是半正定的, 则式(14)成立。

$$v\|\theta\|_1 + (1-v)\|\theta\|_p^p = 1 \quad (14)$$

假设所有 θ_i^0 都相同, 结合式(13)和式(14), 可求解初始值 θ_i^0 为:

$$\theta_i^0 = \frac{\sqrt{v^2 p^2 + 4(1-v)p} - vp}{2(1-v)p} \quad (15)$$

算法 1 为求取基本分类器的基本过程。在算法 1 中, 首先进行相关参数的初始化, 第 1 步采用 SVM 的标准解法更新 α ; 然后计算优化问题, 第 3 步 ~ 第 6 步通过混合网型约束求取核函数的权值, 当违反规范化最大约束时, 算法就会停止, 即已经达到收敛判定准则; 最后对集合分类器进行集成, 即可得到一个基本分类器。

算法 1 弹性网型正则算法

输入 核函数: $K_j(\cdot, \cdot): x \times x \rightarrow \mathbb{R}, j = 1, 2, \dots, M$
 初始化各参数: 容忍误差 $\epsilon = 5 \times 10^{-4}, t = 1, \eta^0 = -\infty, \theta_i^0$ (由式(15)计算得到)

输出 基本分类器 $f_t(x)$

1. 令 $\alpha^{\text{old}} = \alpha$, 使用 $K = \sum_{m=1}^M \theta_m K_m$ 求解 SVM, 更新 α
2. 计算 $\min_{\theta \in \Theta} \max_{\alpha} D(\theta, \alpha)$
3. while $|1 - D'/\eta'| \leq \epsilon$
4. 引入式(8)中 L_1 范数与 L_p 范数混合约束的条件, 使用二阶泰勒展开式进行近似, 求解半无限规划问题, 得到 $(\theta^{t+1}, \eta^{t+1})$
5. $t \leftarrow t + 1$
6. end while
7. 将基核分类器进行集成得到基本分类器: $f_t(x)$

2.2 AdaBoost-EMKL 算法

在采用基于弹性网型正则化选择出核函数后, 即可得到基本分类器。AdaBoost-EMKL 的总体思路是在选择出基本分类器后, 应用提升算法学习一个多核分类器。训练样本的权系数 D_i 表明其对于学习的重要性, 在每一轮的迭代中, 首先根据样本分布 D_i 抽取 n 个子样本, 得到训练子集并给定一定数量的核函数。在得到训练子集和核函数以后, 基于弹性网型正则化进行基本核函数的核权值的选取, 如算法 1 所示。选择出基本分类器 $f_i(x)$ 以后, 根据在 AdaBoost 相似过程中训练数据的分布 D_i 计算每个基本分类器的错分率, 依据错分率进一步度量每个基本分类器的性能, 如式(16)所示。

$$\epsilon^i = \sum_{i=1}^n D_i(i) (f_i(x_i) \neq y_i) \quad (16)$$

根据分类器的错分率 ϵ^i 进行每个基本分类器权重 α_i 的计算, 同时对训练样本的权重分布进行更

新,如式(17)和式(18)所示。

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon^t}{\varepsilon^t} \right) \quad (17)$$

$$D_{t+1} \rightarrow \frac{D_t(i)}{Z^t} \exp(-\alpha_t y_i f_t(x_i)) \quad (18)$$

其中, Z^t 是归一化因子。通过反复学习基本核分类器,迭代完成后根据权重系数对基本分类器进行集成,得到的最终分类器如式(19)所示。

$$f(x) = \text{sign} \left(\sum_{i=1}^T \alpha_i f_i(x) \right) \quad (19)$$

本文算法伪代码如算法2所示。

算法2 AdaBoost-EMKL 算法

输入 训练样本:数据集 $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$

核函数: $K_j(\cdot, \cdot): x \times x \rightarrow \mathbb{R}, j = 1, 2, \dots, M$

初始化训练样本的分布: $D_1(i) = 1/N, i = 1, 2, \dots, N$

输出 强分类器 $f(x) = \text{sign} \left(\sum_{i=1}^T \alpha_i f_i(x) \right)$

1. for $t = 1, 2, \dots, T$ do

2. 根据训练样本的分布 D_t 抽取 n 个样本,得到训练子集

3. 利用混合正则约束求取若干个核函数 K_j 在训练子集

下的最优组合,得到基本分类器 $f_t(x)$

4. 根据训练样本分布 D_t 计算训练错分率:

$$\varepsilon^t = \sum_{i=1}^N D_t(i) (f_t(0_i) \neq y_i)$$

5. 计算基本分类器的权重:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon^t}{\varepsilon^t} \right)$$

6. 更新训练样本的分布:

$$D_{t+1} \rightarrow \frac{D_t(i)}{Z^t} \exp(-\alpha_t y_i f_t(x_i))$$

7. $t \leftarrow t + 1$

8. end for

9. 将基本分类器进行集成得到强分类器:

$$f(x) = \text{sign} \left(\sum_{i=1}^T \alpha_i f_i(x) \right)$$

算法2的输入为相关数据集和初始化的参数,输出为强分类器,第1步~第8步是进行基本分类器的选择。其中,第2步根据样本的分布抽取 n 个样本子集,第3步采用算法1进行基本分类器核权值的选择,第4步计算训练错分率,第5步根据错分率计算当前基本分类器的权值,第6步更新训练样本的分布同时进行归一化操作。进行 T 轮迭代后,结束整个迭代的过程,将得到的基本分类器进行集成为最终的强分类器。AdaBoost-EMKL 算法由于采用了弹性网型正则化选取核函数的权值,避免了核权值稀疏性与非稀疏性所带来的问题。同时该算法具有集成算法的优势,能够通过不断迭代增强分类器的性能。

2.3 时间复杂度

在算法2中,每次进行迭代时,首先都要选择一个基本分类器,如算法2的第3步所示。基本分类器的求解采用算法1中的基本过程。该过程包含了标准 SVM 问题的求解(如算法1的第1步所示)以及二次约束二次规划问题的求解,如算法1的

第4步所示。本文采用的是 SMO 算法求解 SVM 标准问题,其时间复杂度为 $O(n^2)$,二次约束二次规划问题的收敛速度为 $O(n^2)$,因此,算法1总的时间复杂度为 $O(n^2)$ 。而算法2的时间复杂度主要取决于其自身迭代次数与算法1的时间复杂度,因此,算法2的时间复杂度为 $O(T) \times O(n^2)$ 。

3 实验结果与分析

3.1 实验数据与环境

本文采用标准 SVM、SimpleMKL、 L_1 -MKL、 L_p -MKL 以及 MKBoost-D2^[6] 算法作为对比算法,验证本文算法的性能以及可行性。SimpleMKL 与 L_1 -MKL 基于 L_1 范数约束,本文算法与其对比核权值的稀疏性。 L_p -MKL 基于 L_p 范数约束, MKBoost-D2 基于核权值非稀疏的 Adaboost 集成多核方法。上述2种方法在选取核函数的权值时都是非稀疏的,因此能与本文算法形成鲜明的对比。本文实验采用的数据集信息如表1所示,整个实验将在 Microsoft Windows 8(64位),2.7 GHz Intel CPU, 6 GB RAM, Matlab7.8.0 R2009a 平台上进行。

表1 数据集信息

数据集名称	数据集来源	特征数量	特征维数	类别
Ionosphere	UCI	351	34	2
Wdbc	UCI	569	30	2
German. number	Statlog	1 000	24	2
Monks2	UCI	1 033	6	2
Svmguide2	CWH03a	1 243	21	2
Balance-scale	UCI	576	4	4
Segment	Statlog	2 310	19	7

3.2 实验参数设置

本文实验选用的高斯核 σ 的取值分别为 1.5^{-5} , 1.5^{-4} , \dots , 1.5^6 , 2^{-5} , 2^{-4} , \dots , 2^6 , 多项式指数分别为 1, 2, 3。SVM 以及其他算法正则化参数采用5-折交叉验证选择最优参数。设置 L_p -MKL 与本文算法的 p 值为 2、4、6。平衡参数 ν 的值从集合 $\{0.0, 0.1, \dots, 1.0\}$ 中选取。对于 SimpleMKL 算法,采用 SimpleMKL 工具箱进行求解^[15], 当对偶间隙小于 0.001 或者迭代次数超过 500 时停止算法。本文算法采用迭代次数超过 500 次并且核权值相差在 0.001 以内作为停止标准。

3.3 性能比较

3.3.1 分类精度对比

本文对每个算法进行 10 次重复实验,每次实验均从数据集中随机选择一定数量的样本作为训练集,剩余样本作为测试集,各算法的分类精度与训练样本的关系如图1所示。从图1可以看出,随着训练样本数量增多,各算法的分类精度都有所提升,并且在各个数据集中,无论训练样本数量多少,本文算法的分类精度均较高,表明其泛化性能较优。

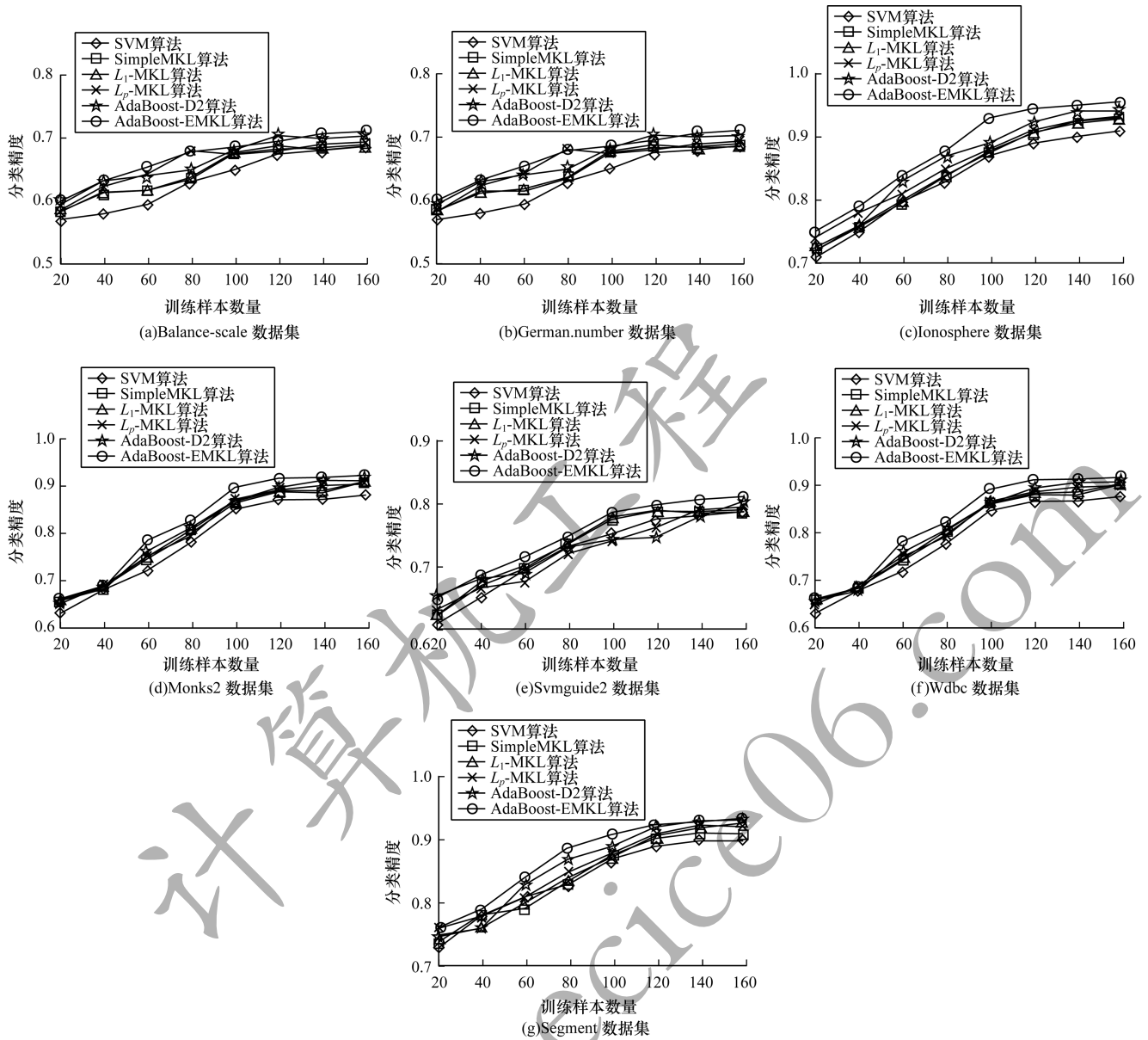


图 1 6 种算法的分类精度与训练样本数之间的关系

随机从 7 个数据集中选择 80% 的样本作为训练集, 剩余样本为测试集。各算法均执行 10 次, 每次参数相同, 在测试集上求出其平均分类精度与标准差, 结果如表 2 所示。

表 2 6 种算法的分类精度与标准差比较

数据集	SVM	SimpleMKL	L_1 -MKL	L_p -MKL	AdaBoost-D2	AdaBoost-FMKL
Ionosphere	0.910 2 ± 0.013 3	0.932 5 ± 0.008 9	0.920 2 ± 0.100 6	0.939 8 ± 0.007 6	0.941 8 ± 0.004 8	0.945 3 ± 0.003 5
Wdbc	0.932 1 ± 0.009 1	0.945 2 ± 0.001 3	0.943 5 ± 0.012 6	0.942 5 ± 0.002 2	0.946 2 ± 0.001 3	0.948 2 ± 0.013 5
German. number	0.692 1 ± 0.014 3	0.692 5 ± 0.002 9	0.701 2 ± 0.130 2	0.712 1 ± 0.003 6	0.713 8 ± 0.003 8	0.715 3 ± 0.073 2
Monks2	0.870 2 ± 0.001 6	0.882 5 ± 0.012 9	0.892 8 ± 0.205 6	0.907 4 ± 0.014 5	0.908 5 ± 0.034 2	0.911 2 ± 0.000 6
Svmguide2	0.785 6 ± 0.145 1	0.789 2 ± 0.006 9	0.804 1 ± 0.070 3	0.809 3 ± 0.007 3	0.811 4 ± 0.002 1	0.815 3 ± 0.016 8
Balance-scale	0.940 2 ± 0.043 1	0.952 8 ± 0.003 7	0.951 2 ± 0.060 6	0.951 2 ± 0.002 1	0.975 6 ± 0.001 1	0.976 7 ± 0.000 8
Segment	0.900 2 ± 0.006 3	0.912 8 ± 0.007 4	0.920 0 ± 0.032 7	0.926 5 ± 0.001 2	0.928 8 ± 0.004 2	0.932 3 ± 0.003 1

从表 2 得到以下结论:

1) 在所有的数据集中 SVM 表现的性能较差。

2) SimpleMKL 基于 L_1 范数约束, 因此, 其与 L_1 -MKL 的分类性能相差较小。

3) L_p -MKL 作为非稀疏类模型, 其泛化性能优

于稀疏模型,但是由于其对噪声敏感并且有冗余信息,分类性能并没有明显优于 L_1 -MKL,且相对于 AdaBoost-D2 分类精度较差。

4) AdaBoost-D2 是在非稀疏性核权值构造的基本分类器基础上集成的强分类器,其分类性能优于 L_p -MKL。

5) Adaboost-EMKL 在选择基本分类器时采用弹性网型正则约束核权值,平衡了核权值的稀疏性与非稀疏性,通过集成学习得到最终的强分类器,因此本文算法在分类性能上优于 AdaBosot-D2。

3.3.2 收敛性对比

在 Ionosphere、Monks2、Segment、Svmguide2 数据集随机选取 80% 样本作为训练集,剩余的为测试集,对比本文算法和 MKBoost-D2 算法的收敛性。算法迭代次数为 30 次,实验次数为 10 次,2 种集成核学习算法的精度与迭代关系如图 2 所示。

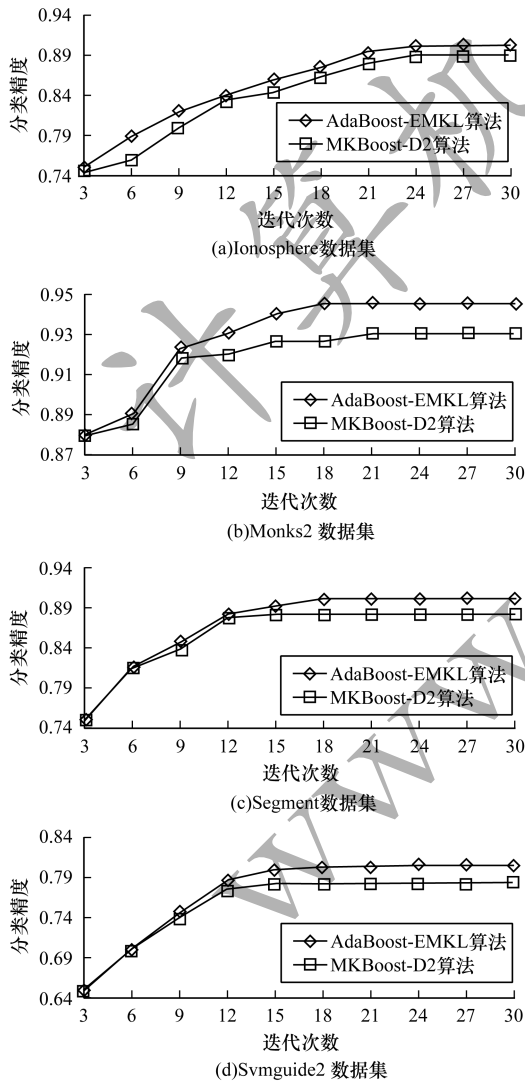


图 2 本文算法的收敛性比较

从图 2 可以看出,算法的分类精度随着迭代次数的增加而提高。在刚开始迭代时,MKBoost-D2 分

类精度高于本文算法,当迭代超过 10 次后,本文算法的分类精度较优,且算法开始收敛,即分类精度逐渐稳定。因此,本文算法的收敛性较好,能在有效的迭代次数内收敛到最优值。

3.3.3 参数 p 对比

为了验证不同 p 值,即在不同范数正则约束情况下,不同算法的时间性能以及核函数数量选择的变化情况,本文在 Ionosphere、Segment、Wdbc 数据集中进行算法评估。随机选择 80% 样本作为训练集,剩余样本为测试集。各算法均执行 10 次,实验结果如表 3~表 5 所示。从表 3~表 5 可以看出, p 值不同本文算法分类精度也不同,因此,选择合适的 p 值对算法至关重要,针对不同的数据集可以采用交叉验证的方法选择出合适的 p 值。当 p = 2 时,本文算法比 L_1 -MKL 选择了更多的核函数。这是因为 L_1 -MKL 的核权值是稀疏的,所以丢弃了许多核函数,忽略一些有用的信息,导致其缺少泛化能力。而 L_p -MKL 选择了所有的核函数,但是其对噪声敏感,因此与 L_1 -MKL 对比,其分类精度并不存在优势,由表 4 可知 L_p -MKL 的分类性能比稀疏的 L_1 -MKL 差。而本文算法采用是弹性网型正则,选择了适量的核函数,所取得的核权值是处于稀疏与非稀疏之间。同时算法的运行时间随着 p 值的变化而不同,当 p 值较小时,算法的运行时间较少,并且在多数数据集中,本文算法运行效率都高于 L_1 -MKL,这是因为当 L_1 -MKL 更新下降方向时,需要进行大量的计算。

表 3 不同范数正则约束情况下 Ionosphere 数据集的测试结果

算法	p	分类精度	核函数数量	运行时间/s
L_p -MKL	2	0.942 1 ± 0.012 5	410	4.5 ± 0.6
	4	0.941 1 ± 0.015 6	410	3.6 ± 0.9
	6	0.938 6 ± 0.069 1	410	5.6 ± 1.2
AdaBoost-EMKL	2	0.946 5 ± 0.008 5	42.5 ± 1.3	7.9 ± 1.5
	4	0.943 2 ± 0.013 6	30.2 ± 1.6	9.6 ± 0.8
	6	0.941 0 ± 0.096 4	21.5 ± 3.1	13.6 ± 2.1
L_1 -MKL	—	0.942 0 ± 0.016 4	35.6 ± 2.9	10.2 ± 2.6

表 4 不同范数正则约束情况下 Segment 数据集的测试结果

算法	p	分类精度	核函数数量	运行时间/s
L_p -MKL	2	0.921 6 ± 0.014 3	251	1.5 ± 0.4
	4	0.920 1 ± 0.012 7	251	1.6 ± 0.2
	6	0.918 4 ± 0.067 7	251	2.4 ± 0.2
AdaBoost-EMKL	2	0.928 9 ± 0.018 5	21.5 ± 1.1	4.1 ± 0.3
	4	0.923 2 ± 0.013 1	17.3 ± 0.6	4.6 ± 0.2
	6	0.919 2 ± 0.015 8	14.5 ± 0.5	7.1 ± 0.1
L_1 -MKL	—	0.924 5 ± 0.014 7	19.6 ± 0.9	3.2 ± 0.6

表 5 不同范数正则约束情况下 Wdbc 数据集的测试结果

算法	p	分类精度	核函数数量	运行时间/s
L_p -MKL	2	0.943 5 ± 0.012 2	396	2.7 ± 0.2
	4	0.941 1 ± 0.027 5	396	3.6 ± 0.9
	6	0.9436 ± 0.061 2	396	5.6 ± 1.2
AdaBoost-EMKL	2	0.946 1 ± 0.003 5	32.7 ± 1.1	8.9 ± 1.5
	4	0.943 2 ± 0.013 1	17.8 ± 1.4	12.4 ± 1.8
L_1 -MKL	6	0.941 0 ± 0.018 6	16.5 ± 1.2	18.4 ± 2.6
	—	0.942 0 ± 0.014 5	25.4 ± 2.7	28.2 ± 2.1

3.3.4 参数 ν 对比

本文选择 Segment 与 Wdbc 数据集测试平衡参数 ν 对于分类精度的影响,实验结果如图 3 所示。

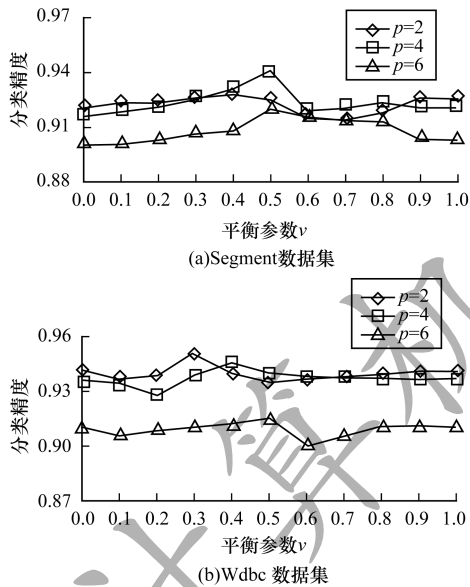


图 3 Adaboost-EMKL 算法精度与 ν 取值的关系

当 $\nu = 0$ 时,由式(12)可知,弹性网型正则退化为 L_p 正则,此时其分类精度是 L_p -MKL 的结果。同理可得,当 $\nu = 1$ 时,弹性网型正则为 L_1 正则。当 $\nu \in (0, 1)$ 时,对于不同的数据集,最优分类精度随着 ν 取值不同而变化,因此本文采用交叉验证来对 ν 和 p 进行调整,以获得更好的分类精度。

本文在 Segment 数据集中测试 Adaboost-EMKL 算法的 ν 与核函数数量的关系,结果如图 4 所示。从图 4 可以看出,随着 ν 值的增大,即核权值更稀疏,所选择的核函数数量逐渐变小,并且该规律不受其他因素影响,即核函数数量只与 ν 相关。

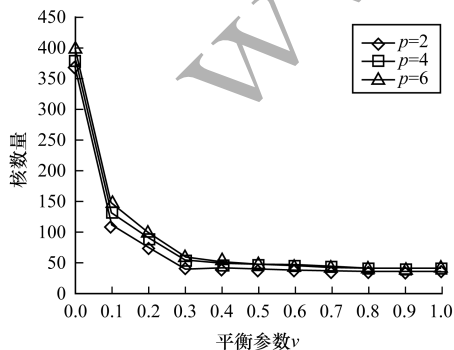


图 4 Adaboost-EMKL 算法中 ν 取值与选择核数量的关系

4 结束语

本文提出 AdaBoost-EMKL 算法,利用 AdaBoost 集成学习的设计思想,将基本分类器集成为强分类器。在迭代求取基本分类器时,采用弹性网型正则化约束选择核权值,以提高模型的泛化能力并降低其对噪声敏感性。实验结果表明,该算法不仅具有 AdaBoost 集成算法的优点,而且能够平衡 L_1 正则的稀疏性与 L_p 正则的非稀疏性,在较少的迭代次数内,获得分类精度高、运行速度快且性能稳定的分类器。

参考文献

- [1] CHAPPELLE O, VAPNIK V, BOUSQUET O, et al. Choosing multiple parameters for support vector machines[J]. Machine Learning, 2002, 46(1/2/3): 131-159.
- [2] KLOFT M, BREFELD U, SONNENBURG S. Efficient and accurate L_p -norm multiple kernel learning [C]// Proceedings of Advances in Neural Information Processing Systems. [S. l.]: NIPS Foundation, Inc., 2009:997-1005.
- [3] SUN Tao, JIAO Licheng, LIU Fang, et al. Selective multiple kernel learning for classification with ensemble strategy[J]. Pattern Recognition, 2013, 46(11): 3081-3090.
- [4] YANG Haiqin, XU Zenglin, YE Jieping, et al. Efficient sparse generalized multiple kernel learning [J]. IEEE Transactions on Neural Networks, 2011, 22(3): 433-446.
- [5] KLOFT M, BREFELD U, SONNENBURG S, et al. Non-sparse regularization and efficient training with multiple kernels [EB/OL]. [2018-03-01]. https://arxiv.org/abs/1003.0079.
- [6] XIA Hao, HOI S C H. MKBoost: a framework of multiple kernel boosting [J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(7): 1574-1586.
- [7] CITI L. Elastic-net constrained multiple kernel learning using a majorization-minimization approach [C]// Proceedings of Computer Science and Electronic Engineering Conference. Washington D. C., USA: IEEE Press, 2015: 29-34.
- [8] 武征鹏, 张学工. 弹性多核学习 [J]. 自动化学报, 2011, 37(6): 693-699.
- [9] 王庆超, 付光远, 汪洪桥, 等. 基于局部空间变稀疏约束的多核学习方法 [J]. 电子学报, 2018, 46(4): 930-937.
- [10] 何佳佳, 陈秀宏, 田进, 等. 基于牛顿梯度优化的弹性多核学习 [J]. 传感器与微系统, 2018(2): 136-139.
- [11] 张仁峰, 吴小俊, 陈素根. 通用稀疏多核学习 [J]. 计算机应用研究, 2016, 33(1): 21-27.
- [12] 梁军, 李世浩, 张飞云, 等. 基于半无限规划的弹性多核学习算法 [J]. 华中科技大学学报(自然科学版), 2015, 43(8): 103-106.
- [13] 曹莹, 苗启广, 刘家辰, 等. AdaBoost 算法研究进展与展望 [J]. 自动化学报, 2013, 39(6): 745-758.
- [14] SCHOLKOPF B, SMOLA A J. Learning with kernels: support vector machines, regularization, optimization, and beyond [M]. Cambridge, USA: MIT Press, 2001.
- [15] RAKOTOMAMONJY A, BACH F R, CANU S, et al. SimpleMKL [J]. Journal of Machine Learning Research, 2008, 9(3): 2491-2521.