

DOI: 10.7641/CTA.2014.30690

# 迭代再权共轭梯度 $q$ 范数正则化线性最小二乘-支持向量机分类算法

刘建伟<sup>†</sup>, 黎海恩, 刘媛, 付捷, 罗雄麟

(中国石油大学 自动化研究所, 北京 102249)

**摘要:**  $L_2$ 范数罚最小二乘-支持向量机(square support vector machine algorithm, LS-SVM)分类器是得到广泛研究和使用的机器学习算法,其算法中正则化阶次是事先给定的,预设 $q = 2$ . 本文提出 $q$ 范数正则化LS-SVM分类器算法,  $0 < q < \infty$ ,把 $q$ 取值扩大到有理数范围. 利用网格法改变正则化权衡参数 $c$ 和正则化阶次 $q$ 的值,在所选的 $c$ 和 $q$ 值上,使用迭代再权方法求解分类器目标函数,找出最小分类预测误差值,使预测误差和特征选择个数两个性能指标得到提高. 通过对不同领域的实际数据进行实验,可以看到提出的分类算法分类预测更加准确同时可以实现特征选择,性能优于 $L_2$ 范数罚LS-SVM.

**关键词:**  $q$ 范数正则化; 最小二乘-支持向量机( LS-SVM); 迭代再权共轭梯度法

中图分类号: TP181 文献标识码: A

## $q$ -norm regularizing least-square-support-vector-machine linear classifier algorithm via iterative reweighted conjugate gradient

LIU Jian-wei<sup>†</sup>, LI Hai-en, LIU Yuan, FU Jie, LUO Xiong-lin

(Research Institute of Automation, China University of Petroleum, Beijing 102249, China)

**Abstract:** The  $L_2$ -norm penalty least-square-support-vector-machine algorithm (LS-SVM) has been extensively studied and is probably the most widely used machine learning algorithm. The regularization parameter in LS-SVM is predetermined with default value  $q = 2$ . Based on the iterative reweighted conjugate gradient algorithm, the  $q$ -norm regularizing LS-SVM is proposed with  $0 < q < 1$ , a rational number. We design a grid method to change the value of two adjustable parameters, the regularization parameter  $c$  and the order  $q$  of regularization, by performance indicators of prediction error rate. On the selected values of  $c$  and  $q$ , using the iterative reweighted conjugate gradient algorithm for solving classification object function and finding the minimum prediction error, we can improve the feature selection and predict the error rate. The experimental results on real datasets in different fields indicate that the prediction performance is more accurate than  $L_2$ -norm LS-SVM, and can carry out feature selection.

**Key words:**  $q$ -norm regularization; LS-SVM; Iterative reweighted conjugate gradient method

### 1 引言(Introduction)

Vapnik提出的支持向量机(support vector machine, SVM)为当前主流的机器学习算法,特别是 $L_2$ 范数支持向量机( $L_2$ -norm support vector machine,  $L_2$ -SVM<sup>[1]</sup>). 给定样例-类标签对 $(\mathbf{x}_i, y_i)$ ,  $i = 1, \dots, m$ ,  $\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, +1\}$ , SVM利用训练样本求解

$$\min_{\mathbf{w}} J(\mathbf{w}) = \text{pen}(\mathbf{w}) + C \sum_{i=1}^m L(y_i, \mathbf{w}\mathbf{x}_i),$$

得到判别函数 $f(x) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$ ,  $J(\mathbf{w})$ 中 $\text{pen}(\mathbf{w})$ 是正则化项,  $L(y_i, \mathbf{w}\mathbf{x})$ 为损失函数. 使用铰链损失的 $L_2$ -SVM问题为

$$\min_{\mathbf{w}} J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^m \max(1 - y_i \mathbf{w}^T \mathbf{x}_i, 0). \quad (1)$$

$L_2$ 范数最小二乘-支持向量机( $L_2$ -norm least square support vector machine,  $L_2$ -LS-SVM)是SVM的变种,最大特点是将SVM中的不等式约束转换为等式约束

$$\min_{\mathbf{w}, \xi} \frac{C}{2} \sum_{i=1}^m \xi_i^2 + \frac{1}{2} \|\mathbf{w}\|_2^2, \quad (2)$$

$$\text{s.t. } y_i \cdot [\langle \mathbf{x}_i, \mathbf{w} \rangle + b] = 1 - \xi_i, \quad i = 1, \dots, m,$$

并使用共轭梯度算法求解对偶问题<sup>[2-3]</sup>

收稿日期: 2013-07-08; 录用日期: 2013-12-02.

<sup>†</sup>通信作者. E-mail: liujw@cup.edu.cn; Tel.: 86 10-89733306.

基金项目: 国家“973”重点基础研究计划资助项目(2012CB720500); 国家自然科学基金资助项目(21006127); 中国石油大学(北京)基础学科研究资助项目(JCXX-2011-07).

$$\begin{bmatrix} 0 & \mathbf{y}^T \\ \mathbf{y} & \mathbf{\Omega} + I/C \end{bmatrix} \begin{bmatrix} b \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1}_v \end{bmatrix}, \quad (3)$$

这里:  $\mathbf{a} = [a_1 \cdots a_n]$ 为拉格朗日乘子,  $\mathbf{\Omega} = \mathbf{z}^T \mathbf{z}$ ,  $\mathbf{z}^T = [x_1 y_1 \cdots x_m y_m]$ ,  $\mathbf{y} = [y_1 \cdots y_m]$ ,  $\mathbf{1}_v = [1 \cdots 1]$ .

$L_2$ -LS-SVM的好处是分类和回归的对偶形式是一样的,且预测性能与 $L_2$ -SVM相近.但是缺点是求得模型不稀疏,分类算法对噪声敏感,不鲁棒.当前,有两种实现稀疏化的方法:训练后去除一些绝对值小的对偶变量再训练,反复执行这个过程直到预测准确率明显下降为止;第2种方法是人为地去除一些对偶变量后直接训练.这种稀疏化是在对偶变量上实现的,没有讨论对样例的特征选择问题.最近, Lopez提出1-Norm SVM算法<sup>[4]</sup>,把松弛变量绝对值和加入目标函数,使得算法更鲁棒, Lopez还提出了使用 $L_0$ 范数对式(2)中的 $\mathbf{w}$ 的正则化替换为对对偶变量 $\mathbf{a}$ 的正则化<sup>[5]</sup>

$$\min_{\mathbf{w}, \xi, \mathbf{a}} \frac{C}{2} \sum_{i=1}^m \xi_i^2 + \frac{1}{2} \sum_{j=1}^n \lambda_j a_j^2,$$

$$\text{s.t. } y_i \cdot \left[ \left( \sum_{k=1}^m a_k x_k \right) \mathbf{x}_i + b \right] = 1 - \xi_i, \quad i = 1, \dots, m.$$

关于 $q$ 范数正则化SVM,文献[6]中提出 $q$ 范数正则化SVM,在目标函数(1)中 $L_2$ 范数换成 $L_q$ 范数 $\|\mathbf{w}\|_q = \left( \sum_{j=1}^m |w_j|^q \right)^{1/q}$ ,  $0 < q < 1$ ,目标函数(1)中的变量 $\mathbf{w}$ 用变量 $\mathbf{v}$ 替代,增加一个约束条件 $-\mathbf{v} < \mathbf{w} < \mathbf{v}$ ,去除目标函数中的绝对值,使用渐进线性近似算法<sup>[7]</sup>求解非凸优化问题.使用文献[8-12]的思路,文献[13]提出了权 $L_q$ 范数LS-SVM算法,利用标准的LS-SVM的解计算 $s_i = \sum_{i=1}^m a_i y_i \mathbf{x}_i x_j + b$ ,给每一个 $\xi_i^2$ 一个权

$$\mu_i = \frac{s_i - \min_{k=1, \dots, m} s_k}{\max_{k=1, \dots, m} s_k - \min_{k=1, \dots, m} s_k},$$

式(2)中的 $\frac{C}{2} \sum_{i=1}^m \xi_i^2$ 换为 $\frac{C}{2} \sum_{i=1}^m \mu_i \xi_i^2$ 得到该问题式(3)相似的对偶问题,只是矩阵 $\mathbf{\Omega}$ 中各元素多了一个 $(C\mu_i)^{-1}$ 项,最终求解的问题为

$$\begin{aligned} & \min_{\beta} \|\beta\|_q, \\ & \text{s.t. } \begin{bmatrix} 0 & \mathbf{y}^T \\ \mathbf{y} & \mathbf{\Omega} + I/C \end{bmatrix} \beta = \begin{bmatrix} 0 \\ \mathbf{1}_v \end{bmatrix}, \end{aligned}$$

这里 $\beta = [b \ \mathbf{a}]^T$ ,  $1 \leq q \leq 2$ ,最终使用进化算法确定模型参数,并用Enrique Brito提出的IRMS方法求解该问题<sup>[14]</sup>. Kloft提出的 $L_q$ 范数多核SVM学习算法<sup>[15]</sup>,限制 $q > 1$ .

最近,有许多研究 $L_1, L_2$ 范数罚及其组合罚的SVM文献,说明使用不同阶次范数罚构造分类器是重要的研究方向. Zhu Ji提出基于 $L_1$ 范数罚 $\text{pen}(\mathbf{w}) =$

$\|\mathbf{w}\|_1$ 正则化SVM,能够同时实现特征选择和模式分类<sup>[16]</sup>, Liu Yufeng提出了组合 $L_0$ 和 $L_1$ 范数罚 $\text{pen}(\mathbf{w}) = \lambda \|\mathbf{w}\|_0 + (1 - \lambda) \|\mathbf{w}\|_1$ 正则化SVM分类算法<sup>[17]</sup>,通过范数组合,能够克服 $L_0$ 范数罚SVM分类器数值不稳定和 $L_1$ 范数罚SVM选择稀疏程度不够的问题, Liu Yufeng同时提出根据数据自适应选择 $L_1, L_2$ 范数 $\text{pen}(\mathbf{w}) = \|\mathbf{w}\|_q (q \in \{1, 2\})$ 的SVM分类算法<sup>[18]</sup>. 由于 $L_1$ 范数罚对相关组特征只取一个的缺点, Hui Zou提出了类似于不动点理论中同伦法的弹性网罚 $\text{pen}(\mathbf{w}) = \lambda \|\mathbf{w}\|_1 + (1 - \lambda) \|\mathbf{w}\|_2$ 分类算法,能够实现组特征全部提出<sup>[19]</sup>.

基于逻辑斯蒂广义线性统计模型的 $L_1, L_2$ 范数正则化分类算法也是当前研究的热点. Su-In Lee提出 $L_1$ 范数罚正则化逻辑斯蒂回归算法<sup>[20]</sup>, Zhenqiu Liu提出了 $L_q$ 罚和弹性网罚正则化逻辑斯蒂分类算法<sup>[21]</sup>,并应用于癌症分类,使用最大受试者工作特征(receiver-operating characteristic, ROC)曲线和曲线下面积(area under the curve, AUC)方法确定正则化参数 $q$ 的值.另一个值得关注的研究热点是引入非凸罚函数构造分类回归算法的研究<sup>[22-24]</sup>,在这些算法中罚函数为分数范数 $L_q, 0 < q < 1$ .

从以上文献中可以看出,研究不同正则化阶次的分类算法是当前的研究方向,而且所有以上文献均对某一种范数阶次的正则化LS-SVM分类算法进行研究,但根据笔者的实验研究,不同的数据适用于不同正则化阶次,范数 $L_q$ 的阶次 $q$ 可能的范围很宽,  $0 < q < \infty$ .

上述文献的LS-SVM分类算法中,均把正则化阶次 $q$ 事先定为某一个固定值问题,通过大量实验,笔者发现:最优的分类效果并非出现在 $q$ 等于某一个事先给定的值,而是可能出现在某一个区间内,具体出现在什么位置,需要利用训练样本在 $q$ 的取值范围内求解 $q$ 范数正则化LS-SVM分类器目标函数,寻找 $q$ 为变量时的最优分类器性能.笔者提出求解以下 $q$ 范数正则化LS-SVM分类器( $L_q$ -LS-SVM)问题:

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \sum_{i=1}^m \xi_i^2 + \frac{c}{q} \|\mathbf{w}\|_q^q, \quad (4)$$

$$\text{s.t. } y_i (\mathbf{x}_i^T \mathbf{w} + b) = 1 - \xi_i, \quad i = 1, 2, \dots, m,$$

这里 $0 < q < \infty$ .注意到目标函数(4)中,当取 $q = 2$ 时,此问题变为标准的 $L_2$ 范数罚LS-SVM,当取 $q = 1$ 时,此问题变为标准的 $L_1$ 范数罚LS-SVM.此问题在 $0 < q < 1$ 时为非凸问题, $q = 1$ 时为非平滑问题, $q = 2$ 时为二次规划问题,那么是否存在有效的方法求解该问题呢?文献[25-27]中提出利用共轭梯度算法求解非凸非平滑问题,可见 $L_q$ -LS-SVM问题可以解决.最近, Tianyi Zhou在文献[28]提出了基于Nesterov梯度下降(Nesterov's method)SVM分类算法,证明基于梯度的算法优于现有的各种SVM解法,如流行的SVM-Perf<sup>[29]</sup>, Pegasos<sup>[30]</sup>, SVM-Light<sup>[31]</sup>和LIBSVM<sup>[32]</sup>,可

以看出传统的梯度下降方法既可以解决非凸非平滑问题,同时与其他方法比较有性能优势.基于此,本文提出两阶段分类算法,首先利用网格法对两个变量 $q$ 和 $c$ 的取值范围进行划分,参数 $q$ 的候选值范围为 $2^0$ 到 $2^{13}$ 之间,以指数分布选取共14个候选值;参数 $c$ 的候选值范围为 $2^{-4}$ 到 $2^{13}$ 之间,共18个候选值.在候选值网格点上利用本文提出的迭代再权共轭梯度算法求解 $q$ 和 $c$ 为固定值时的问题(4),通过比较实验结果,笔者发现对于许多高维小样本数据,取得最优分类性能时, $q$ 值是不固定的.

本文提出的迭代再权共轭梯度算法第 $k$ 次迭代时,利用

$$|\mathbf{w}^{(k)} + \lambda^{(k)} \mathbf{d}^{(k)}|^{q-2} \approx |\mathbf{w}^{(k-1)} + \lambda^{(k-1)} \mathbf{d}^{(k-1)}|^{q-2},$$

即第 $k$ 次迭代时, $|\mathbf{w}^{(k)} + \lambda^{(k)} \mathbf{d}^{(k)}|^{q-2}$ 的值用第 $k-1$ 次的已知值 $|\mathbf{w}^{(k-1)} + \lambda^{(k-1)} \mathbf{d}^{(k-1)}|^{q-2}$ 替代,从而使得

$$\sum_{j=1}^n |w_j^{(k)} + \lambda_i^{(k)} d_i^{(k)}|^q \approx \sum_{j=1}^n |w_j^{(k-1)} + \lambda_i^{(k-1)} d_i^{(k-1)}|^{q-2} |w_j^{(k)} + \lambda_i^{(k)} d_i^{(k)}|^2,$$

即把原来的 $q$ 范数用近似二范数替代,即实现了平滑,同时又凸化了原问题.

本文提出的算法不使用核映射,数据不经过非线性核映射,理由是:首先核的选择问题是一个难题,不同的数据可能适用不同的核,而且即使核的形式确定后,核参数的选择又是一个难题;其次,要实现特征选择,必须显式求出 $\mathbf{w}$ 来,这对于高斯核等无穷维核映射,现在还无法实现<sup>[33-34]</sup>.

随着基因工程的发展,人类对蛋白质的研究不断深入,基因的调控规律成为大家研究的热点<sup>[35-36]</sup>,由于基因微阵列数据的特点,机器学习理论成为主要的研究工具.对于癌症的研究,由于癌症细胞基因对应的矩阵表示形式,样本维数多,可高达上千维.但由于患病机率,测试成本等原因,实际检测只能提取低维特征,同时只能得到少量的样本个数,基因微阵列数据有高维数,低样本个数的特点,对这种数据的分类预测特别有挑战性,是本文的实验对象.

本文主要研究内容有5个部分:

1) 提出了 $q$ 范数正则化LS-SVM两阶段分类器设计模式,分类算法能够在 $0 < q \leq \infty$ 范围内,不只局限于某个固定的正则化范数阶次,改变正则化范数的阶次 $q$ ,使得分类器预测误差最小.

2) 提出了求解 $L_q$ -LS-SVM的两阶段算法,利用网格法对两个变量 $q$ 和 $c$ 的取值范围进行划分,在网格点上利用笔者提出的迭代再权共轭梯度算法求解 $q$ 和 $c$ 为固定值时的 $L_q$ -LS-SVM问题.

3) 利用笔者提出的算法,通过大量的实验发现,

不同的高维小样本数据上,通过改变正则化阶次 $q$ 的值,可以改进分类器的预测准确率.

4) 笔者提出的算法训练得到的模型向量 $\mathbf{w}$ 的大多数分量为零,这样使得模型向量 $\mathbf{w}$ 分量为零的维所对应的样本的相应坐标分量不再参与模型的构造,只有那些模型向量 $\mathbf{w}$ 的分量不为零的维所对应的样本的坐标分量才参与模型构造,故实现了特征选择,同时实现了模型稀疏化和鲁棒化.

5) 采用 $L_q$ -LS-SVM分类算法对癌症微阵列基因分类数据、Tr11文本数据、Arcene质谱仪癌症分类数据集、Sylva森林覆盖类型数据集、PIE-32  $\times$  32和Yaleb-32  $\times$  32人脸图进行分类预测和特征选择实验研究,通过选择不同的正则化参数 $c$ 和正则化阶次 $q$ 值,得到的分类结果比LS-SVM更加准确.

## 2 $L_q$ -LS-SVM分类器( $L_q$ -LS-SVM classifier)

### 2.1 目标函数(Objective function)

假定训练样本集为 $\mathbf{X}^{m \times n}$ ,其中 $m \ll n$ .这里 $m$ 为样例个数, $n$ 为样例维数,定义 $\mathbf{X}^{m \times n}$ 的一行 $\mathbf{x}_i \in \mathbb{R}^m$ 为一个样例,相应的类标签为 $y_i \in \{\pm 1\}$ , $\mathbf{w} \in \mathbb{R}^m$ ,无约束 $L_q$ -LS-SVM分类算法的目标函数为

$$\min_{\mathbf{w}, b} f(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^m L(y_i, \mathbf{x}_i^T \mathbf{w} + b) + \eta \|\mathbf{w}\|_q^q, \quad (5)$$

其中: $\|\mathbf{w}\|_q = (\sum_{j=1}^n |w_j|^q)^{1/q}$ ,  $0 < q < \infty$ ,  $w_j$ 是 $\mathbf{w}$ 的第 $j$ 个元素分量, $L(\cdot)$ 是损失函数, $\eta$ 是控制变量稀疏程度的正则化参数.

$L_q$ -LS-SVM分类算法的目标函数为

$$\min_{\mathbf{w}, \boldsymbol{\xi}} f_1(\mathbf{w}, \boldsymbol{\xi}) = \min_{\mathbf{w}, \boldsymbol{\xi}} \frac{1}{2} \sum_{i=1}^m \xi_i^2 + \frac{c}{q} \|\mathbf{w}\|_q^q, \quad (6)$$

s.t.  $y_i(\mathbf{x}_i^T \mathbf{w} + b) = 1 - \xi_i, i = 1, 2, \dots, m,$

式中的 $c$ 与式(5)中的 $\eta$ 具有一一对应的关系<sup>[37]</sup>.为计算简便,引入 $\mathbf{w}^T := [w_1 \ \dots \ w_n \ b]^T$ ,  $\mathbf{x}^T := [x_1 \ \dots \ x_n \ 1]$ ,则得到无偏差项的简化表示形式:

$$\min_{\mathbf{w}, \boldsymbol{\xi}} f_2(\mathbf{w}, \boldsymbol{\xi}) = \min_{\mathbf{w}, \boldsymbol{\xi}} \frac{1}{2} \sum_{i=1}^m \xi_i^2 + \frac{c}{q} \|\mathbf{w}\|_q^q, \quad (7)$$

s.t.  $y_i(\mathbf{x}_i^T \mathbf{w}) = 1 - \xi_i, i = 1, 2, \dots, m.$

把约束条件代入目标函数,消去松弛变量 $\xi_i$ 得

$$\min_{\mathbf{w}} f_2(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^m (1 - y_i \cdot (\mathbf{x}_i^T \mathbf{w}))^2 + \frac{c}{q} \|\mathbf{w}\|_q^q. \quad (8)$$

### 2.2 共轭梯度步长 $\lambda^{(k)}$ 求解(Solution for conjugate gradient step-size)

$$\text{令 } h(\mathbf{w}) = \|\mathbf{w}\|_q = (\sum_{j=1}^n |w_j|^q)^{1/q}, k(\mathbf{w}) = h^q(\mathbf{w}),$$

由于

$$\text{sgn } w_j = \frac{w_j}{|w_j|},$$

当 $0 < q < \infty$ 时, 其导数为

$$\frac{\partial k(\mathbf{w})}{\partial w_j} = q|w_j^{q-2}| \cdot w_j. \quad (9)$$

对函数 $f(\mathbf{w})$ 关于变量 $w_j$ 求导, 得一阶偏导

$$\frac{\partial f}{\partial w_j} = \sum_{i=1}^m y_i \cdot x_{i,j} \cdot (y_i \cdot (\mathbf{x}_i^T) - 1) + c|w_j|^{q-2} \cdot w_j, \quad j = 1, 2, \dots, n. \quad (10)$$

定义 $g^{(k)} := \frac{\partial f}{\partial \mathbf{w}}$ 是梯度 $g$ 第 $k$ 次迭代的估计值,  $\mathbf{w}^{(k)}$ 为第 $k$ 次迭代的模型向量的估计值,  $\lambda^{(k)}$ 为第 $k$ 次迭代的步长. 梯度下降模型可使用下面的更新规则:

$$\begin{aligned} \mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - \lambda^{(k)} g^{(k)}, \\ g^{(k)} &= \sum_{i=1}^m y_i x_{i,j} \cdot (y_i \cdot (\mathbf{x}_i^T \cdot \mathbf{w}^{(k)} - 1) + c|w_j^{(k)}|^{q-2} \cdot w_j^{(k)}). \end{aligned}$$

选用共轭梯度法来提高算法的性能时, 先给定 $\mathbf{w}$ 的初始值 $w_0$ , 初始共轭梯度方向 $d^{(1)}$ 等于梯度方向, 即 $d^{(1)} = -g^{(1)}$ . 在第 $k$ 次迭代时, 共轭梯度法的权值更新公式为

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)}, \quad (11)$$

代入目标函数式(8)得

$$\begin{aligned} \min f_2(\mathbf{w}) &= \frac{1}{2} \sum_{i=1}^m (1 - y_i \cdot (\mathbf{x}_i^T (\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)})))^2 + \\ &\frac{c}{q} \|\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)}\|_q^q. \end{aligned} \quad (12)$$

上式中利用

$|\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)}|^{q-2} \approx |\mathbf{w}^{(k-1)} + \lambda^{(k-1)} d^{(k-1)}|^{q-2}$ , 即第 $k$ 次迭代时上式中 $|\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)}|^{q-2}$ 的值用第 $k-1$ 次的值 $|\mathbf{w}^{(k-1)} + \lambda^{(k-1)} d^{(k-1)}|^{q-2}$ 替代, 则得

$$\begin{aligned} \min f_2(\mathbf{w}) &\approx \frac{1}{2} \sum_{i=1}^m (1 - y_i \cdot (\mathbf{x}_i^T (\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)})))^2 + \\ &\frac{c}{q} \sum_{j=1}^n |\mathbf{w}_i^{(k-1)} + \lambda_i^{(k-1)} d_i^{(k-1)}|^{q-2} |w_i^{(k)} + \lambda_i^{(k)} d_i^{(k)}|^2. \end{aligned} \quad (13)$$

假定第 $k$ 次迭代的模型向量 $\mathbf{w}^{(k)}$ 和共轭梯度向量 $d^{(k)}$ 已知, 式(13)对 $\lambda^{(k)}$ 求最小:

$$\begin{aligned} \min_{\lambda^{(k)}} f_2(\lambda) &\approx \frac{1}{2} \sum_{i=1}^m (y_i^2 \cdot (\mathbf{x}_i^T)^2 (\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)})^2 - \\ &2y_i \cdot (\mathbf{x}_i^T \cdot (\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)}))) + \frac{c}{q} |\mathbf{w}^{(k-1)} + \\ &\lambda^{(k-1)} d^{(k-1)}|^{q-2} |\mathbf{w}^{(k)} + \lambda^{(k)} d^{(k)}|^2 = \end{aligned}$$

$$\begin{aligned} &\frac{1}{2} \sum_{i=1}^m (y_i^2 \cdot (\mathbf{x}_i^T)^2 \cdot (\lambda^{(k)} d^{(k)})^2 + \\ &2y_i^2 (\mathbf{x}_i^T)^2 \cdot \mathbf{w}^{(k)} \lambda^{(k)} d^{(k)} - 2y_i \cdot (\mathbf{x}_i^T \cdot \lambda^{(k)} d^{(k)})) + \\ &\frac{c}{q} |\mathbf{w}^{(k-1)} + \lambda^{(k-1)} d^{(k-1)}|^{q-2} (2\mathbf{w}^{(k)} \cdot \lambda^{(k)} d^{(k)} + \\ &(\lambda^{(k)} d^{(k)})^2). \end{aligned} \quad (14)$$

上式中, 凡是与 $\lambda^{(k)}$ 无关的项, 在求最优值时均消去.

定义 $c_1 = \frac{c}{q} |\mathbf{w}^{(k-1)} + \lambda^{(k-1)} d^{(k-1)}|^{q-2}$ , 对式(14)求导得

$$\begin{aligned} \frac{\partial f_2(\lambda^{(k)})}{\partial \lambda^{(k)}} &\approx \sum_{i=1}^m (y_i \cdot (\mathbf{x}_i^T)^2 \lambda^{(k)} (d^{(k)})^2 + \\ &2y_i^2 \cdot (\mathbf{x}_i^T)^2 \cdot \mathbf{w}^{(k)} d^{(k)} - 2y_i \cdot (\mathbf{x}_i^T \cdot d^{(k)})) + \\ &c_1 (2\mathbf{w}^{(k)} \cdot d^{(k)} + 2\lambda^{(k)} \cdot (d^{(k)})^2). \end{aligned} \quad (15)$$

令 $\frac{\partial f_2(\lambda^{(k)})}{\partial \lambda^{(k)}} = 0$ , 解得共轭梯度步长为

$$\begin{aligned} \lambda^{(k)} &\approx \frac{\sum_{i=1}^m y_i \cdot (\mathbf{x}_i^T d^{(k)}) - y_i^2 \cdot (\mathbf{x}_i^T)^2 \cdot \mathbf{w}^{(k)} d^{(k)} - c_1 \mathbf{w}^{(k)} \cdot d^{(k)}}{y_i^2 \cdot (\mathbf{x}_i^T)^2 \cdot (d^{(k)})^2 + c_1 (d^{(k)})^2}. \end{aligned} \quad (16)$$

### 2.3 再权共轭梯度迭代模型求解(Solution for iterative reweighted conjugate gradient model)

由于 $c_1 = \frac{c}{q} |\mathbf{w}^{(k-1)} + \lambda^{(k-1)} d^{(k-1)}|^{q-2}$ , 目标函数(8)变为

$$\min_{\mathbf{w}} f_2(\mathbf{w}) = \frac{1}{2} (1 - \mathbf{Y} \mathbf{X} \mathbf{w})^T (1 - \mathbf{Y} \mathbf{X} \mathbf{w}) + c_1 \mathbf{w}^T \mathbf{w}, \quad (17)$$

这里:  $\mathbf{Y} = \text{diag}\{y_1, y_2, \dots, y_m\}$ ,  $\mathbf{X} = (x_1^T, x_2^T, \dots, x_m^T)^T$ ,  $\mathbf{w} = (w^{(1)}, w^{(2)}, \dots, w^{(m)})^T$ .

把式(17)写成二次函数的形式:

$$\min_{\mathbf{w}} f_2(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{G} \mathbf{w} + \mathbf{b}^T \mathbf{w} + a, \quad (18)$$

其中:  $\mathbf{G} = (\mathbf{Y} \mathbf{X})^T (\mathbf{Y} \mathbf{X}) + c_1 \mathbf{I}$ ,  $\mathbf{b} = -2\mathbf{Y} \mathbf{X}$ ,  $a = 1$ . 由文献[38]可得到 $\beta_k$ 的表达式:

$$\beta_k = \frac{g_{k+1}^T (g_{k+1}^T - g_k^T)}{d_k^T (g_{k+1}^T - g_k^T)} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}. \quad (19)$$

由此得到共轭梯度各迭代模型为

$$\begin{cases} d^{(1)} = -g^{(1)}, \\ d^{(k+1)} = -g^{(k+1)} + \beta_{k+1} \cdot d^{(k)}, \\ \beta_{k+1} = \frac{\langle g^{(k+1)}, g^{(k+1)} \rangle}{\langle g^{(k)}, g^{(k)} \rangle}. \end{cases} \quad (20)$$

最终得到共轭梯度法的迭代公式为

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \lambda d^{(k)},$$

其中:  $d^{(k)}$ 是迭代第 $k$ 次对应的共轭梯度方向,  $\langle \cdot, \cdot \rangle$ 表示

内积,  $\lambda$ 是每次的线性搜索的步长.

注意到共轭梯度模型不能保证目标函数值单调变化, 因此当  $\langle g^{(k)}, d^{(k)} \rangle \leq 0$ , 使用  $-g^{(k)}$  代替  $d^{(k)}$  作为搜索方向, 来保证算法在下降方向是收敛的. 算法中  $f^{(k)}$  是第  $k$  次迭代的目标函数值, 当目标函数的变化量大于设定值  $\tau_1$ : 可表示为  $|f^{(k)} - f^{(k-1)}| > \tau_2$ ; 而且共轭梯度方向和梯度方向之间的夹角小于设定角度  $\tau_2$ , 即  $\angle(g^{(k)}, d^{(k)}) < \tau_2$  时, 笔者使用  $d^{(k)}$  作为搜索方向, 否则, 当  $\angle(g^{(k)}, d^{(k)}) \geq \tau_2$  时, 共轭方向不再是下降方向, 使用  $-g^{(k)}$  为下降方向. 在笔者的算法中, 设置  $\tau_1 = 1e - 5$ ,  $\tau_2 = \frac{4}{9}\pi$ .

## 2.4 再权共轭梯度迭代算法(Iterative algorithm of reweighted conjugate gradient)

$L_q$ -LS-SVM分类算法的计算步骤:

1) 首先对数据集进行处理, 得到样  $(x_1, y_1), \dots, (x_n, y_n)$ , 把数据集分为训练样本集和测试样本集, 以便寻找最优的  $c$  值和  $q$  值.

2) 使用共轭梯度  $L_q$ -LS-SVM分类算法对训练样本集进行训练. 初始化  $w = [1 \ \dots \ 1]$ ,  $t = 1$ , 设定参数  $\tau_1 = 1e - 5$ ,  $\tau_2 = \frac{4}{9}\pi$ ,  $\tau_3 = 1e - 5$ ; 同时为方便计算设定如下参数:

$rw$ : 当  $w$  的第  $i$  个元素为 0 时,  $rw(i) = 0$ ; 当  $w$  的第  $i$  个元素不为 0 时,

$$rw(i) = c|w_i|^{q-2},$$

$$tmp1 = y \cdot X =$$

$$\left[ \sum_{i=1}^m y_i \times x_{i,1} \quad \sum_{i=1}^m y_i \times x_{i,2} \quad \dots \quad \sum_{i=1}^m y_i \times x_{i,n} \right]^T,$$

$$tmp2 = tmp1 \cdot w - 1.$$

3) 选择不同的正则化参数  $c, q$  进行实验, 在笔者的实验中, 迭代次数选择 500 次, 在  $t = 1$  时计算  $g^{(1)}$ , 共轭梯度  $d^{(1)} = -g^{(1)}$ . 当  $t > 1$  时判断第  $k$  次迭代的目标函数值的变化量, 若  $|f^{(k)} - f^{(k-1)}| < \tau_1$  即当目标函数的变化量小于设定值  $\tau_1$  停止计算, 否则梯度通过  $g^{(k)} = tmp1^T \cdot tmp2 + rw \cdot w^{(k)}$  公式计算, 共轭梯度方向  $d^{(k)}$  按照共轭梯度公式  $d^{(k)} = -g^{(k)} + \beta_k \cdot d^{(k-1)}$  计算, 其中  $\beta_k = \frac{\langle g^{(k)}, g^{(k)} \rangle}{\langle g^{(k-1)}, g^{(k-1)} \rangle}$ . 为保证下降方向是收敛的, 共轭梯度方向和梯度方向之间的夹角要小于设定角度  $\tau_2$ , 即  $\angle(g^{(k)}, d^{(k)}) < \tau_2$ , 本文使用  $d^{(k)}$  作为搜索方向, 当  $\angle(g^{(k)}, d^{(k)}) \geq \tau_2$  时, 使用  $d^{(k)} = -g^{(k)}$  为下降方向;

4) 更新权值  $w^{(k+1)} = w^{(k)} + \lambda d^{(k)}$ , 并且对  $w^{(k)}$  进行稀疏化. 算法的评判标准选定为  $w$  的某一维特征  $w_j$  小于  $w$  的无穷大范数  $\|w\|_\infty$  的万分之一, 就把对应的特征向量置为 0.

5) 在测试样本集进行实验, 得到预测误差率. 通

过选取最优的正则化参数使算法分类准确度达到最优值. 提出的算法的伪代码如算法 1 所示.

**算法 1** 迭代再权共轭梯度  $L_q$ -LS-SVM 分类算法.

### Procedure Iter-Reweig-ConjG

**Input:** pairs of samples  $(x_1, y_1), \dots, (x_n, y_n)$

**Output:**  $w$

Initialization: set  $c = (\text{ones}(1, 18)) * 2)^{(-4:13)}$ ,  $q = 0.2 : 0.2 : 4$

$w = [1, \dots, 1]$ ,  $t = 1$ ,  $\tau_1 = 1e - 5$ ,  $\tau_2 = \frac{4}{9}\pi$ ,  $\tau_3 = 1e - 5$

**for**  $j=1$  **to** length(c) **do**

**for**  $i=1$  **to** length(q) **do**

**for**  $t=1$  **to** 500 **do**

**if**  $t=1$  **then**

$$g^{(k)} = tmp1^T \cdot tmp2 + rw \cdot w^{(k)}$$

$$d^{(1)} = -g^{(1)}$$

**else if**  $|f^{(t)} - f^{(t-1)}| > \tau_1$  **then**

$$g^{(t)} = tmp1^T \cdot tmp2 + rw \cdot w^{(t)}$$

$$d^{(t)} = -g^{(t)} + \beta_t \cdot d^{(t-1)}$$

$$\beta_t = \frac{\langle g^{(t)}, g^{(t)} \rangle}{\langle g^{(t-1)}, g^{(t-1)} \rangle}$$

computation:  $\angle(g^{(t)}, d^{(t)})$

**if**  $\angle(g^{(t)}, d^{(t)}) \geq \tau_2$  **then**

$$d^{(t)} = -g^{(t)}$$

**end if**

**end if**

**end if**

update:  $w^{t+1} := w^t + \lambda^{(t)} \cdot d^{(t)}$

select features:

**if**  $w_i^{(t)} < \tau_3 \|w^{(t)}\|_\infty$  **then**

$$w_i^{(t)} = 0$$

**end if**

**end for**

**end for**

**end for**

**return**  $w$

**end** Iter-Reweig-ConjG

## 3 算法复杂性分析(Complexity analysis)

标准的共轭梯度算法计算复杂性为  $O(m^2)$ , 其中  $m$  为样本个数. 实际程序运行表明, 只需要几次到十余次迭代过程就能基本收敛. 由于提出的算法也要交叉校验选取  $c, q$  值, 总体上, 本文的  $L_q$ -LS-SVM 分类算法的计算复杂性为  $O(m^2 \cdot s \cdot t)$ , 这里  $s$  和  $t$  为  $c, q$  参数选择, 网格划分后的网格的行数和列数, 实际上由于支持向量才参与运算, 且样例的很多维随着算法的计算过程变得稀疏, 故给出的复杂性为最坏情况的

算法复杂性.

表1给出了10个数据集上 $L_p$ -SVM运行时间, 交叉校验选取 $c, q$ 值时, 表中时间需乘以 $s$ 和 $t$ , 不过标准的 $L_2$ 范数LS-SVM的运行时间也需要乘以 $s$ .

表 1 10个数据集上 $L_p$ -SVM运行时间

Table 1 The running time results of  $L_p$ -SVM on 10 datasets

数据名称	数据维数	正则化参数 $c$	正则化阶次 $q$	$L_q$ -LS-SVM 耗时/s
膀胱癌数据	57×2215	4096	3.6	0.19
结肠癌数据	96×4026	0.25	1.0	0.13
淋巴瘤数据	62×2000	0.25	1.2	1.15
黑素瘤数据	78×3750	64	4.0	1.86
PIE_0102	340×1024	2048	1.4	2.58
Yaleb_0102	128×1024	2048	1.2	1.45
Yaleb_2627	128×1024	4096	1.2	2.05
Tr11	143×6424	32	3.2	1.87
Arcene	100×10000	128	2.4	5.32
Gisette	6000×5000	8	2.4	11.14

## 4 实验结果(Experiments)

### 4.1 实验预处理(Pretreatment)

在实验部分, 本文把提出的 $L_q$ -LS-SVM分类算法首先应用在3组癌症数据集进行了详细的实验研究. 3组数据为膀胱癌数据集(bladder)、结肠癌数据集(colon)和淋巴瘤(lymphoma)数据集. 膀胱癌数据集来自于<http://www.ihes.fr/~zinovyev/princmanif2006/>, 淋巴瘤数据集来自于<http://lmpp.nih.gov/lymphoma/data/rawdata/>, 结肠癌数据集来自于<http://perso.telecom-paristech.fr/~gfort/GLM/Programs.html>. 膀胱癌数据集(bladder)样本包含2215个基因; 结肠癌数据集(colon)样本包含2000个基因; 淋巴瘤(lymphoma)数据集包含4026个基因. 所有的样本取四分之一的样本个数为测试样本, 其余为训练样本. 膀胱癌数据集(bladder)样本个数为57个, 则膀胱癌数据集可分为训练样本数42, 测试样本个数为15; 结肠癌数据集(colon)样本个数为62个, 划分得训练样本46, 测试样本16个; 淋巴瘤数据集(lymphoma)样本个数为96个, 划分得训练样本72, 测试样本24个.

为了验证和比较 $L_q$ -LS-SVM分类算法与 $L_2$ -LS-SVM分类算法的分类性能, 最后, 选取不同领域的数据集: Tr11 文本数据 (<http://glaros.dtc.umn.edu/gkhome/cluto/cluto>)、高维数小样本数据Arcene质谱仪癌症分类数据集、Sylva森林覆盖类型数据集(<http://www.modelselect.inf.ethz.ch/>)和PIE\_32×32和Yaleb\_32×32人脸图 (<http://www.zjucadcg.cn/dengcai/Data/data.html>)对两个算法进行了实验比较.

首先对数据集进行预处理, 选取区间 $[-1, 1]$ ,  $x^i$ 为

第 $i$ 个样本, 其中 $i \in \{1, 2, \dots, m\}$ , 定义如下映射函数:

$$f(x) = \begin{cases} -1, & x < -1, \\ x, & -1 < x < 1, \\ 1, & x \geq 1. \end{cases}$$

对每个样本进行预处理. 并利用以下函数对预处理的数据进行归一化处理:

$$g(x) = \frac{f(x) + 1}{2}.$$

经过以上预处理的数据, 作为算法的输入.

### 4.2 实验总结及与 $L_2$ -LS-SVM的比较(Conclusion and comparison to $L_2$ -LS-SVM)

首先扩大 $q$ 的取值范围为 $0 < q < \infty$ 在膀胱癌数据集(bladder)、结肠癌数据集(colon)和淋巴瘤(lymphoma)数据集上进行实验分析, 观察正则化阶次 $q$ 对分类准确性的影响. 因为 $q > 4$ 后分类效果基本相同, 且最优预测误差点分布在 $0 < q < 4$ , 所以缩小 $q$ 的取值范围更好的分析 $q$ 对预测误差的影响. 对应于不同的正则化参数 $c$ 和正则化阶次 $q$ , 预测误差呈现一定的规律性, 当 $q$ 较小时, 正则化阶次 $q$ 与参数 $c$ 的指数成一定的比值性. 随着 $q$ 的增大,  $c$ 的选取对分类结果影响不再明显. 通过分析预测误差分布的盒形图, 找到最优的 $(c, q)$ 组合使得得到的分类效果最好. 表2为对各数据集最优预测分类时, 参数 $c, q$ 的取值和最优分类误差的大小. 在表2中可以看出最优分类时, 正则化阶次不一定是 $L_2, L_1, L_0$ 等整数范数.

表 2 3个癌症数据集的预测结果

Table 2 Prediction error rates of the three cancer datasets

数据名	正则化参数 $c$	正则化阶次 $q$	最小分类误差/%
膀胱癌(bladder)	4096	3.6	13.07
结肠癌(colon)	0.25	1	12.95
淋巴瘤(lymphoma)	0.25	1.2	4.72

进一步分析预测分类效果最优时, 算法对样本的特征选择情况. 因为实现特征选择的正则化参数集中在 $q = 1$ 附近, 以参数 $q \in (0, 2]$ 为范围进行分析, 每个 $q$ 对应不同的 $c$ 值, 形成的节点从中选取最优的 $(c, q)$ 对使分类效果最优, 可得到10个最优点, 画出预测误差率和选取特征维数的盒形图观察正则化阶次 $q$ 不同取值时, 对应不同参数 $c$ 时, 预测误差最小时, 算法进行特征选择的效果. 然后以实现特征选择为目标, 对应预测误差的变化. 选取最优的 $(c, q)$ 对使选取的主特征个数最少, 得到10个最优点, 也分别给出特征维数和预测误差率的盒形图. 通过调整算法的两个可调参数:

正则化参数 $c$ 和正则化阶次 $q$ 的值,可以使预测误差和特征选择个数两个性能指标达到最优.下面表3(a)为对应不同的数据集,当以选择精确分类为目标时,特征向量的选择个数最优值,表3(b)为对应不同的数据集,以样本稀疏化为目标得到的预测误差值.

表 3(a) 最优分类时最少特征选择个数

Table 3(a) Select features numbers on best classification error

数据名	最小分类误差/%	特征选择个数
膀胱癌(bladder)	13.07	558
结肠癌(colon)	12.95	553.13
淋巴瘤(lymphoma)	4.72	1401.6

表 3(b) 最优特征选择时分类误差

Table 3(b) Classification error on best feature selection

数据名	最少特征选择个数	分类误差/%
膀胱癌(bladder)	3.375	20.95
结肠癌(colon)	7	42.19
淋巴瘤(lymphoma)	10.875	7.29

为了与标准的 $L_2$ -LS-SVM对比,使用LS-SVMlab v1.7软件(<http://www.esat.kuleuven.be/sista/lssvmlab/>)与本文的 $L_q$ -LS-SVM做了比较.LS-SVM分类器使用高斯核非线性分类器,分类器性能经过10倍交叉校验,选择最优的核宽度参数和 $C$ 值,3种分类数据上通过10倍交叉校验得到的核宽度sig和正则化参数 $C$ 的值如表4(a)所示.

表4(b)为 $L_q$ -LS-SVM算法滑动选取 $q$ 取值和 $L_2$ -LS-SVM选取最优的核宽度sig和正则化参数 $C$ 情况下,在3组数据集上实验得到的最小的错误率对比.

从表4给出的 $L_2$ -LS-SVM高斯核非线性分类器和本文提出的算法性能对比可以看出,扩大 $q$ 的取值范围,滑动选取 $q$ 可以得到比 $L_2$ -LS-SVM算法更精确的分类预测结果.

表 4(a) 3个实际数据集正则化参数 $C$ 和核宽度sig的值

Table 4(a) The values of regularization parameter  $C$  and kernel width sig on the three real datasets

参数名	正则化参数 $C$	核宽度sig
膀胱癌(bladder)	1.79	2080
结肠癌(colon)	2.07	2369
淋巴瘤(lymphoma)	3.84	7325

表 4(b) 3个癌症数据集分类错误率比较

Table 4(b) Comparison of test error rates on the three cancel datasets

数据名	LS-SVM错误率/%	自动选取 $q$ 对应错误率/%
膀胱癌(bladder)	16.93	13.07
结肠癌(colon)	16.95	12.95
淋巴瘤(lymphoma)	13.79	4.72

为验证算法的分类效果,下面选取多组实验数据进行 $L_q$ -LS-SVM实验分析并且与 $L_2$ -LS-SVM做比较.采用的10组数据为:上述的3组基因数据和黑素瘤基因数据、Tr11文本数据、高维数小样本数据Arcene质谱仪癌症分类数据集、Sylva森林覆盖类型数据集、PIE\_32×32和Yaleb\_32×32人脸图.其中Yaleb\_32×32数据为人脸数据,数据类型为灰度值,取值范围0~255.该数据包含38个人脸数据,每个人有59~64张人脸照片,每张照片含1024个像素.本实验取出第1个人和第2个人的人脸数据,共128张照片,其中第1个人有64张照片,第2个人有64张照片.取标签值为1,2的数据组成新的样本数据Yaleb\_0102人脸数据集;取出第26个人和第27个人的人脸数据,共128张照片,组成样本数据Yaleb\_2627人脸数据集;同理得到样本数据PIE\_0102人脸数据集.表5分别给出使用 $L_q$ -LS-SVM时达到的最好的分类结果,列出了最优的参数选择 $C, q$ 值.同时使用 $L_2$ -LS-SVM对10组数据进行分类预测,两组算法对比分析.

表 5 10组数据集的预测结果和对比分析

Table 5 Prediction error rates of the ten datasets and compare with  $L_2$ -LS-SVM

数据名	数据维数	正则化参数 $C$	正则化阶次 $q$	$L_q$ -LS-SVM错误率/%	$L_2$ -LS-SVM错误率/%
膀胱癌数据	57×2215	4096	3.6	13.07	16.93
结肠癌数据	96×4026	0.25	1.0	12.95	16.95
淋巴瘤数据	62×2000	0.25	1.2	4.72	13.79
黑素瘤数据	78×3750	64	4.0	10.69	14.71
PIE_0102	340×1024	2048	1.4	6.86	5.56
Yaleb_0102	128×1024	2048	1.2	2.60	3.30
Yaleb_2627	128×1024	4096	1.2	6.58	5.97
Tr11	143×6424	32	3.2	17.82	21.36
Arcene	100×10000	128	2.4	6.67	8.92
Gisette	6000×5000	8	2.4	9.43	11.28

表5为10组数据重复运行100次的实验结果, 通过实验结果可以看出 $L_q$ -LS-SVM主要在高维小样本数据集上的预测结果要优于标准LS-SVM的预测, 只有在PIE\_0102和Yaleb\_2627人脸数据上 $L_q$ -LS-SVM比 $L_2$ -LS-SVM稍差. 证明提出的 $L_q$ -LS-SVM具有很好的分类预测性.

## 5 结论与展望(Conclusion and prospect)

本文提出了 $L_q$ -LS-SVM分类器算法设计模式, 与正则化的阶次为整数 $L_2, L_1, L_0$ 范数不同, 使用网格法扩大 $q$ 的取值范围, 使分类器算法在不同的正则化范数的阶次 $q$ 下,  $0 < q < \infty$ 进行分类, 通过调整正则化参数 $c, q$ 的值, 使分类准确性得到提高. 同时该算法在特征选择方面取得了进展, 实现了模型稀疏化. 在实验部分, 选取不同领域的数据集, 包括癌症微阵列基因分类数据、Tr11文本数据、Arcene质谱仪癌症分类数据集、Sylvia森林覆盖类型数据集、PIE\_32 $\times$ 32和Yaleb\_32 $\times$ 32人脸图, 进行分类和特征选择性能评价, 实验结果表明提出的分类算法能够同时实现了最优分类和特征选择, 相比 $L_2$ -LS-SVM算法分类准确性得到了提高.

本文的工作, 主要集中于线性分类器的设计, 如何扩展提出的算法应用于核化LS-SVM, 是作者下一步要做的工作.

## 参考文献(References):

- [1] 刘建伟, 李双成, 罗雄麟.  $p$ 范数正则化支持向量机分类算法 [J]. 自动化学报, 2012, 38(1): 76 – 87.  
(LIU Jianwei, LI Shuangcheng, LUO Xionglin. Classification algorithm of support vector machine via  $p$ -norm regularization [J]. *Acta Automatica Sinica*, 2012, 38(1): 76 – 87.)
- [2] SUYKENS J A K, VAN GESTEL T, DE BRABANTER K, et al. *Least Squares Support Vector Machines* [M]. Singapore: World Scientific, 2002.
- [3] SUYKENS J A K, VANDEWALLE J. Least squares support vector machine classifiers [J]. *Neural Processing Letters*, 1999, 9(3): 293 – 300.
- [4] LOPEZ J, DORRONSORO J R. Least 1-norm SVMs: a new SVM variant between standard and LS-SVMs [C] // *Proceedings of the 18th European Symposium on Artificial Neural Networks*. Bruges, Belgium, 2010, 135 – 140.
- [5] LOPEZ J, DE BRABANTER J, SUYKENS J A K, et al. Sparse  $L_s$ -SVMs with  $L_0$ -norm minimization [C] // *Proceedings of the 19th European Symposium on Artificial Neural Networks*. Bruges, Belgium: Elsevier, 2011, 189 – 194.
- [6] TAN J Y, ZHANG C H, DENG N Y. Cancer related gene identification via  $p$ -norm support vector machine [C] // *Proceedings of the Fourth International Conference on Computational Systems Biology*. Suzhou, China: World Publishing Corporation, 2010: 101 – 108.
- [7] BRADLEY P S, MANGASARIAN O L, STREET W N. Feature selection via mathematical programming [J]. *INFORMS Journal on Computing*, 1998, 10(2): 209 – 217.
- [8] SUYKENS J A K, DE BRABANTER J, LUKAS L, et al. Weighted least squares support vector machines: robustness and sparse approximation [J]. *Neurocomputing*, 2002, 48(1): 85 – 105.
- [9] VALYON J, HORVATH G. A weighted generalized LS-SVM [J]. *Electrical Engineering and Computer Science*, 2003, 47(3/4): 229 – 251.
- [10] ZHOU L, LAI K K, YEN J. Credit scoring models with AUC maximization based on weighted SVM [J]. *International Journal of Information Technology & Decision Making*, 2009, 8(4): 677 – 696.
- [11] LESKI J M. Iteratively reweighted least squares classifier and its  $L_2$ - and  $L_1$ -regularized Kernel versions [J]. *Bulletin of The Polish Academy of Sciences: Technical Sciences*, 2010, 58(1): 171 – 182.
- [12] SUN D, LI J, WEI L. Credit risk evaluation: Support vector machines with adaptive  $L_q$  penalty [J]. *Journal of Southeast University (English Edition)*, 2008, 24(S): 3 – 6.
- [13] LIU J, LI J, XU W, et al. Weighted  $L_q$  adaptive least squares support vector machine classifiers-robust and sparse approximation [J]. *Expert Systems with Applications*, 2011, 38(3): 2253 – 2259.
- [14] BRITO A E. *Iterative adaptive extrapolation applied to SAR image formation and sinusoidal recovery* [D]. EI Paso: The University of Texas, 2001.
- [15] KLOFT M, BREFELD U, SONNENBURG S, et al. Efficient and accurate  $L_p$ -norm multiple kernel learning [J]. *Advances in Neural Information Processing Systems*, 2009, 22(22): 997 – 1005.
- [16] ZHU J, ROSSET S, HASTIE T, et al. 1-norm support vector machines [J]. *Advances in Neural Information Processing Systems*, 2004, 16(1): 49 – 56.
- [17] LIU Y, WU Y. Variable selection via a combination of the  $L_0$  and  $L_1$  penalties [J]. *Journal of Computational and Graphical Statistics*, 2007, 16(4): 782 – 798.
- [18] LIU Y, HELEN ZHANG H, PARK C, et al. Support vector machines with adaptive  $L_q$  penalties [J]. *Computational Statistics & Data Analysis*, 2007, 51(12): 6380 – 6394.
- [19] MARTIN S. Training support vector machines using Gilbert's algorithm [C] // *Proceedings of the Fifth IEEE International Conference on Data Mining*. Houston, USA: IEEE, 2005: 306 – 313.
- [20] LEE S I, LEE H, ABBEEL P, et al. Efficient  $L_1$  regularized logistic regression [C] // *Proceedings of the 21st National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 2006: 401 – 408.
- [21] LIU Z, JIANG F, TIAN G, et al. Sparse logistic regression with  $L_p$  penalty for biomarker identification [J]. *Statistical Applications in Genetics and Molecular Biology*, 2007, 6(1): 1 – 22.
- [22] BRADLEY P S, MANGASARIAN O L. Feature selection via concave minimization and support vector machines [C] // *Proceedings of the 15th International Conference on Machine Learning*. Madison, USA: Morgan Kaufmann, 1998, 82 – 90.
- [23] FOUCAIT S, LAI M J. Sparsest solutions of underdetermined linear system via  $L_q$ -minimization for  $0 < q < 1$  [J]. *Applied and Computational Harmonic Analysis*, 2009, 26(3): 395 – 407.
- [24] FAN J, LI R. Variable selection via nonconcave penalized likelihood and its oracle properties [J]. *Journal of the American Statistical Association*, 2001, 96(456): 1348 – 1360.
- [25] PYTLAK R. *Conjugate Gradient Algorithms in Nonconvex Optimization* [M]. New York: Springer, 2009.



- [26] SCHIROTZEK W. *Nonsmooth Analysis* [M]. New York: Springer, 2007.
- [27] MISHRA S K, GIORGI G. *Invexity and Optimization* [M]. New York: Springer, 2008.
- [28] ZHOU T, TAO D, WU X. NESVM: a fast gradient method for support vector machines [C] // *Proceedings of the Tenth IEEE International Conference on Data Mining*. Sydney, Australia: IEEE, 2010: 679 – 688.
- [29] JOACHIMS T. Training linear svms in linear time [C] // *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, USA: ACM, 2006: 217 – 226.
- [30] SHALEV-SHWARTZ S, SINGER Y, SREBRO N, et al. Pegasos: primal estimated sub-gradient solver for svm [J]. *Mathematical Programming*, 2011, 127(1): 3 – 30.
- [31] JOACHIMS T. Making large-scale SVM learning practical [C] // *Proceedings of Advances in Kernel Methods-Support Vector Machines*. Cambridge, USA: MIT Press, 1999: 169 – 184.
- [32] CHANG C C, LIN C J. LIBSVM: a Library for support vector machines [J]. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011, 2(3): 27.
- [33] SINDHWANI V, KEERTHI S S. Large scale semi-supervised linear SVMs [C] // *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. New York: ACM, 2006: 477 – 484.
- [34] SONNENBURG S, FRANCO V. COFFIN: a computational framework for linear SVMs [C] // *Proceedings of the 27nd International Conference on Machine Learning*. Haifa, Israel: Omnipress, 2010: 999 – 1006.
- [35] LIU J, LI S C, LUO X. Iterative reweighted noninteger norm regularizing SVM for gene expression data classification [J/DB]. *Computational and Mathematical Methods in Medicine*, 2013. ID 768404. <http://dx.doi.org/10.1155/2013/768404>.
- [36] CHEN P C, HUANG S Y, CHEN W J, et al. A new regularized least squares support vector regression for gene selection [J]. *BMC Bioinformatics*, 2009, 10(1): 10 – 44.
- [37] BERTSEKAS D P. *Nonlinear Programming (Second edition)* [M]. Belmont, England: Athena Scientific, 1999.
- [38] 袁亚湘, 孙文瑜. 最优化理论和方法 [M]. 北京: 科学出版社, 1999. (YUAN Yaxiang, SUN Wenyu. *Optimization Theory and Method* [M]. Beijing: Science Press, 1999.)

### 作者简介:

刘建伟 (1966–), 男, 副研究员, 硕士生导师, 目前研究方向为机器学习、智能信息处理、复杂系统分析、预测与控制等, E-mail: liujw@cup.edu.cn;

黎海恩 (1988–), 女, 硕士研究生, 目前研究方向为机器学习、概率图模型等, E-mail: lihaien1988@163.com;

刘媛 (1989–), 女, 硕士研究生, 目前研究方向为机器学习等, E-mail: ysht.25@sina.com;

付捷 (1988–), 女, 硕士研究生, 目前研究方向为机器学习等, E-mail: liuyliujw@sina.com;

罗雄麟 (1963–), 男, 教授, 博士生导师, 目前研究方向为智能控制、非线性控制等, E-mail: luoxl@cup.edu.cn.