

基于改进 SVM 的城市道路短时交通状态预测

闫 贺¹, 朱 丽^{2,3}, 戚 湧¹

(1.南京理工大学计算机科学与工程学院,江苏 南京 210094)

(2.东南大学交通学院,江苏 南京 210096)

(3.苏交科集团股份有限公司规划研究所,江苏 南京,210019)

[摘要] 为提高短时交通状态预测的精度,使交通管理者更有效地进行交通规划和管理,本文把基于 L_1 范数距离度量的最小二乘孪生有界支持向量机(twin bounded support vector machine, TBSVM)扩展成多分类算法用于短时交通状态预测,简称 MLSTBSVM_{L1}. 在实验数据上对 MLSTBSVM_{L1} 算法的有效性进行验证,实验结果表明,相比于其他预测算法,提出的 MLSTBSVM_{L1} 算法在预测精度上有较大提升.

[关键词] 短时交通状态预测,机器学习,MLSTBSVM_{L1} 算法,TBSVM 算法

[中图分类号] TP391 [文献标志码] A [文章编号] 1001-4616(2019)03-0129-09

Short-term Traffic Condition Prediction of Urban Road Based on Improved SVM

Yan He¹, Zhu Li^{2,3}, Qi Yong¹

(1.School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

(2.School of Transportation, Southeast University, Nanjing 210096, China)

(3.JSTI Group Co., Ltd, Planning Institute, Nanjing 210019, China)

Abstract: To improve the accuracy of short-term traffic condition prediction and make planning and management more effective for traffic managers, the least squares Twin Bounded Support Vector Machine (TBSVM) based on L_1 -norm distance is extended to a new algorithm (MLSTBSVM_{L1}) which could solve multi-classification problems. The validity of the proposed MLSTBSVM_{L1} is verified through experiments and the results demonstrate that the MLSTBSVM_{L1} algorithm has significant improvement in prediction accuracy compared with other prediction algorithms.

Key words: short-term traffic condition prediction, machine learning, MLSTBSVM_{L1} algorithm, TBSVM algorithm

随着社会快速发展,有限的道路资源与急剧增长的机动车数量之间的供需矛盾变得越来越尖锐,导致交通供求之间难以平衡^[1],这使得各城市中的交通拥堵问题日益严峻. 道路交通控制与诱导系统是应对城市交通问题的重要方法,短时交通状态预测是实现上述方法的关键问题之一,如何精确预测短时交通状态一直是城市智能交通重点研究的问题.

目前,许多学者已提出多种类型的短时交通流量预测模型和方法. 姚智胜^[2]使用支持向量机回归(support vector machine regression, SVMR)算法建立模型,采用遗传算法对 SVMR 参数进行优化,提出基于 SVMR 的区域路网断面交通流的短时预测方法. 赵亚萍等^[3]在分析交通量影响因素的基础上,提出基于最小二乘 SVM 的交通流量预测模型. 欧阳俊^[4]利用多核混合 SVM 识别和处理不同类别数据的能力,提出基于多核混合 SVM 的城市短时交通预测方法. 该方法以统计分析交通状态等级数据样本为前提,并继承了 SVM 的优点,采用改进的粒子群算法对 SVM 的参数进行优化选择. 为提高交通流预测的精度和速度,孙占全等^[5]提出用 SVM 与主成分分析(principal component analysis, PCA)相结合的交通流预测方法. PCA 用于对预测变量的特征抽取,用主要特征代替原预测变量,并将选取的主要特征代入非线性 SVM,进行交

收稿日期:2018-04-14.

基金项目:国家重点研究计划政府间国际科技创新合作重点专项(2016YFE0108000)、国家重点自然科学基金项目(51238008)、国家自然科学基金(61272419,61772273)、江苏省自然科学基金(BK20141403)、2018 江苏省普通高校学术学位研究生科研创新计划项目(KYCX18_0424).

通讯联系人:戚湧,教授,博士生导师,研究方向:机器学习,数据挖掘,网络信息安全,社会计算,智能交通系统. E-mail:qyong@njnu.edu.cn

通流预测. 针对短时交通流变化周期性与随机性的特点,樊娜等^[6]结合非参数回归模型和 BP 神经网络模型,提出一种新的混合模型对短时交通流进行预测. 该模型通过模糊控制算法确定各单项模型的权重,并按照不同权重进行组合. 刘燕^[7]根据交通流数据具有的随机波动性,提出基于小波分析的短时交通流组合预测方法. 该方法通过小波分析理论将短时交通流序列分解为稳定部分和随机序列,分别运用径向基神经网络和马尔科夫链对稳定和随机两部分序列进行预测,并将预测结果组合作为最终的预测值.

SVM 具有完整的理论基础和出色的泛化能力,使用核函数可解决样本维数问题^[8-9]. 基于经验风险最小化原则的神经网络,因其结构选择缺乏理论指导,预测结果容易陷入局部极值点,且需要大量训练数据来生成预测模型,故神经网络的鲁棒性受到限制. 鉴于此,本文把基于 L_1 范数距离度量的最小二乘孪生有界支持向量机^[10] (L_1 -LSTBSVM) 扩展成多分类算法用于城市道路短时交通状态的预测,简称 MLSTBSVM_{L1}. L_1 范数距离的应用有利于提高模型的鲁棒性^[10-13],故 MLSTBSVM_{L1} 的使用有利于实时预报城市道路交通运行状态,为交通管理者有效地进行交通规划和管理提供强有力的技术支撑.

1 相关工作

1994 年,美国交通运输研究会公布了公路通行能力指南,在专题报道 209 中指出,根据道路中各路段的平均速度将道路交通等级划分为六级(A~F),每个字母代表一级交通状态,A 级代表最畅通的道路状态,F 级则表示最拥堵的道路状态,见表 1.

交通状态预测旨在探究未来时刻的交通运行状态,使城市道路高效的运行,为交通参与者提供高品质的交通服务. 根据公众对交通状态的认知,可将交通状态预测分为两类:(1) 根据交通状态的等级或类别进行预测;(2) 根据交通状态参数(交通量,占有率等)进行预测. 在实际应用中,根据交通状态等级或类别预测更容易被交通参与者理解和接受,故本文采用第一种预测方式.

交通状态预测可依据预测时间的长短分为 3 类:短时预测(5~30 min 的预测时间),中时预测(30 min 至数小时的预测时间),长时预测(以时间间隔不低于 24 h 的预测时间). 交通等级 A 到 F 用 1 到 6,6 个数字表示,每个数字代表一级交通状态,见表 1.

表 1 城市道路交通状态等级划分
Table 1 The classification of urban road traffic condition

运行等级	快速路	主干路	次干路
A 级	$V>56.3$	$V>48.3$	$V>40.2$
B 级	$45.1<V\leq 56.3$	$38.6<V\leq 48.3$	$30.6<V\leq 40.2$
C 级	$35.4<V\leq 45.1$	$28.9<V\leq 38.6$	$20.9<V\leq 30.6$
D 级	$27.4<V\leq 35.4$	$22.5<V\leq 28.9$	$14.5<V\leq 20.9$
E 级	$20.9<V\leq 27.4$	$16.1<V\leq 22.5$	$11.3<V\leq 14.5$
F 级	$V\leq 20.9$	$V\leq 16.1$	$V\leq 11.3$

其中, V 表示平均速度,单位为 km/h.

本文涉及的向量均为列向量,用小写字母表示,列向量右上方 T 符号表示转换成行向量,矩阵用大写加粗字母表示. 向量 e_1 和 e_2 表示相应维度的单位列向量, I 表示相应维度的单位矩阵. 我们用矩阵 $A = (A_1^{(1)}, A_2^{(1)}, \dots, A_{m_1}^{(1)})^T$ 表示正类样本,剩余为负类样本,即用矩阵 $B = (B_1^{(2)}, B_2^{(2)}, \dots, B_{m_2}^{(2)})^T$ 表示,其样本大小分别为 $m_1 \times n$ 和 $m_2 \times n$,且 $m_1 + m_2 = m, m, n$ 分别表示样本个数与维度. 简单回顾两个经典的分类算法:基于广义特征值的最接近支持向量机 (proximal support vector machine via generalized eigenvalues, GEPSVM)^[14] 和孪生有界支持向量机 (twin bounded support vector machine, TBSVM)^[15].

1.1 基于广义特征值的最接近支持向量机 (GEPSVM)

GEPSVM 通过求解两个广义特征值问题来获得两个非平行最优分割平面,如下

$$x^T w_1 + b_1 = 0, \quad x^T w_2 + b_2 = 0, \tag{1}$$

式中, $w_1, w_2 \in R^n$ 是权向量, $b_1, b_2 \in R$ 是偏差,式(1)的目标是最小化正类样本点到这两个平面的欧几里得距离. GEPSVM 的目标函数如下:

$$\min_{(w_1, b_1) \neq 0} \frac{\|Aw_1 + e_1 b_1\|_2^2 / \|(w_1 \quad b_1)^T\|_2^2}{\|Bw_1 + e_2 b_1\|_2^2 / \|(w_1 \quad b_1)^T\|_2^2}, \tag{2}$$

$$\min_{(\mathbf{w}_2, b_2) \neq 0} \frac{\|\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2\|_2^2 / \|\mathbf{w}_2 \quad b_2\|_2^T}{\|\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2\|_2^2 / \|\mathbf{w}_2 \quad b_2\|_2^T}, \quad (3)$$

式中, $\|\cdot\|_2$ 表示 L_2 范数距离, 且 $\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1 \neq 0$ 和 $\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2 \neq 0$, 重写式(2)和(3)如下:

$$\min_{\mathbf{w}_1, b_1} \frac{\|\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1\|_2^2}{\|\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1\|_2^2}, \quad (4)$$

$$\min_{\mathbf{w}_2, b_2} \frac{\|\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2\|_2^2}{\|\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2\|_2^2}. \quad (5)$$

在求解方程组时, 半正定矩阵可能会参与计算, 为避免奇异问题, 式(4)和(5)分别引入正则项, 正则项的引入使 GEPSVM 失去了原有的几何意义.

$$\min_{\mathbf{w}_1, b_1} \frac{\|\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1\|_2^2 + \delta \|\mathbf{w}_1 \quad b_1\|_2^T}{\|\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1\|_2^2}, \quad (6)$$

$$\min_{\mathbf{w}_2, b_2} \frac{\|\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2\|_2^2 + \delta \|\mathbf{w}_2 \quad b_2\|_2^T}{\|\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2\|_2^2}, \quad (7)$$

式中, $\delta \|\mathbf{w}_1 \quad b_1\|_2^T$ 和 $\delta \|\mathbf{w}_2 \quad b_2\|_2^T$ 是正则项, δ 是正则化因子. 正则项的引入可提高 GEPSVM 的稳定性和分类精度. 此外, GEPSVM 的优化目标是确保每个平面离两类样本中的一类尽可能的近, 同时离另一类尽可能的远. 式(6)和(7)等价于:

$$\min_{z_1} \frac{\mathbf{z}_1^T \mathbf{E} \mathbf{z}_1}{\mathbf{z}_1^T \mathbf{F} \mathbf{z}_1}, \quad (8)$$

$$\min_{z_2} \frac{\mathbf{z}_2^T \mathbf{L} \mathbf{z}_2}{\mathbf{z}_2^T \mathbf{M} \mathbf{z}_2}, \quad (9)$$

式中, 矩阵 $\mathbf{H} = (\mathbf{A} \quad \mathbf{e}_1)$, $\mathbf{G} = (\mathbf{B} \quad \mathbf{e}_2)$, $\mathbf{E} = \mathbf{H}^T \mathbf{H} + \delta \mathbf{I}$, $\mathbf{F} = \mathbf{G}^T \mathbf{G}$, $\mathbf{L} = \mathbf{G}^T \mathbf{G} + \delta \mathbf{I}$, $\mathbf{M} = \mathbf{H}^T \mathbf{H}$, 增广向量 $\mathbf{z}_1 = (\mathbf{w}_1 \quad b_1)^T$, $\mathbf{z}_2 = (\mathbf{w}_2 \quad b_2)^T$. 式(8)和(9)是瑞利商(Rayleigh Quotient)问题, 通过求解广义特征值即可获得它们的解.

$$\mathbf{E} \mathbf{z}_1 = \lambda_1 \mathbf{F} \mathbf{z}_1, \quad \mathbf{z}_1 \neq 0 \quad (10)$$

$$\mathbf{L} \mathbf{z}_2 = \lambda_2 \mathbf{M} \mathbf{z}_2, \quad \mathbf{z}_2 \neq 0 \quad (11)$$

式(8)获得最小值时的特征向量对应于式(10)的最小特征值 λ_1 , 因此, 增广向量 \mathbf{z}_1 表示对应于 λ_1 的特征向量, 即 $\mathbf{z}_1 = (\mathbf{w}_1 \quad b_1)^T$. 其前 d 个分量表示第一个分类平面的权重向量 \mathbf{w}_1 , 最后一个分量为偏差 b_1 , 故 $\mathbf{x}^T \mathbf{w}_1 + b_1 = 0$ 分类平面取决于增广向量 \mathbf{z}_1 的取值. 同理可得, $\mathbf{x}^T \mathbf{w}_2 + b_2 = 0$ 取决于增广向量 $\mathbf{z}_2 = (\mathbf{w}_2 \quad b_2)^T$ 的取值.

1.2 孪生有界支持向量机(TBSVM)

与 GEPSVM 不同, TBSVM^[15]需求解两个凸二次规划问题(Quadratic Programming Problems, QPPs)来获得两个非平行的最优分类面, 为实现结构风险最小化, TBSVM 通过最大化间隔的思想引入正则项, 其目标函数如下

$$\min_{\mathbf{w}_1, b_1, q_2} \frac{1}{2} c_3 (\|\mathbf{w}_1\|_2^2 + b_1^2) + \frac{1}{2} \|\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1\|_2^2 + c_1 \mathbf{e}_2^T \mathbf{q}_2, \quad (12)$$

$$\text{s.t. } -(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 \geq \mathbf{e}_2, \quad \mathbf{q}_2 \geq 0,$$

$$\min_{\mathbf{w}_2, b_2, q_1} \frac{1}{2} c_4 (\|\mathbf{w}_2\|_2^2 + b_2^2) + \frac{1}{2} \|\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2\|_2^2 + c_2 \mathbf{e}_1^T \mathbf{q}_1, \quad (13)$$

$$\text{s.t. } (\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2) + \mathbf{q}_1 \geq \mathbf{e}_1, \quad \mathbf{q}_1 \geq 0.$$

式中, $\|\cdot\|_2$ 表示 L_2 范数距离, $(1/2) c_3 (\|\mathbf{w}_1\|_2^2 + b_1^2)$ 是正则项, 此项的引入保证结构风险最小化. TBSVM 的原始目标函数中涉及不等式约束, 需求解两个 QPPs 获得最优分类面. 当训练样本较大, 计算两个 QPPs 导致较大的计算开销, 这不利于对大样本数据的处理. 构建式(12)的拉格朗日函数如下:

$$L(\mathbf{w}_1, b_1, q_2, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} c_3 (\|\mathbf{w}_1\|_2^2 + b_1^2) + \frac{1}{2} \|\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1\|_2^2 + c_1 \mathbf{e}_2^T \mathbf{q}_2 - \boldsymbol{\alpha}^T (-(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 - \mathbf{e}_2) - \boldsymbol{\beta}^T \mathbf{q}_2, \quad (14)$$

式中, $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$ 是拉格朗日乘子. 分别求解拉格朗日函数 $L(\mathbf{w}_1, b_1, q_2, \boldsymbol{\alpha}, \boldsymbol{\beta})$ 关于 \mathbf{w}_1, b_1 和 q_2 的偏导数, 并令

它们的偏导数等于 0, 可得 (Karush-Kuhn-Tucker, KKT) 必要条件.

$$\frac{\partial L}{\partial \mathbf{w}_1} = c_3 \mathbf{w}_1 + \mathbf{A}^T (\mathbf{A} \mathbf{w}_1 + \mathbf{e}_1 b_1) + \mathbf{B}^T \boldsymbol{\alpha} = 0, \quad (15)$$

$$\frac{\partial L}{\partial b_1} = c_3 b_1 + \mathbf{e}_1^T (\mathbf{A} \mathbf{w}_1 + \mathbf{e}_1 b_1) + \mathbf{e}_2^T \boldsymbol{\alpha} = 0, \quad (16)$$

$$\frac{\partial L}{\partial \mathbf{q}_2} = c_1 \mathbf{e}_2 - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0, \quad (17)$$

$$-(\mathbf{B} \mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 \geq \mathbf{e}_2, \quad \mathbf{q}_2 \geq 0, \quad (18)$$

$$\boldsymbol{\alpha}^T (-(\mathbf{B} \mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 - \mathbf{e}_2) = 0, \quad \boldsymbol{\beta}^T \mathbf{q}_2 = 0. \quad (19)$$

结合式(15)和(16), 可得增广向量 \mathbf{z}_1 .

$$\mathbf{z}_1 = (\mathbf{w}_1 \quad b_1)^T = -(\mathbf{H}^T \mathbf{H} + c_3 \mathbf{I})^{-1} \mathbf{G}^T \boldsymbol{\alpha}, \quad (20)$$

式中, 令 $\mathbf{H} = (\mathbf{A} \quad \mathbf{e}_1)$, $\mathbf{G} = (\mathbf{B} \quad \mathbf{e}_2)$. 把式(20)代入拉格朗日函数并使用 KKT 条件, 得到式(12)的对偶问题, 如下:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}_2^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{G} (\mathbf{H}^T \mathbf{H} + c_3 \mathbf{I})^{-1} \mathbf{G}^T \boldsymbol{\alpha}, \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha} \leq c_1 \mathbf{e}_2. \end{aligned} \quad (21)$$

解出 $\boldsymbol{\alpha}$ 后, 代入式(20)即可. 用极其相似的方法, 可求得增广向量 \mathbf{z}_2 .

$$\mathbf{z}_2 = (\mathbf{w}_2 \quad b_2)^T = (\mathbf{G}^T \mathbf{G} + c_4 \mathbf{I})^{-1} \mathbf{H}^T \boldsymbol{\beta}. \quad (22)$$

通过增广向量 \mathbf{z}_1 和 \mathbf{z}_2 , 可得到两个非平行分类面(式(1)).

2 多分类的交通状态预测模型

SVM 是一种基于 Vapnik-Chervonenkis (VC) 维和统计学习理论的有监督的机器学习方法. 对于二分类问题, SVM 是一个完善并且有效的分类方法. 但在现实生活中, 多分类问题更有待于被有效的解决, 因此, 许多研究者成功地将 SVM 及其改进版本扩展成多分类问题^[16-20]. 迄今为止, SVM 已被广泛应用于各种分类和回归问题中^[21-24]. 对于多分类的 MLSTBSVM_{L1} 算法而言, 它构建了 N 个二分类的 LSTBSVM_{L1} 分类器. 其中, 第 m ($m \in 1, 2, \dots, N$) 个 LSTBSVM_{L1} 分类器把第 m 类视为正类样本, 同时把其他类看作负类样本. MLSTBSVM_{L1} 求解 N 个线性方程组并生成 N 个分类平面, 每一类样本对应一个平面. MLSTBSVM_{L1} 的目标函数构建如下:

$$\begin{aligned} \min_{\mathbf{w}_m, b_m, \mathbf{q}_m} \quad & \frac{1}{2} c_{2m} (\|\mathbf{w}_m\|_2^2 + b_m^2) + \frac{1}{2} \|\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m\|_1 + \frac{1}{2} c_{1m} \|\mathbf{q}_m\|_1, \\ \text{s.t.} \quad & -(\mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m) + \mathbf{q}_m = \mathbf{e}_{2m}. \end{aligned} \quad (23)$$

式中, $\|\cdot\|_1$ 表示 L_1 范数距离, c_{1m} 和 c_{2m} 是惩罚系数, 可以克服 MLSTBSVM_{L1} 算法中样本不平衡的问题. \mathbf{e}_{1m} 和 \mathbf{e}_{2m} 是相应维度的单位列向量, \mathbf{q}_m 是松弛变量, 第 m 类样本和剩余的 $N-1$ 类样本分别由矩阵 \mathbf{A}_m 和 \mathbf{B}_m 组成. 式(23)的优化目标是第 m 类的样本距其相应的平面尽可能的近, 同时离其它类的平面尽可能的远, 重写式(23), 如下

$$\begin{aligned} \min_{\mathbf{w}_m, b_m, \mathbf{q}_m} \quad & \frac{c_{2m}}{2} (\|\mathbf{w}_m\|_2^2 + b_m^2) + \frac{1}{2} (\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m)^T \mathbf{D}_{1m} (\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m) + \frac{c_{1m}}{2} (\mathbf{e}_{2m} + \mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m)^T \mathbf{D}_{2m} (\mathbf{e}_{2m} + \mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m) \\ \text{s.t.} \quad & -(\mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m) + \mathbf{q}_m = \mathbf{e}_{2m}, \end{aligned} \quad (24)$$

式中, $\mathbf{D}_{1m} = \frac{1}{|\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m|}$, $\mathbf{D}_{2m} = \frac{1}{|\mathbf{e}_{2m} + \mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m|}$. 式(24)中包含绝对值操作, 不易于直接求解, 为解决此类问题, 本文设计一个迭代优化算法, 其基本思想是迭代更新目标函数值直到收敛到一个固定值 (0.001), 见算法 1. 分别求解式(24)关于 \mathbf{w}_m 和 b_m 的偏导数, 并令它们的偏导数等于 0.

$$c_{2m} \mathbf{w}_m + \mathbf{A}_m^T \mathbf{D}_{1m} (\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m) + c_{1m} \mathbf{B}_m^T \mathbf{D}_{2m} (\mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m + \mathbf{e}_{2m}) = 0, \quad (25)$$

$$c_{2m} b_m + \mathbf{e}_{1m}^T \mathbf{D}_{1m} (\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m) + c_{1m} \mathbf{e}_{2m}^T \mathbf{D}_{2m} (\mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m + \mathbf{e}_{2m}) = 0. \quad (26)$$

联合式(25)和(26),

$$\frac{c_{2m}}{c_{1m}} \begin{pmatrix} \mathbf{I}(\mathbf{w}_m) \\ b_m \end{pmatrix} + \frac{1}{c_{1m}} \begin{pmatrix} \mathbf{A}_m^T \mathbf{D}_{1m} \mathbf{A}_m & \mathbf{A}_m^T \mathbf{D}_{1m} \mathbf{e}_{1m} \\ \mathbf{e}_{1m}^T \mathbf{D}_{1m} \mathbf{A}_m & m_{1m} \mathbf{D}_{1m} \end{pmatrix} \begin{pmatrix} \mathbf{w}_m \\ b_m \end{pmatrix} + \begin{pmatrix} \mathbf{B}_m^T \mathbf{D}_{2m} \mathbf{B}_m & \mathbf{B}_m^T \mathbf{D}_{2m} \mathbf{e}_{2m} \\ \mathbf{e}_{2m}^T \mathbf{D}_{2m} \mathbf{B}_m & m_{2m} \mathbf{D}_{2m} \end{pmatrix} \begin{pmatrix} \mathbf{w}_m \\ b_m \end{pmatrix} + \begin{pmatrix} \mathbf{B}_m^T \mathbf{D}_{2m} \mathbf{e}_{2m} \\ m_{2m} \mathbf{D}_{2m} \end{pmatrix} = 0, \quad (27)$$

式(27)等价于式(28),

$$\frac{c_{2m}}{c_{1m}} \begin{pmatrix} \mathbf{I}(\mathbf{w}_m) \\ b_m \end{pmatrix} + \frac{1}{c_{1m}} \begin{pmatrix} \mathbf{A}_m^T \\ \mathbf{e}_{1m}^T \end{pmatrix} \mathbf{D}_{1m} (\mathbf{A}_m \quad \mathbf{e}_{1m}) \begin{pmatrix} \mathbf{w}_m \\ b_m \end{pmatrix} + \begin{pmatrix} \mathbf{B}_m^T \\ \mathbf{e}_{2m}^T \end{pmatrix} \mathbf{D}_{2m} (\mathbf{B}_m \quad \mathbf{e}_{2m}) \begin{pmatrix} \mathbf{w}_m \\ b_m \end{pmatrix} = - \begin{pmatrix} \mathbf{B}_m^T \\ \mathbf{e}_{2m}^T \end{pmatrix} \mathbf{D}_{2m} \mathbf{e}_{2m}. \quad (28)$$

已定义 $\mathbf{z}_m = (\mathbf{w}_m \quad b_m)^T$, $\mathbf{H}_m = (\mathbf{A}_m \quad \mathbf{e}_{1m})$, $\mathbf{G}_m = (\mathbf{B}_m \quad \mathbf{e}_{2m})$, 使用这些符号, 重写式(28),

$$\mathbf{z}_m = (\mathbf{w}_m \quad b_m)^T = -((1/c_{1m})(\mathbf{H}_m^T \mathbf{D}_{1m} \mathbf{H}_m + c_{2m} \mathbf{I}) + \mathbf{G}_m^T \mathbf{D}_{2m} \mathbf{G}_m)^{-1} \mathbf{G}_m^T \mathbf{D}_{2m} \mathbf{e}_{2m}, \quad (29)$$

式中, $\mathbf{D}_{1m} = \text{diag}(\mathbf{D}_{11}, \mathbf{D}_{12}, \dots, \mathbf{D}_{1m})$, $\mathbf{D}_{2m} = \text{diag}(\mathbf{D}_{21}, \mathbf{D}_{22}, \dots, \mathbf{D}_{2m})$, 分别为对角矩阵. 通过式(29)即可获得权向量 \mathbf{w}_m 和偏差 b_m , 获得的最优的非平行分类平面如下:

$$\mathbf{x}^T \mathbf{w}_m + b_m = 0. \quad (30)$$

一个新的数据点 $\mathbf{x} \in \mathbb{R}^n$ 被分配给 m 类 ($m \in 1, 2, \dots, N$), 根据其到它的决策函数的垂直距离的大小决定. 决策函数如下,

$$f(\mathbf{x}) = \arg \min_{m \in 1, \dots, N} \left(\frac{|\mathbf{x}^T \mathbf{w}_m + b_m|}{\|\mathbf{w}_m\|} \right), \quad (31)$$

式中, $|\cdot|$ 表示绝对值操作.

算法 1 描述了本文设计迭代算法的求解过程. 在每次迭代中, 根据当前的 \mathbf{D}_{1m}^p 和 \mathbf{D}_{2m}^p 计算 $\mathbf{z}_m^{(p+1)}$. 并根据计算结果 $\mathbf{z}_m^{(p+1)}$ 更新 \mathbf{D}_{1m}^p 和 \mathbf{D}_{2m}^p , 重复迭代直至算法收敛.

算法 1 基于 L_1 范数距离优化的 MLSTBSVM_{L1} 算法

初始化: 迭代次数 $p=0$, $\mathbf{z}_m = (\mathbf{w}_m \quad b_m)^T$ 是最小二乘 TBSVM 的一个初始值. 根据 \mathbf{z}_m 计算对角矩阵 $\mathbf{D}_{1m}^p = \text{diag}(1/|\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m|)$ 和 $\mathbf{D}_{2m}^p = \text{diag}(1/|\mathbf{e}_{2m} + \mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m|)$, 式(23)的目标函数值 $Q(\mathbf{z}_m)$;

while($\|Q(\mathbf{z}_m^p) - Q(\mathbf{z}_m^{(p+1)})\|_1 \geq 0.001$ 且 $p \leq 30$)

步骤 1: 计算 $\mathbf{z}_m^{(p+1)} = -((1/c_{1m})(\mathbf{H}_m^T \mathbf{D}_{1m}^p \mathbf{H}_m + c_{2m} \mathbf{I}) + \mathbf{G}_m^T \mathbf{D}_{2m}^p \mathbf{G}_m)^{-1} \mathbf{G}_m^T \mathbf{D}_{2m}^p \mathbf{e}_{2m}$;

步骤 2: 根据 $\mathbf{z}_m^{(p+1)}$ 更新对角矩阵 $\mathbf{D}_{1m}^{(p+1)}$ 和 $\mathbf{D}_{2m}^{(p+1)}$, 它们的第 m 个元素分别为 $\mathbf{D}_{1m} = 1/|\mathbf{A}_m \mathbf{w}_m + \mathbf{e}_{1m} b_m|$ 和 $\mathbf{D}_{2m} = 1/|\mathbf{e}_{2m} + \mathbf{B}_m \mathbf{w}_m + \mathbf{e}_{2m} b_m|$;

步骤 3: 计算式(23)的目标函数值 $Q(\mathbf{z}_m^{(p+1)})$;

步骤 4: 转 while 判定, 若为 true, 则更新迭代次数 $p=p+1$, 继续执行步骤 1; 否则, 终止运行, 得到最优预测结果.

end while

3 交通状态预测实例及结果分析

3.1 实验数据及预测流程

为检验改进模型的有效性, 实验数据选自美国 Bellevue 市 405 公路(市中心段)两周的数据(2015.11.2—11.15)作为训练样本和测试样本, 采样时间间隔为 5 min. 选取 405 公路的采样地点分别是 10.31, 10.79, 11.28, 11.78, 12.28, 13.31, 13.92 和 14.27 英里处, 每个采样地点每天产生 288 个记录每天可获得 2 304 个交通状态数据. 此外, 实验数据提供平均车速、平均交通量、平均拥堵率 3 个参数作为研究对象, 研究该路段的交通状态变化趋势, 以避免交通拥堵. 本文按照表 1 中快速路的划分标准对实验数据中的平均车速进行等级划分.

本文提出的基于改进 SVM 的城市道路短时交通状态预测的流程图如图 1、2 所示. 城市道路短时交通状态预测的步骤如下,

步骤 1: 对历史交通数据进行预处理, 即根据道路中各路段的平均速度, 将道路交通状态进行等级划分和样本数据归一化;

步骤 2: 用 K 折交叉验证的方法把预处理后的交通数据分为 K 份, $K-1$ 份为测试数据集, 另外一份为训练数据集;

步骤 3: 交叉验证选择最佳参数 c_{1m} 和 c_{2m} ;

步骤 4: 利用最佳参数计算 MLSTBSVM_{L1} 模型的目标函数值, 判断该值是否满足停止条件, 如果满足则生成 MLSTBSVM_{L1} 预测模型, 否则重新调整参数, 执行步骤 4;

步骤 5:输入测试数据集,通过生成的 $MLSTBSVM_{LI}$ 预测模型获得交通状态预测结果,并对预测结果进行性能评估,以此判断提出的 $MLSTBSVM_{LI}$ 算法的可行性.

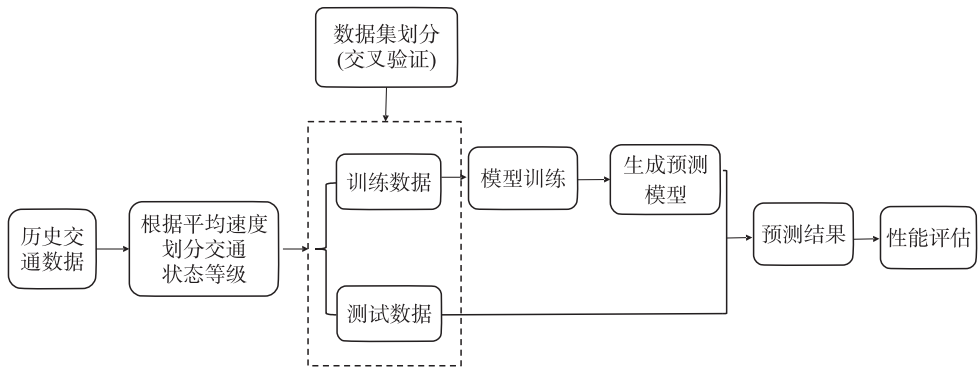


图 1 交通状态预测的流程图

Fig. 1 The flow chart of traffic condition prediction

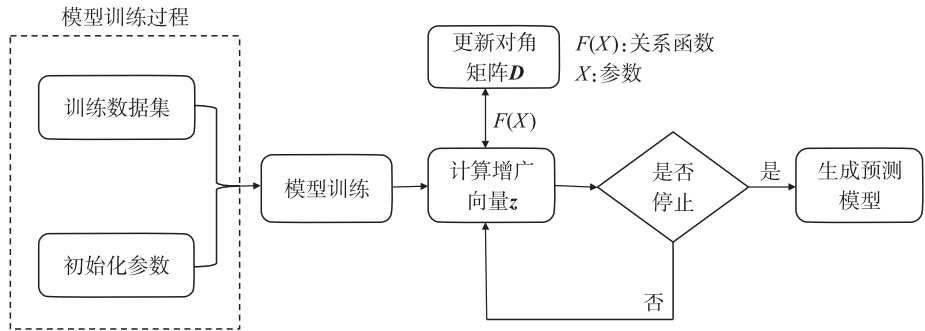


图 2 $MLSTBSVM_{LI}$ 算法的训练过程

Fig. 2 The training process of $MLSTBSVM_{LI}$ algorithm

3.2 数据预处理

为避免数据之间的差异性,在模型学习之前,需要对样本数据(类标除外)作预处理,通常是把样本数据值换算为 $[-1,1]$,这一过程称为归一化.若训练样本为 $X=(x_1,x_2,\cdots,x_n)\in R^n$,则任意一个元素 $x_{ij}(i,j=1,2,\cdots,n)$ 的归一化过程如下 $d=(x_{ij}-\min(X))/(\max(X)-\min(X))$. 训练样本归一化可避免模型训练过程中数值计算的溢出,加快算法的收敛速度.

3.3 实验方案

(1)对城市道路数据分别按照时间间隔为 5,10,15,20 min 取样,用交叉验证(Cross Validation)的思想进行模型的训练,用于城市道路短时交通状态预测.此处取 $K=7$, K 为交叉验证的折数,见表 2.

(2)对于相同的数据集,分别比较在不同的分类算法情况下, $K(2-10)$ 折交叉验证的短时交通状态预测准确率,见表 3,图 3 和图 4.

(3)对于不同的数据集,分别比较在相同分类算法的情况下, K 折交叉验证的短时交通状态预测准确率,见表 4($K=5,10$),图 5($K=6,8$).

3.4 实验结果分析

为检验 $MLSTBSVM_{LI}$ 算法的可行性和有效性,与其他多分类算法(多分类的 GEPSVM 与 TWSVM,分别简称 MGEPSVM 和 MTBSVM)在 4 个数据集上做对比实验,如下.

表 2 3 种多分类算法在数据集上 $K=7$ 的分类结果

Table 2 The results of three multi-classification algorithms on datasets when $K=7$

数据集 ($m\times n$)	MGEPSVM 分类精度 \pm 标准差/%	MTBSVM 分类精度 \pm 标准差/%	$MLSTBSVM_{LI}$ 分类精度 \pm 标准差/%
Traffic_5 29 952 \times 4	68.33 \pm 8.92	66.27 \pm 5.37	70.92 \pm 0.55
Traffic_10 14 976 \times 4	64.94 \pm 1.06	65.36 \pm 1.10	72.54 \pm 0.82
Traffic_15 9 984 \times 4	65.80 \pm 0.41	68.27 \pm 0.48	71.31 \pm 0.88
Traffic_20 7 488 \times 4	66.43 \pm 1.09	69.55 \pm 1.65	71.33 \pm 1.32

表 3 3 种算法在 Traffic_10 数据集上不同 K 值的分类结果

Table 3 The results of three algorithms on Traffic_10 datasets when K is different

交叉验证折数	MGEPSVM 分类精度 \pm 标准差/%	MTBSVM 分类精度 \pm 标准差/%	MLSTBSVM _{L1} 分类精度 \pm 标准差/%
$K=2$	64.70 \pm 0.51	65.26 \pm 0.19	72.40 \pm 0.41
$K=3$	64.77 \pm 0.88	65.32 \pm 0.86	72.50 \pm 0.68
$K=4$	64.93 \pm 0.55	65.30 \pm 0.58	72.54 \pm 0.22
$K=5$	64.91 \pm 0.86	65.30 \pm 0.77	72.54 \pm 0.40
$K=6$	64.85 \pm 0.92	65.26 \pm 0.95	72.56 \pm 0.84
$K=7$	64.94 \pm 1.06	65.36 \pm 1.10	72.54 \pm 0.82
$K=8$	64.92 \pm 0.78	65.33 \pm 0.76	72.50 \pm 0.79
$K=9$	64.94 \pm 1.00	65.33 \pm 0.97	72.56 \pm 0.97
$K=10$	64.99 \pm 1.15	65.32 \pm 1.13	72.60 \pm 0.86

表 2 展示 3 种多分类算法在 4 个数据集上 $K=7$ 的分类结果,从表中数据可知,提出的 MLSTBSVM_{L1} 算法获得最好的分类精度.就 3 种算法的分类精度而言,MLSTBSVM_{L1} 算法比 MGEPSVM 和 MTBSVM 算法至少分别高 2.59% 和 1.78%.这可能归因于 L_1 范数距离在 MLSTBSVM_{L1} 算法中的嵌入,说明 L_1 范数距离有利于数据的分类,检验了提出的 MLSTBSVM_{L1} 算法可行性和有效性.

为观察交叉验证折数 K 对提出算法的影响,本文设置以下实验:(1)3 种分类算法在 Traffic_10 数据集上不同 $K(2-10)$ 值的比较(表 3);(2)3 种分类算法在 Traffic_5 数据集上不同 K 值的比较(图 3(a));(3)3 种分类算法在 Traffic_15 数据集上不同 K 值的比较(图 3(b));(4)3 种分类算法在 Traffic_20 数据集上不同 K 值的比较(图 4).

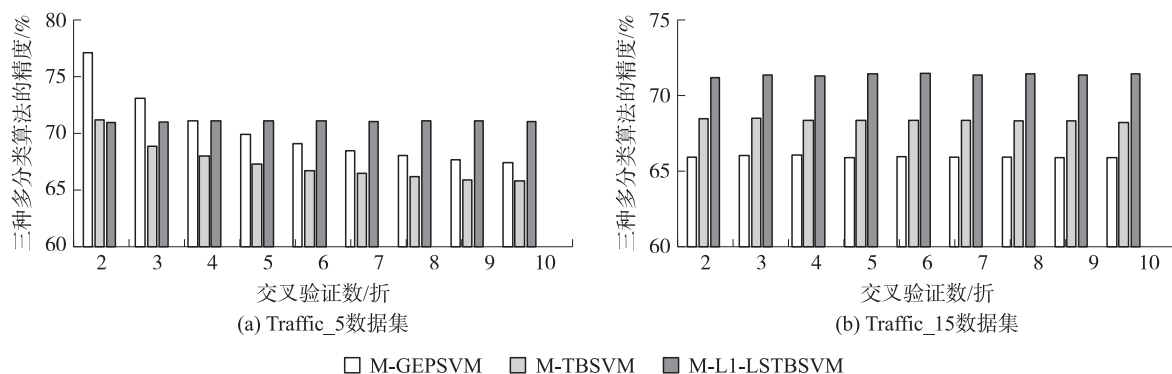


图 3 在 Traffic_5 和 Traffic_15 数据集上不同 K 值的分类结果

Fig. 3 The results of three algorithms on Traffic_5 and Traffic_15 datasets when K is different

通过观察实验结果发现,在图 3(a)中, $K=2,3,4$ 时, MGEPSVM 算法的分类精度均高于 MTBSVM 算法和 MLSTBSVM_{L1} 算法的分类精度.此外,当取不同的 K 值时,在大多数情况下 MLSTBSVM_{L1} 算法的分类精度均高于 MGEPSVM 算法和 MTBSVM 算法的分类精度,且 MLSTBSVM_{L1} 算法的精度值上下波动幅度较其他两个算法变化较小.总的来说,对于不同的 K 值,MLSTBSVM_{L1} 算法均可获得较高的分类精度,这表明 MLSTBSVM_{L1} 算法具有较好的泛化能力,并对 MLSTBSVM_{L1} 算法的可行性和有效性做了进一步的检验.

为进一步检验提出的 MLSTBSVM_{L1} 算法的有效性,本文在 4 个不同的数据集上取相同的 K 值(K 分别为 $K=5,6,8,10$)比较 3 种算法的分类精度,如表 4,图 5 所示,其中,图 5(a)和 5(b)中的 Test 1、Test 2、Test 3、Test 4 分别对应于 3 种分类算法在 4 个不同数据集上的实验结果.

通过观察表 4、图 5 的分类结果可得出,当 K 分别等于 5、6、8、10 时,在 4 个数据集上 MLSTBSVM_{L1} 算法的分类精度总是高于其他相比较的算法.另外,在相同的实验场景下,MLSTBSVM_{L1} 算法的精度值远远

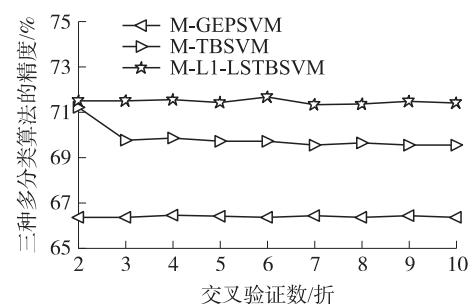


图 4 3 种算法在 Traffic_20 上不同 K 值的分类结果

Fig. 4 The results of three algorithms on Traffic_20 datasets when K is different

高于 MGEPSVM 算法和 MTBSVM 算法的精度值,如图 5(a)和图 5(b)中的 Test 2,相比于 MGEPSVM 算法和 MTBSVM 算法,分别高 7.71%、7.3%和 7.58%、7.17%。这充分体现 $MLSTBSVM_{L_1}$ 算法的优越性,有利于城市道路短时交通状态的预测,可以提高城市道路的通行能力。

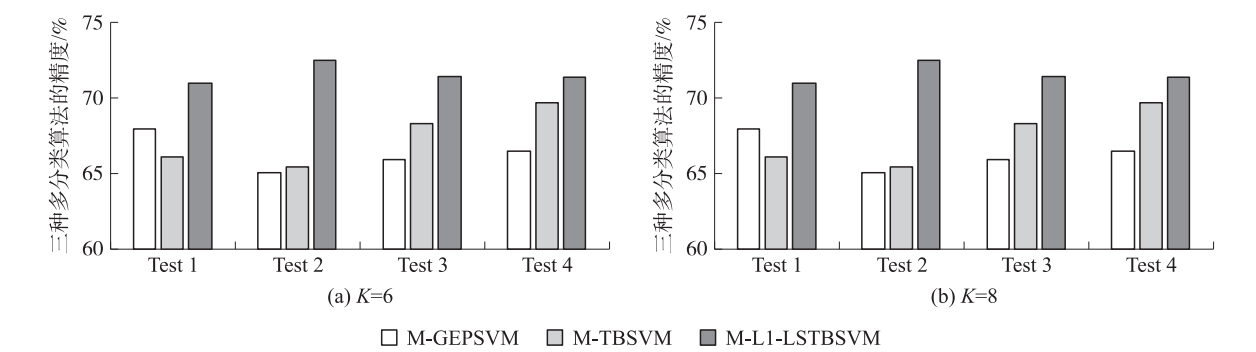


图 5 在 4 个数据集上的分类结果

Fig. 5 The classification results of three algorithms on four datasets

表 4 3 种算法在不同数据集上 $K=5,10$ 的分类结果

Table 4 The results of three multi-classification algorithms on datasets when $K=5$ and $K=10$

交叉验证折数	算法	多分类算法分类精度/%			
		Traffic_5 (29 952×4)	Traffic_10 (14 976×4)	Traffic_15 (9 984×4)	Traffic_20 (7488×4)
K=5	MGEPSVM	69.78±10.23	64.91±0.86	65.77±0.81	66.40±1.09
	MTBSVM	67.09±6.27	65.30±0.77	68.25±0.93	69.71±0.93
	MLSTBSVM _{L1}	70.94±0.40	72.54±0.40	71.35±1.02	71.41±1.26
K=10	MGEPSVM	67.25±7.76	64.99±1.15	65.77±1.28	66.35±1.35
	MTBSVM	65.61±4.88	65.32±1.13	68.12±1.21	69.55±1.44
	MLSTBSVM _{L1}	70.93±0.55	72.60±0.86	71.36±1.37	71.39±1.30

4 结束语

本文提出一种基于改进 SVM 的城市道路短时交通状态预测方法(简称 L_1 -LSTBSVM)。并把 L_1 -LSTBSVM 扩展成多分类算法应用于交通状态的预测,以提高城市道路短时交通状态预测的精度,为交通出行者提供更加准确可靠的交通信息,改善城市道路的通行能力。 $MLSTBSVM_{L_1}$ 算法利用 L_1 范数距离度量点到平面的距离,提高了模型的预测精度及泛化能力。通过实验检验了提出方法的有效性和可行性。

目前,本文提出的 $MLSTBSVM_{L_1}$ 算法仅适用于实验数据量和维度较少的情况,若实验数据量或维度较大,则当前的算法无法对此做出有效的处理,需考虑结合常用的维度约减方法对原数据进行降维,再建模。在后续工作中,我们将对此做进一步的研究。

[参考文献]

[1] 刘擎超. 基于集成学习的交通状态预报方法研究[D]. 南京:东南大学,2015.

[2] 姚智胜. 基于实时数据的道路网短时交通流预测理论与方法研究[D]. 北京:北京交通大学,2007.

[3] 赵亚萍,张和生,杨军,等. 基于最小二乘支持向量机的交通流量预测模型[J]. 北京交通大学学报(自然科学版), 2011,35(2):114-117.

[4] 欧阳俊. 基于多核混合支持向量机的城市短时交通预测[D]. 长沙:中南大学,2011.

[5] 孙占全,潘景山,张赞军,等. 基于主成分分析与支持向量机结合的交通流预测[J]. 公路交通科技,2009,26(5):127-131.

[6] 樊娜,赵祥模,戴明,等. 短时交通流预测模型[J]. 交通运输工程学报,2012,12(4):114-119.

[7] 刘燕. 城市道路交通流状态辨识及决策方法研究[D]. 合肥:合肥工业大学,2011.

[8] 杨兆升,王媛,管青. 基于支持向量机方法的短时交通流量预测方法[J]. 吉林大学学报(工学版),2006,36(6):

881-884.

- [9] 韦凌翔,陈红,王永岗,等. 短时交通流量预测方法[J]. 山东交通学院学报,2017,25(3):22-29.
- [10] YAN H, YE Q, ZHANG T, et al. Least squares twin bounded support vector machines based on L_1 -norm distance metric for classification[J]. Pattern recognition, 2017, 74:434-447.
- [11] YAN H, YE Q, ZHANG T, et al. L_1 -norm GEPSVM classifier based on an effective iterative algorithm for classification[J]. Neural processing letters, 2017, 4:1-26.
- [12] YE Q, YANG X, GAO S, et al. L_1 -norm distance minimization based fast robust twin support vector k -plane clustering[J]. IEEE transactions on neural networks and learning systems, 2018, 29(9):4494-4503.
- [13] YAN R, YE Q, ZHANG L, et al. A feature selection method for projection twin support vector machine[J]. Neural processing letters, 2018, 47(1):21-38.
- [14] MANGASARIAN O L, WILD E W. Multisurface proximal support vector machine classification via generalized eigenvalues[J]. IEEE transactions on pattern analysis and machine intelligence, 2006, 28(1):69-74.
- [15] SHAO Y H, ZHANG C H, WANG X B, et al. Improvements on twin support vector machines[J]. IEEE transactions on neural networks, 2011, 22(6):962-968.
- [16] CARRASCO M, L PEZ J, MALDONADO S. A multi-class SVM approach based on the L_1 -norm minimization of the distances between the reduced convex hulls[J]. Pattern recognition, 2015, 48(5):1598-1607.
- [17] TOMAR D, AGARWAL S. Multiclass least squares twin support vector machine for pattern classification[J]. International journal of database theory and application, 2015, 8(6):285-302.
- [18] XIAO C, NIE F, HUANG H, et al. Multi-class $L_{2,1}$ -norm support vector machine[C]//Proceedings of the IEEE International Conference on Data Mining. Vancouver, Canada: IEEE, 2012:91-100.
- [19] NIE F, WANG X, HUANG H. Multiclass capped L_p -norm SVM for robust classifications[C]//Proceedings of the AAAI Conference on Artificial Intelligence. San Francisco, USA, 2017:1-7.
- [20] DING S, ZHAO X, ZHANG J, et al. A review on multi-class TWSVM[J]. Artificial intelligence, 2017, 2:1-27.
- [21] BRERETON R G, LLOYD G R. Support vector machines for classification and regression[J]. Analyst, 2010, 135(2):230-267.
- [22] HUANG W, SHEN L. Weighted support vector regression algorithm based on data description[C]//Proceedings of the Isecs International Colloquium on Computing, Communication, Control, and Management. USA: IEEE Computer Society, Computer Engineering and Applications, 2008:250-254.
- [23] CHE J X. Support vector regression based on optimal training subset and adaptive particle swarm optimization algorithm[J]. Applied soft computing, 2013, 13(8):3473-3481.
- [24] DIVYA, AGARWAL S. Weighted support vector regression approach for remote healthcare monitoring[C]//Proceedings of the International Conference on Recent Trends in Information Technology. Piscataway: IEEE Press, 2011:969-974.

[责任编辑:陆炳新]