基于粒子群算法的支持向量机的参数优化

陈晋音 熊 晖 郑海斌
(浙江工业大学信息工程学院 杭州 310000)

摘 要 支持向量机(Support Vector Machine,SVM)对内部参数有着极高的依赖性,因此参数的好坏直接决定了SVM的分类效果,比如径向基核函数的参数。为了寻找出与分类问题相契合的参数,将样本数据投影到高维度特征空间,从而在特征空间中计算类内平均距离与类外中心距离之差,并将其作为参数评估的适应值;利用粒子群算法的全局寻优能力,在定义域内生成种群以代表不同的参数取值;利用粒子的随机游走来进行最优参数搜索,并将结果代入SVM进行样本训练。将所提算法与网格算法等进行了比较,结果表明所提算法的参数设定更加准确,分类准确率有显著提高,且算法复杂度并没有明显增加。

关键词 支持向量机,粒子群优化算法,群智能,参数优化,演化算法
中图法分类号 TP3-05 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.035

Parameters Optimization for SVM Based on Particle Swarm Algorithm

CHEN Jin-yin XIONG Hui ZHENG Hai-bin
(College of Information Engineering,Zhejiang University of Technology,Hangzhou 310000,China)

Abstract Support vector machine has high dependence for Hyper-parameters,so parameter setting determines the classification of SVM such as the parameters of RBF kernel function. In order to select proper parameters corresponding to the classification problem,the data set is mapped to the high-dimensional feature space to calculate average distance between classes and the distance between two centers. The difference between results is taken as the fitness value of parameter assessment. Through global optimization ability of particle swarm algorithm,population representing different parameters are generated in the defined domain. The optimal parameter search is performed by random walk of particles,and the results are taken into SVM for training. Compared with grid algorithm,the parameters setting of the proposed algorithm is more accurate,the classification accuracy is significantly improved,and the complexity of the algorithm doesn't increase.

Keywords Support vector machine,Particle swarm optimization algorithm,Swarm intelligent,Parameter optimization,Evolutionary algorithm

1 引言

当今,互联网和移动设备不断普及,大数据时代随之到来,这对人类生活和社会发展产生了重大影响。与此同时,针对大数据的各种数据挖掘算法也在蓬勃发展,如人工神经网络[1]、支持向量机[2]等。其中,支持向量机通过最大化数据集在特征空间上的间隔,将二分类问题转化为一个凸二次规划问题进行求解。相比于其他传统算法,支持向量机不再是最小化经验风险,而是最小化置信范围,因此在处理小样本问题时也拥有较好的分类效果。支持向量机在处理高维数据时拥有精度高、学习能力强等优势,因此被广泛应用于文本分类[3]、手写字特征识别[4]、图像分类[5]、生物序列分析[6]等领域。

支持向量机(SVM)最早由Vapnik等人于1995年提出[2]。之后,Platt等人[7]引入了序贯最小优化算法(Sequential Minimal Optimization,SMO),成功求解了大规模样本的SVM模型训练问题,极大地降低了计算的复杂度。

SVM虽然拥有诸多优点,但在处理多噪点、大样本容量等分类问题时存在不足。当训练样本数为N时,支持向量机的时间复杂度接近$O(N^2)$。另外,SVM在选取参数时具有高度依赖性,对核函数的选取和惩罚因子的设定都有着极高的要求;同时,针对不同的数据集,需要设定不同的参数来契合数据集。

Huang等人利用遗传算法对SVM参数进行优化,并基于精确率和权重惩罚因子提出了一种新的适应值评估函数[8]。Friedrichs等人提出利用固定步长来进行网格搜索,从而寻找合适的参数,但这仅仅适用于参数较少的情况,实际运用过程中存在计算复杂、耗时长等问题[9]。Pan等人[10]在遗传算法的基础上对训练样本的特征进行了加权,并将网格算

到稿日期:2017-04-03 返修日期:2017-06-11 本文受国家自然科学青年基金(61502423),浙江省科技厅科研院专项(2016F50047)资助。
陈晋音(1982-),女,副教授,主要研究方向为数据挖掘、智能计算等,E-mail:chenjinyin@zjut.edu.cn(通信作者);熊 晖(1995-),男,主要研究方向为数据挖掘及其应用;郑海斌(1995-),男,硕士生,主要研究方向为大数据分析、机器学习等。

法与遗传算法这两种参数优化方法进行了比较，加权算法的分类效果得到了显著提升，但加权大大增加了参数的个数，当样本规模增大时，优化效果随之减弱。Kentaro 等人[11]提出利用交叉验证算法来优化参数，其由于精确度较高而成为目前 SVM 的主流参数优化方法。

ACO 由 Dorigo 于 20 世纪 90 年代初提出[12]，随后他和 Gambardella 对其进行了改进，并提出了蚁群系统(Ant Colony System，ACS)[13]。ACO 凭借卓越的全局寻优能力和强大的鲁棒性，在许多优化领域得到了应用。Zhang 等人[14]用 ACO 来优化支持向量机的参数，相比于网格算法，其在一定程度上提升了 SVM 的分类性能，减少了时间消耗。因此，本文将 ACO 作为基本对比算法，比较两者在 SVM 参数优化方面的实验表现。

粒子群优化算法(Particle Swarm Optimization，PSO)是一种基于群智能的随机优化算法，它从随机解出发，通过迭代寻找最优解，并用适应度来评价解的品质。Shi 等人[15-16]通过引入惯性权重，使得算法的探测能力得到了更好的优化。PSO 因具有操作简单、通用性强等特点，引起了学术界的广泛关注，并被应用于诸多领域，成功解决了各种难题。

目前对 SVM 的研究多集中于核函数的选择以及参数的设定上。本文将粒子群算法与支持向量机相结合，利用粒子群的随机游走在定义域内搜寻最优参数解。不同于利用精确率来评估参数，本文利用样本在高维特征空间的类内平均距离与类外中心距离之差来评价参数对于分类器的契合程度，这在很大程度上降低了算法的内存消耗和计算复杂度。

2 相关方法

2.1 SVM 的基本思想

支持向量机作为一种传统的监督学习模型，其核心在于寻找一个能正确对二分类数据进行特征空间划分的超平面，而支持向量则是指在间隔区边缘的训练样本点。

以二维数据的二分类为例。假设已知训练数据集 $S=\{(x_1,y_1),(x_2,y_2),\cdots,(x_i,y_i)\}$，$x_i$ 代表对应的数据向量，y_i 代表 x_i 对应的数据类别，其中 $y_i\in\{1,-1\}$，1 代表属于 A 类，-1 代表属于 B 类，则存在分类超平面 $(w\cdot x)+b=0$，从而对所有样本进行正确划分，使得满足式(1)：

$$y_i((w\cdot x_i)+b)\geqslant 0 \tag{1}$$

超平面如图 1 所示，图中的直线就是二维空间的超平面，直线上下两侧的点代表两类样本的分布情况。

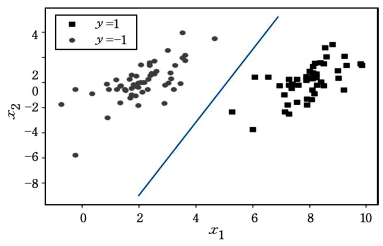

图 1 支持向量机的超平面示意图

Fig. 1 Hyperplane diagram of SVM

若超平面已知，则可以将 x_i 到超平面的距离表示为：

$$D=\|(w\cdot x)+b\| \tag{2}$$

为了使 SVM 的分类效果最佳，需要寻找最优超平面，以最大化两类样本间的间隔。因此，最优分类面问题可以转化为以下二次规划问题：

$$\min\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i \quad \text{s.t. } y_i(w^{\mathrm{T}}x_i+b)\geqslant 1 \tag{3}$$

当数据集线性不可分时，将数据向量 X 向高维特征空间 R^n 投影，从而形成新的向量 X^n，这样便将其转化成了线性可分的情况。

线性不可分的问题虽然得到了解决，但向量内积的求解问题又随之而来。当原样本数据本身的维度偏高时，再向高维度投影可能造成维度爆炸，也就是高维特征空间的维度过多，导致无法计算向量的内积。

为了解决由高维投影导致的维度爆炸问题，引入核函数 $K(x_1,x_2)$，不再是 X 投影之后再求向量 X^n 的内积，而是直接将 X 作为核函数的输入参数代入，然后求出内积值。这不仅巧妙地解决了维度问题，也大大简化了运算过程。

常用的核函数包括：

1)线性核函数：$K(x_1,x_2)=x_1\cdot x_2$；

2)径向基核函数：$K(x_1,x_2)=[(x_1\cdot x_2)+1]^d$；

3)多项式核函数：$K(x_1,x_2)=\exp(-\frac{\|x_1-x_2\|^2}{2\sigma^2})$。

2.2 SVM 参数的设定

以径向基核函数(Radial Basis Function，RBF)为例，其参数的改变实质上是支持向量机向高维度投影的特征空间的复杂度的改变。当核参数 σ 增大时，投影空间的复杂度降低，线性可分程度也将降低；而当 σ 趋于 0 时，特征空间的复杂度也就趋向于无穷，此时虽然能将任意数据映射为线性可分，但会造成严重的过拟合问题。因此，需要针对数据集设置正确的核参数，从而使其与数据样本的分布相契合，由此训练出来的 SVM 的分类效果自然较优。

为了提升支持向量机的性能，设定超参数的常用方法大致分为两类：一类是在参数定义域空间进行网格式搜索，将其中误差最小的参数作为输出结果，如交叉验证；另一类是采用启发式优化，如蚁群算法[17-18]、遗传算法[19-20]、粒子群算法等。

第一类方法虽然稳定性高、参数估计准确，但计算量大且复杂程度高，样本数量偏大时性能表现不佳。相比之下，第二类方法能更加快速地寻找到最优参数，但对于参数的评估，大多使用 ROC 曲线[21]、马修斯相关系数[22]等，这都建立在 SVM 的分类结果基础上，从而不得不重复训练 SVM，导致搜寻效率大幅降低。因此，本文引入了粒子群算法，并采用一种直接在高维特征空间对样本分布进行评估的方法，从而避免了反复计算 SVM 模型。

2.3 粒子群优化算法的主要思想

粒子群优化算法由 Eberhart 和 Kennedy[23]最早提出，是一种源于鸟群和鱼群的群体运动行为的群智能算法，其因为迭代过程中只需要利用目标的取值信息，无需梯度信息，因此操作简单，具有很强的通用性[24]。

群智能是指群体由于个体之间以及个体与环境之间交互而表现出来的智能。蜜蜂采蜜、筑巢、蚂蚁觅食等就是大自然

中典型的例子。受此启迪，MarkMillonas 于 1994 年提出构建一个群智能系统应满足 5 条基本原则[25]。

1)Proximity Principle：群内个体具有能执行简单的时间或空间上的评估和计算的能力；

2)Quality Principle：群内个体能对环境(包括群内其他个体)中关键性因素的变化做出响应；

3)Principle of Diverse Response：群内不同个体对环境中的某一变化所表现出的响应行为具有多样性；

4)Stability Principle：并非环境的每次变化都会导致整个群体的行为模式发生改变；

5)Adaptability Principle：在环境所发生的变化中，若群体的适应度函数发生了相应的改变，那么群体必须能够改变其行为模式。

这些基本原则为群智能的发展奠定了基石，并成功促使了粒子群算法和蚁群算法等诸多智能算法的产生。

粒子群的行动类似于人类的社会行为：某人如果认同社会上存在的某个事物，那么就会努力接近它；反之，则远离它。该事物对应于粒子群算法的优化目标，即问题的最优解；人则类比于粒子，多个粒子组成了种群，而种群对事物的认同与否则取决于其价值，也就是当前粒子解的适应值。为了向目标靠拢，粒子受到 3 方面的影响：1)自身的惯性限制，即移动过程中粒子总会保留部分自身因素，以保证种群的多样化；2)自身目标的限制，即粒子会受到个体最优的吸引，也就是粒子本身找到的最优解；3)种群的限制，即粒子会向群体的最优解靠近。

粒子在移动的过程中，一方面在不断优化自身，另一方面也在搜寻着更优解，自身的适应值如果达到个体最优甚至种群最优时，也就成为了新的优化目标；若迭代数达到上限或者种群的最优解一段时间未得到更新，则粒子停止移动，输出当前的最优个体，并将其作为优化问题的最终解。

3　基于 PSO 的 SVM 实现

3.1　PSVM

为了解决 SVM 参数设定的问题，将粒子群优化算法与支持向量机结合，利用 PSO 的粒子游走来优化 SVM 的参数，建立新的 SVM 模型——PSVM。3.2 节介绍粒子群优化算法中粒子种群的主要演化过程；3.3 节介绍如何利用 SMO 算法来训练支持向量机。

假设已知二类训练数据集 $\{(x_1,y_1),(x_2,y_2),\cdots,(x_i,y_i)\}$，$i$ 表示数据集样本数，目标是建立分类器对测试数据集进行分类：

$$\begin{cases} f(x)\geqslant 0, & y_i=1 \\ f(x)<0, & y_i=-1 \end{cases} \tag{4}$$

其中，$f(x)$表示分类器，y_i 表示数据类型。

本文选用应用最广泛的径向基函数(RBF)作为 SVM 的核函数，即：

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

其中，σ 为需要优化的核函数参数。

SVM 为标准 SVM 模型，也即以下优化问题：

$$\min \frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i \tag{6}$$

其中，C 为惩罚因子，ξ_i 为松弛变量。

用粒子代表核参数 σ，即 $x_i=\{\sigma\}$，可行解的寻找过程通过粒子搜索 R^d 空间完成。每个粒子都有相应的适应值以及速度 V，分别用于评估解对于优化问题的优化程度和表示粒子的空间移动趋势。一方面，为了防止粒子速度过快，可设定速度上限 $V_{\max}$；另一方面，为了加速最优解的寻找，可以限制粒子的空间移动范围。

设定粒子种群数为 20，随机初始化各个粒子的初始位置和初始速度；设定粒子的位置范围为 $x_i\in(-5,5)$，若粒子超出移动范围边界，则将位置重新初始化，继续搜索。同时，设定粒子速度：

$$V\in\left(-\frac{V_m}{2},\frac{V_m}{2}\right) \tag{7}$$

其中，$V_m=X_{\max}-X_{\min}$。

所有粒子生成完毕后，计算出两类数据的中心点：

$$x_m^{(1)}=\frac{\sum_{i=1}^{n(1)}x_i^{(1)}}{n(1)},x_m^{(-1)}=\frac{\sum_{i=1}^{n(-1)}x_i^{(-1)}}{n(-1)} \tag{8}$$

其中，$x_m^{(1)}$ 表示 1 类样本中心，$x_m^{(-1)}$ 表示 -1 类样本中心，$n(1)$表示 1 类数据的样本数，$n(-1)$ 表示 -1 类数据的样本数。

通过式(9)，可计算样本点之间的距离：

$$\begin{aligned} D(x_1,x_2)&=\sqrt{K(x_1,x_1)-2*K(x_1,x_2)+K(x_2,x_2)} \\ &=\sqrt{2-2*K(x_1,x_2)} \end{aligned} \tag{9}$$

根据适应度函数计算各个粒子的适应度。适应度函数的计算如下：

$$Fit(x_i)=\frac{\sum_{i=1}^{n(1)}D(x_m^{(1)},x_i^{(1)})+\sum_{j=1}^{n(-1)}D(x_m^{(-1)},x_j^{(-1)})}{n(1)+n(-1)}-D(x_m^{(1)},x_m^{(-1)}) \tag{10}$$

初始化结束后，记录种群最优粒子的索引以及各个粒子的个体最优解。

3.2　演化过程

初始化完成后，粒子开始逐步移动，进行迭代，且随着移动不断更新种群最优解和个体最优解，如式(11)所示：

$$\begin{cases} v_{id}^{(t+1)}=w\cdot v_{id}^{(t)}+c_1r_1(p_{id}^{(t)}-x_{id}^{(t)})+c_2r_2(p_{gd}^{(t)}-x_{id}^{(t)}) \\ x_{id}^{(t+1)}=x_{id}^{(t)}+v_{id}^{(t+1)} \end{cases} \tag{11}$$

其中，$d=1,2,\cdots,n$(n 表示特征空间的维度)；$i=1,2,\cdots,m$(m 表示种群规模)；t 表示当前粒子进化代数；r_1 和 r_2 为分布于[0,1]的随机数；c_1 和 c_2 为加速常数；$v_{id}^{(t)}$ 表示第 i 个粒子在特征空间中的第 d 维的速度；$x_{id}^{(t)}$ 表示第 i 个粒子在特征空间中的第 d 维的坐标；$p_{id}^{(t)}$ 表示粒子的个体最优解，$p_{gd}^{(t)}$ 表示种群的全局最优解。具体示意效果如图 2 所示。

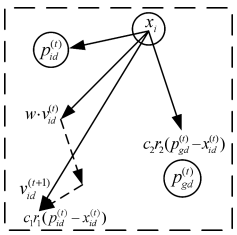

图 2　粒子移动示意图

Fig. 2　Schematic diagram of particle movement

式(11)中,第一项 $w \cdot v_{id}^{(t)}$ 为惯性部分,又称记忆项,其保留了部分上一代的速度和方向,赋予了粒子记忆性,使得粒子具有全局搜索能力;第二项 $c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)})$ 为自身认知项,赋予了粒子个体"个体经验";第三项 $c_2 r_2 (p_{gd}^{(t)} - x_{id}^{(t)})$ 为社会认知项,是粒子与种群连接的"桥梁"。

惯性权重 w 权衡着粒子群的局部搜索和全局搜索,其取值直接影响着粒子群的寻优能力。当权重较大时,粒子群的全局寻优能力强,局部寻优能力弱,有利于防止种群陷入局部最优解;当权重较小时,粒子群的全局寻优能力弱,局部寻优能力强,有利于算法收敛。因此,自适应粒子群优化算法采用动态的惯性权重,使 w 随着迭代的进行而线性减小。通常设置 w 的上限为 0.9,下限为 0.4,使得粒子群前期着重于全局搜索,后期着重于局部寻优:

$$\begin{aligned} W &= (W_{\max} - W_{\min}) \frac{T_{\max} - T}{T_{\max}} + W_{\min} \\ &= 0.5 \frac{T_{\max} - T}{T_{\max}} + 0.4 \end{aligned} \tag{12}$$

动态惯性权重的引入使得 PSO 算法在全局和局部之间找到了一个平衡点,大大提升了算法的性能,促进了 PSO 的广泛应用。

移动过程中,种群不断更新种群最优解和个体最优解,并记录下种群中最优粒子的索引。

当迭代次数达到设定的上限或者算法收敛时,停止迭代,输出当前全局最优粒子的参数 N,为训练最终 SVM 分类器做准备。算法流程图如图 3 所示。

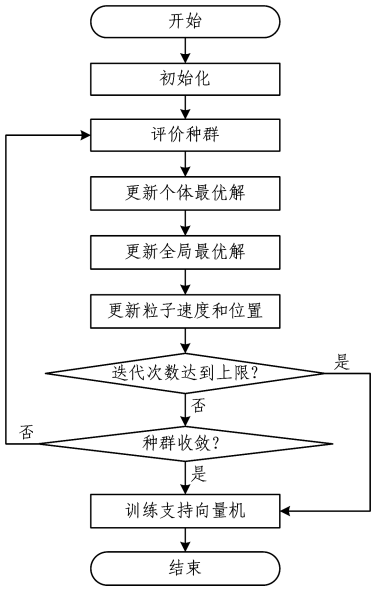

图 3 粒子群优化算法的流程图

Fig. 3 Flowchart of particle swarm optimization algorithm

3.3 SMO 算法

为了求解约束条件式(3)下的二次规划问题,引入拉格朗日函数,具体的计算过程如式(13)所示:

$$L(w,b,\alpha) = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{n} \alpha_i (y_i (w \cdot x + b) - 1) \tag{13}$$

其中,α_i 为拉格朗日系数。固定 α_i 对 w 和 b 求偏导:

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \\ \dfrac{\partial L}{\partial b} = 0 \end{cases} \tag{14}$$

化简后得到:

$$\begin{cases} w = \sum_{i=1}^{n} \alpha_i y_i x_i \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{15}$$

将式(15)代入式(13),使上述问题转化为对偶问题:

$$\max_{\alpha} L(w,b,\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$\text{s.t. } \alpha_i \geqslant 0, i = 1,2,\cdots,n, \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{16}$$

根据库克塔恩(KKT)条件,该问题的解需要满足以下条件:

$$\begin{cases} \text{when } \alpha_i = 0, & w \cdot x_i + b \geqslant 1 \\ \text{when } \alpha_i \in (0,C), & w \cdot x_i + b = 1 \\ \text{when } \alpha_i = C, & w \cdot x_i + b \leqslant 1 \end{cases} \tag{17}$$

为了解决式(16)中当样本容量增大时 L 计算存在的计算量巨大的问题,Platt 引入序贯最小优化算法[7],将原规划问题分解为一系列求解两个拉格朗日算子的子问题。

将训练样本分为两类:1)支持向量,即在间隔区边缘的训练样本点;2)非支持向量,即远离超平面的样本点。具体如式(18)所示:

$$\begin{cases} y_i (w \cdot x_i + b) \leqslant 1, & \text{support vector} \\ y_i (w \cdot x_i + b) > 1, & \text{non-support vector} \end{cases} \tag{18}$$

算法的核心思想是忽略非支持向量,依次选择部分支持向量进行训练。

具体来讲,在式(13)中,因为非支持向量的拉格朗日算子为 0,即可以理解为对于超平面的选择无影响,所以在训练过程中将其剔除;而对于支持向量,从中选取两个来分块训练超平面,其他拉格朗日算子保持不变:

$$\alpha_1 y_1 + \alpha_2 y_2 = \alpha_1^{\text{new}} y_1 + \alpha_2^{\text{new}} y_2 = -\sum_{i=3}^{n} \alpha_i y_i \tag{19}$$

其中,α_1^{new} 为更新后的 α_1,α_2^{new} 为更新后的 α_2。

令 $s = y_1 y_2$,则 $\alpha_1 + s\alpha_2 = N$。将 $\alpha_1 = N - s\alpha_2$ 代入式(16)中,对 α_2 求偏导,得:

$$\begin{aligned} \frac{\partial L(w,b,a)}{\partial \alpha_2} &= sK(x_1,x_1) * (N - s\alpha_2) - K(x_2,x_2) * \alpha_2 - \\ &\quad s(N - s\alpha_2) K(\alpha_1,\alpha_2) + y_2 v_1 - y_2 v_2 - s + 1 \\ &= 0 \end{aligned} \tag{20}$$

其中,v_1 和 v_2 的计算公式如下:

$$v_i = \sum_{j=3}^{n} y_j \alpha_j K(x_i, x_j) \tag{21}$$

通过核函数求取高维特征空间样本点的距离,计算公式如下:

$$\begin{aligned} \eta &= \| X_1^n - X_2^n \|^2 \\ &= X_1^n \cdot X_1^n + X_2^n \cdot X_2^n - 2X_1^n \cdot X_2^n \\ &= K(x_1,x_1) + K(x_2,x_2) - 2K(x_1,x_2) \end{aligned} \tag{22}$$

其中,X_1^n 和 X_2^n 分别表示拉格朗日乘子 α_1 和 α_2 对应的输入样本向量在高维特征空间 R^n 的投影。

由此简化式(20),得:

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2 (E_1 - E_2)}{\eta} \tag{23}$$

其中,E_i 表示分类结果与真实值之差,其计算公式为:

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j) - b - y_i \tag{24}$$

同时,为了满足 KKT 条件,两个拉格朗日乘子 α_1 和 α_2

受惩罚因子 C 的约束。设 $\alpha_2 \in [L,H]$，则当 $s=-1$ 时：

$$\begin{cases} L=\max(0,\alpha_2-\alpha_1) \\ H=\min(C,C+\alpha_2-\alpha_1) \end{cases} \tag{25}$$

当 $s=1$ 时：

$$\begin{cases} L=\max(0,\alpha_2+\alpha_1-C) \\ H=\min(C,\alpha_2+\alpha_1) \end{cases} \tag{26}$$

求取拉格朗日算子的上、下限后，可得：

$$\begin{cases} \alpha_2=H, & \alpha_2^{\text{new}}>H \\ \alpha_2=\alpha_2^{\text{new}}, & \alpha_2^{\text{new}}\in[L,H] \\ \alpha_2=L, & \alpha_2^{\text{new}}<L \end{cases} \tag{27}$$

超平面的偏置 b 可由式(28)求得：

$$\begin{cases} b=b_1, & \alpha_1^{\text{new}}\in(0,C) \\ b=\dfrac{b_1+b_2}{2}, & \alpha_1^{\text{new}}\notin(0,C)\&\alpha_2^{\text{new}}\notin(0,C) \\ b=b_2, & \alpha_2^{\text{new}}\in(0,C) \end{cases} \tag{28}$$

其中，b_1 和 b_2 的计算公式为：

$$b_1=b^{\text{old}}-E_1-y_1(\alpha_1^{\text{new}}-\alpha_1^{\text{old}})K(x_1,x_1)-y_2(\alpha_2^{\text{new}}-\alpha_2^{\text{old}})K(x_1,x_2) \tag{29}$$

$$b_2=b^{\text{old}}-E_2-y_1(\alpha_1^{\text{new}}-\alpha_1^{\text{old}})K(x_1,x_2)-y_2(\alpha_2^{\text{new}}-\alpha_2^{old})K(x_2,x_2) \tag{30}$$

在每一次迭代过程中，更新拉格朗日乘子后，都需要重新计算偏置量 b 以及 E_i。

利用 SMO 算法求解上述问题后，分类函数可表示为：

$$f(x)=\sum_{i=1}^{N}\alpha_i y_i K(x_i,x)+b \tag{31}$$

其中，x 为测试集的输入向量；输出 $f(x)$ 表示分类器的分类结果。

算法流程如图 4 所示。

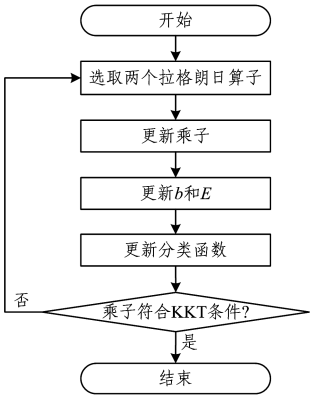

图 4　SMO 算法的流程图

Fig. 4　Flowchart of SMO algorithm

3.4　算法总流程

算法总流程如下：

步骤 1　初始化。设定加速常数 c_1 和 c_2、最大迭代次数 $T_{\max}$、粒子种群数量 m，将当前的迭代数设为 1。本实验中，选择 $c_1=c_2=2.0$，$m=20$，$T_{\max}=500$。在特征空间中随机生成种群 $\{x_1,x_2,\cdots,x_m\}$ 和各个粒子的初始速度 $\{v_1,v_2,\cdots,v_m\}$，若生成的粒子超出空间范围或者速度超出速度上、下限，则将初始值设为最值的一半。

步骤 2　评价种群。加载训练样本数据，然后将每个粒子对应的核参数代入适应度计算函数来计算相对应的适应值。

步骤 3　更新个体最优解。若为第一代粒子，则将每个粒子的个体最优解设为当前位置 $x_{pd}^{(t)}=N^{(t)}$；否则与个体最优解比较粒子的适应度，如果当前位置 $x_{id}^{(t)}$ 比个体最优 $x_{pd}^{(t)}$ 的适应度更优，则将个体最优更改为当前位置，并记录此时的最优适应度。

步骤 4　更新全局最优解。若粒子的个体最优解更改，则比较更改后的粒子与当前种群最优粒子 $x_{gd}^{(t)}$ 的适应值。若当前值比种群的最优值更优，则将种群最优更改为当前位置，并记录种群最优粒子 $x_{gd}^{(t)}$ 的索引。

步骤 5　更新粒子的速度与位置。根据式(11)同步更新种群中所有粒子的位置与速度。

步骤 6　按照步骤 2－步骤 5 进行迭代，并记录迭代次数，当次数达到最大迭代次数 $T_{\max}$ 或者种群收敛时，结束迭代，输出种群全局最优的粒子位置 x_{best}。

步骤 7　将 x_{best} 作为 SVM 的参数，对训练数据集进行训练。

步骤 8　初始化所有的拉格朗日算子，并将所有算子置为 0。

步骤 9　选取两个拉格朗日算子。若为第一次选取，则随机选择；否则，从违反 KKT 条件的算子中选择一个作为 α_2，再从其他的支持向量对应的算子中选取一个作为 α_1，满足 $|E_1-E_2|$ 最大。

步骤 10　更新分类器。利用 SMO 算法更新 α_1，α_2，b，E_i。

步骤 11　遍历所有的拉格朗日乘子，判断其是否满足 KKT 条件，若全部满足，则结束迭代；否则转步骤 9。

步骤 12　根据式(31)求出分类器。用训练完成的 SVM 对测试集进行分类，并计算正确率。

具体的算法流程如图 5 所示。

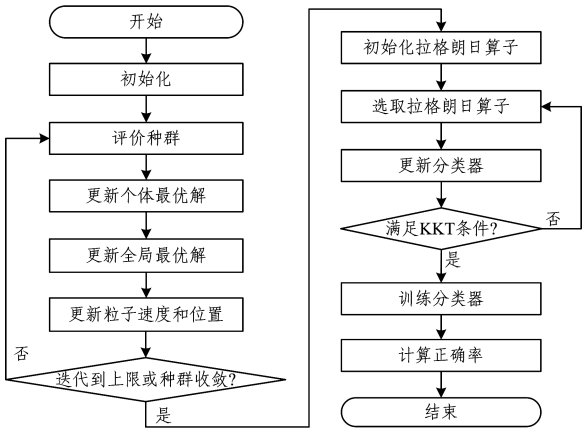

图 5　PSVM 算法的流程图

Fig. 5　Flowchart of PSVM algorithm

3.5　时间复杂度分析

PSVM 主要由粒子移动和 SMO 两部分组成。假设粒子数为 r，迭代次数为 k，适应度计算的时间复杂度 $O(N)$，则 PSO 迭代过程的时间复杂度为 $O(rkN)$，SMO 的时间复杂度在 $O(N)\sim O(N^{2.2})$[26] 之间，因此 PSVM 的复杂度在 $O(rkN+N)\sim O(N^{2.2}+rkN)$ 之间。

PSVM 的空间复杂度主要是指 SVM 训练的空间复杂度：1)核函数的计算；2)SMO 算法，复杂度为 $O(N^2)$。粒子群的移动占据空间约为 $O(rN)$。因此，PSVM 的空间复杂度为 $O(N^2)$。

4 仿真与分析

为了验证 PSVM 的有效性，选取了来自 UCI[27] 的 6 组数据集进行测试，如表 1 所列。其中，针对高维样本，选取 Sonar 数据集；针对大容量样本，选取 Breast cancer 和 Diabetes 数据集，样本总数均达到 600 以上；针对中等规模的样本，选取 Ionosphere，Thyroid 和 Heart disease 数据集。

表 1 6 种数据集的分析

Table 1 Analysis of six datasets

数据集	样本数	数据类标签数	正类数	负类数	样本维度
Sonar	208	2	97	111	60
Ionosphere	351	2	225	126	34
Breast cancer	683	2	444	239	10
Heart disease	270	2	120	150	13
Thyroid	215	2	65	150	5
Diabetes	768	2	268	400	8

表 2 是 PSVM 与 ACO-SVM 在不同数据集上的实验结果，其中 ACO-SVM 的实验结果来自于文献[14]，共 4 组数据集。对 ACO 算法的参数设置如下：蚂蚁数目取 80，信息素挥发系数取 0.5，浓度取 100，初始值取 100。表中 *Gamma* 指径向基核函数的参数，如式(32)所示：

$$K(x_1,x_2)=\exp(-Gamma*\|x_1-x_2\|^2) \tag{32}$$

表 2 PSVM 与 ACO-SVM 的横向对比结果

Table 2 Horizontal contrast results between PSVM and ACO-SVM

数据集	ACO-SVM		PSVM	
	最优核参数 *Gamma*	分类正确率 *Rate*	最优核参数 *Gamma*	分类正确率 *Rate*
Breast cancer	0.00240	0.7403	5.3380	0.9721
Diabetes	0.00002176	0.7700	0.1000	0.9980
Heart	0.12500	0.8400	0.1451	0.8943
Thyroid	0.10145	0.9733	0.0012	0.9999

本实验操作平台系统为 Windows7，内存为 4 GB，软件开发环境为 Pycharm2016.3.1，编程语言为 python3.6，程序由林智仁教授等人基于 LIBSVM[28] 开发[1)]。

为了获得结果，采用 k 折交叉验证法(k-fold Cross Validation Technique)，也即将原样本集随机分为 10 份，依次将其中一份作为测试集，其他部分作为训练集，从而计算出 10 个分类器的平均正确率，以评估分类的效果。同时，选取蚁群算法作为对照，比较两种算法的测试结果，并与其他算法进行分类正确率的比较，结果如表 2 和表 3 所列。可以看出，ACO 对于初始信息素分布的依赖性高，参数设置不当容易影响算法的运行效率；而引入 PSO 后，算法有效地根据训练数据集的分布寻找出了参数，设定更加准确，并且在处理高维度数据方面表现出了极大的优势。

表 3 与其他传统分类算法进行对比的实验结果

Table 3 Comparison of experimental results with other traditional classification algorithms

数据集	网格算法	C4.5	随机森林	模糊贝叶斯	Boosting	PSVM		*SVM*	
						Gamma	*Rate*	*Gama*	*Rate*
Sonar	0.8750	0.7115	0.8462	0.6779	0.7163	0.3584	0.9333	0.0167	0.7700
Ionosphere	0.9459	0.9145	0.9259	0.8177	0.9088	1.1010	0.9476	0.0294	0.9343
Breast cancer	0.9657	0.9399	0.9700	0.9585	0.9499	5.3380	0.9721	0.1000	0.9673
Heart disease	0.8444	0.7667	0.8148	0.8593	0.8000	0.1451	0.8943	0.7692	0.8330

同时，为了进行纵向比较，选取 Ionosphere 数据集，以 0.1 为间隔改变 *Gamma* 取值来测试分类效果，结果如图 6 所示。其中，横坐标表示 SVM 核参数 $N(Gamma)$ 的取值变化，N 取 $\frac{1}{2\sigma^2}$；右纵坐标表示适应值 D(类内均距离和类外中心距离差)的变化；左纵坐标表示针对 Ionosphere 数据集的分类正确率。实线和空心曲线分别表示随着核参数 $N(Gamma)$ 的增大，适应值 D 与平均分类正确率 *Rate* 的变化趋势。从中可以看出，正确率随着适应度的改变而改变，并且两者的走势基本吻合。因此，适应度准确地反映了数据集的可分度，也与分类器的分类效果息息相关。

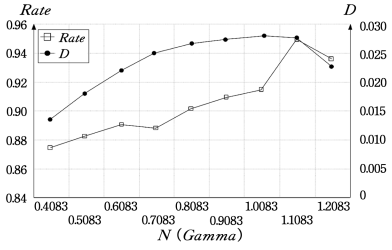

图 6 Ionosphere 数据集的纵向比较结果

Fig. 6 Portrait comparison results of Ionosphere dataset

综上，新的模型可以得到更加契合数据集的核参数，能获取更好的分类超平面，并且能更加突出支持向量机在处理高维数据方面的优势。

表 3 列出分别使用网格算法、C4.5、随机森林、模糊贝叶斯、Boosting、传统 SVM(*Gamma* 参数默认取维度的倒数)与本文算法对 6 种数据集进行分类的实验结果。可以看出，相比其他算法，PSVM 拥有绝对的优势。

结束语 本文针对支持向量机的参数设定问题，引入了粒子群算法，利用粒子在定义域内的随机游走来搜索最优参数。同时，为了提高搜索速度，加速迭代，采用类内平均距离与类外中心距离之差作为粒子的适应度，以此评估粒子的分类效果。相比传统算法将分类正确率作为评价标准，该方法大大地减少了运算量，降低了运算复杂度。

但与此同时，算法依旧存在诸多问题。粒子群算法的引入使 PSVM 在处理大规模样本的分类问题时会一定程度地增加运算量，加剧了时间消耗；且类内距离与类外距离的比较并不能完全反映数据样本的分布情况，也不能用来评估惩罚因子 C 的取值，从而容易造成分类器过拟合或者欠拟合的情况。

1) http://www.csie.ntu.edu.tw/~cjlin/libsvm/

鉴于此，下一步将集中研究如何降低时间复杂度，并建立一个新的模型来更加完善地反映样本的可分度。

参 考 文 献

[1] HAYKIN S O. Neural networks and learning machines[M]. Upper Saddle River, NJ, USA: Pearson, 2009.

[2] CORTES C, VAPNIK V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273-297.

[3] LI B, CHEN N, WEN J, et al. Text categorization system for stock prediction[J]. International Journal of u-and e-Service, Science and Technology, 2015, 8(2): 35-44.

[4] BAUTISTA R M J S A, NAVATA V J L, NG A H, et al. Recognition of handwritten alphanumeric characters using projection histogram and support vector machine[C]// 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). IEEE, 2015: 1-6.

[5] FOODY G M. The effect of mis-labeled training data on the accuracy of supervised image classification by SVM[C]// 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2015: 4987-4990.

[6] BEJERANO G. Automata Learning and Stochastic Modeling for Biosequence Analysis[J]. Goldsmiths University of London, 2003, 4(1): 266-267.

[7] PLATT J. Sequential minimal optimization: A fast algorithm for training support vector machines[C]// Advances in Kernel Methods-support Vector Learning. 1998: 212-223.

[8] HUANG C L, WANG C J. A GA-based feature selection and parameters optimizationfor support vector machines[J]. Expert Systems with Applications, 2006, 31(2): 231-240.

[9] FRIEDRICHS F, IGEL C. Evolutionary tuning of multiple SVM parameters[J]. Neurocomputing, 2005, 64(2): 107-117.

[10] PHAN A V, NGUYEN M L, BUI L T. Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems[J]. Applied Intelligence, 2016, 46(2): 1-15.

[11] ITO K, NAKANO R. Optimizing Support Vector regression hyperparameters based on cross-validation[C]// International Joint Conference on Neural Networks. IEEE Xplore, 2003: 2077-2082

[12] DORIGO M. Optimization, Learning and Natural Algorithms [D]. Italy: Politecnico Di Milano, 1992.

[13] GAMBARDELLA L M, DORIGO M. Solving symmetric and asymmetric TSPs by ant colonies[C]// IEEE International Conference on Evolutionary Computation. IEEE, 1996: 622-627.

[14] ZHANG X L, CHEN X F, HE Z J. An ACO-based algorithm for parameter optimization of support vector machines[J]. Expert Systems with Applications, 2010, 37(9): 6618-6628.

[15] SHI Y, EBERHART R. A modified particle swarm optimizer [C]// IEEE World Congress on Computational Intelligence. 1998: 69-73.

[16] SHI Y, EBERHART R C. Empirical study of particle swarm optimization[C]// Proceedings of the 1999 Congress on Evolutionary Computation, 1999(CEC 99). IEEE, 1999: 1945-1950.

[17] CHEN W, TIAN Y. Parameter Optimization of SVM Based on Improved ACO for Data Classification[J]. International Journal of Multimedia and Ubiquitous Engineering, 2016, 11(1): 201-212.

[18] RONGALI S, YALAVARTHI R. Parameter Optimization of Support Vector Machine by Improved Ant Colony Optimization [C]// Second International Conference on Computer and Communication Technologies. Springer India, 2016: 671-678.

[19] PUNCH III W F, GOODMAN E D, PEI M, et al. Further Research on Feature Selection and Classification Using Genetic Algorithms[C]// International Conference on Genetic Algorithms. Morgan Kaufmann Publishers Inc, 1993: 557-564.

[20] CHOU J S, CHENG M Y, WU Y W, et al. Optimizing parameters of support vector machine using fast messy genetic algorithm for dispute classification[J]. Expert Systems with Applications, 2014, 41(8): 3955-3964.

[21] FAWCETT T. An introduction to ROC analysis[J]. Pattern Recognition Letters, 2006, 27(8): 861-874.

[22] MATTHEWS B W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme[J]. Biochimica et Biophysica Acta (BBA)-Protein Structure, 1975, 405(2): 442-451.

[23] EBERCHART R C, KENNEDY J. Particle swarm optimization [C]// IEEE International Conference on Neural Networks. Perth, Australia, 1995.

[24] ZHANG L P. The Theorem and Practice upon the Particle Swarm Optimization Algorithm[D]. Hangzhou: Zhengjiang University, 2005. (in Chinese)
张丽平. 粒子群优化算法的理论及实践[D]. 杭州：浙江大学，2005.

[25] MILLONAS M, SWARMS M. Phase Transitions and Collective Intelligence[M]// Computational Intelligence: A Dynamic System Perspective. 1992: 137-151.

[26] JOACHIMS T. Making Large-Scale SVM Learning Practical: Technical Reports[R]. 1999: 499-526.

[27] BLAKE C, MERZ C J. UCI repository of machine learning databases[EB/OL]. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[28] CHANG C C, LIN C J. LIBSVM: A library for support vector machines[J]. Acm Transactions on Intelligent Systems & Technology, 2011, 2(3): 27.