

文章编号:1001-9081(2009)12-3273-04

基于非线性共轭梯度法的混沌微粒群优化算法

陈红安, 张英杰, 吴建辉

(湖南大学 计算机与通信学院, 长沙 410082)

(hongan_chen@163.com)

摘要:为了寻找多峰函数的全部极值点,提出一种基于非线性共轭梯度法的混沌微粒群算法。该算法引入混沌序列设置微粒群位置以提高种群的多样性;然后使用改进的微粒群认知模型对可行域内的所有极值点进行全局搜索;最后利用非线性共轭梯度法对混沌微粒群算法搜索到的较优解进行局部搜索以提高解的精度。仿真实验表明,该算法能准确、快速地找到连续可微多峰函数的全部极值点。

关键词:微粒群算法;非线性共轭梯度法;混沌;多峰函数;极值点

中图分类号: TP18 文献标志码:A

Chaotic particle swarm optimization algorithm based on nonlinear conjugate gradient algorithm

CHEN Hong-an, ZHANG Ying-jie, WU Jian-hui

(School of Computer and Communication, Hunan University, Changsha Hunan 410082, China)

Abstract: For searching for all local optimization of the multi-modal function, a chaos-PSO algorithm based on nonlinear conjugate gradient algorithm was proposed. This algorithm employed chaos sequence to initialize particle swarm location in order to enhance the diversity of population, and then utilized improved PSO cognitive model to search all local optimization in the feasible region, and then used nonlinear conjugate gradient algorithm to improve the accuracy of the sub-optimal solution which chaos-PSO has found. The experiments manifest that the hybrid algorithm can properly and quickly find all local optimization of the continuous-differential function.

Key words: Particle Swarm Optimization (PSO) algorithm; nonlinear conjugate gradient algorithm; chaos; multi-modal function; extreme point

0 引言

多峰函数优化问题或多模态函数优化问题(Multi-modal Function Optimization, MFO)中,不仅存在全局最优解,而且有可能存在多个全局最优解和局部最优解。如何构造一种优化算法,使之能求出全部全局最优解和尽可能多的局部最优解,已成为进化算法的一个重要研究方向。人们提出了多种多峰寻优算法,如:协同多群体遗传算法^[1]、多种群进化策略^[2]、混沌微粒群算法^[3]、人工免疫算法^[4]、引入小生境技术的进化策略^[5-6]等。这些算法取得了一定的效果,但也存在着求得的解精度不高等问题。

受人工生命的启示,文献[7]提出了微粒群优化算法,它是一类基于群智能的随机优化算法。微粒群算法由于概念简单、实现容易等优点,已广泛应用于函数优化、神经网络训练等领域。但是,微粒群算法也存在易陷入局部极值点、进化后期收敛速度慢、求解精度较差等缺点。经典的优化算法如:最速下降法、共轭梯度法、Newton 法^[8]等对于连续可微函数,具有收敛速度快、精度高的特点。但是,对于许多多峰函数却容易陷入局部最小(大)值,很难保证达到全局最优。

为了进一步提高多峰寻优求解精度,本文融合了具有较强全局搜索能力的混沌微粒群和具有强大局部搜索能力非线

性共轭梯度法,提出一种基于非线性共轭梯度法的混沌微粒群算法。该算法使用改进的混沌微粒群认知模型对可行域内所有峰值进行搜索,使算法具有全局搜索性;对微粒群得到的较优解使用非线性共轭梯度法进行搜索以提高算法精度和收敛速度,从而克服了微粒群优化算法求解精度较差、后期收敛速度慢的缺点。仿真实验的结果表明,本算法对于寻找连续可微多峰函数的多个极值点效果较好。

1 微粒群算法及其认知模型的改进

1.1 标准的微粒群算法^[9]

标准微粒群优化(Standard Particle Swarm Optimization, SPSO)算法首先初始化一群随机微粒(初始解),然后进化(迭代)找到最优解。每个微粒通过跟踪两个“极值”来更新自己:一个极值是微粒本身找到的最优解,这个位置称为个体极值 p_{best} ;另一个极值是所有微粒找到的最优位置,通常称作 g_{best} 。SPSO 算法的数学描述为:设在一个 n 维的搜索空间中,由 m 个微粒组成的种群 $X = \{x_1, x_2, \dots, x_m\}$ 。其中第 i 个微粒位置为 $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}^T$, 速度为 $v_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}^T$ 。该微粒迄今为止搜索到的最好位置记为 $P_i = \{P_{i1}, P_{i2}, \dots, P_{in}\}^T$, 所有微粒迄今为止搜索到的最好位置为 $p_g = \{p_{g1}, p_{g2}, \dots, p_{gn}\}^T$, 微粒 i 的速度和位置更新公式分别为:

$$v_{id}^{k+1} = w \times v_{id}^k + c_1 \times \text{rand}() \times (p_{id} - x_{id}^k) +$$

收稿日期:2009-06-11;修回日期:2009-08-12。

作者简介:陈红安(1983-),男,湖南益阳人,硕士研究生,主要研究方向:智能计算、智能控制、嵌入式系统; 张英杰(1970-),男,湖南邵阳人,副教授,博士,主要研究方向:复杂工业系统计算机控制、智能计算、计算机系统、CAD; 吴建辉(1970-),男,湖南邵阳人,博士研究生,主要研究方向:智能计算、数字系统。

$$c_2 \times \text{rand}() \times (p_{gd} - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

其中: $d = 1, 2, \dots, n$, $i = 1, 2, \dots, m$ (n 为搜索空间维数, 即待优化的变量个数; m 为种群规模); t 为当前进化代数; c_1, c_2 表示正的加速常数; $\text{rand}()$ 表示 0, 1 之间的均匀分布随机数; ω 为惯性权重, 它决定了微粒先前速度对当前速度的影响程度。

1.2 微粒群认知模型的改进

文献[7]定义了最早的简单微粒群模型。为确定 c_1, c_2 对算法性能影响, 文献[10]根据参数 c_1, c_2 的不同分为四种模型: 知识模型、社会模型、完全模型和自私模型, 并通过实验详细讨论了四种模型的寻优效果。

在知识模型中 $c_2 = 0$, 这意味着整个微粒群所记忆的最好位置(g_{best})对微粒的飞行速度没有影响, 微粒根据自己飞行过程中的最好位置(p_{best})进行搜索, 实际上就是微粒对自身的认知。而在理想情况下, 微粒群中的每个个体对自身的认知, 就代表多目标寻优中的每一个局部优化值。^[3]

为了使可行域内每一个极值点都成为微粒群算法中某些微粒的“必经之路”, 本文将知识模型速度公式中 $\text{rand}()$ 去掉, 并且加入惯性权重 ω ^[3], 得到了改进的微粒群知识模型如式(3)~(4):

$$v_{id}^{k+1} = \omega \times v_{id}^k + c_1 \times (p_{id} - x_{id}^k) \quad (3)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (4)$$

2 混沌微粒群优化算法

2.1 优化问题描述

一般地, 最优化问题由目标函数和约束条件两部分构成:

$$\text{Min}(f(x)) = f(x_1, x_2, \dots, x_n)$$

$$X = (x_1, x_2, \dots, x_n) \in S$$

满足所有约束条件的 S 称为可行域, 可行域中的解称为可行解, 在可行域中使目标函数值最小的解为最优解。对于最大化问题, 可将目标函数乘以(-1), 转化为最小化问题求解。

2.2 基于混沌序列的微粒初始化

混沌运动具有遍历性、内在随机性等特点, 能在一定范围内按其自身的“规律”不重复地遍历所有状态。因此, 将混沌变量引入 PSO 优化算法, 用映射到优化问题的变量取值范围内的混沌序列来产生 PSO 算法中微粒的初始位置, 并且每经过一定的迭代次数就利用混沌序列来更新微粒的位置, 这样使初始微粒的分布更加合理, 从而有利于找到所有极值点^[3]。

选用式(5)所示的 Logistic 映射产生混沌序列:

$$x_{k+1} = \mu \times x_k \times (1.0 - x_k) \quad (5)$$

其中 μ 是控制参数。设 $0 < x_k < 1$ ($k = 1, 2, \dots, n$)。当 $\mu = 4$ 取时, 由式(5)确定的系统完全处于混沌状态。

2.3 非线性共轭梯度法

为了加快微粒群算法的收敛性, 发挥传统数值优化方法在收敛速度和计算精度上的优势, 本文将非线性共轭梯度法引入微粒群算法中。共轭梯度法的基本思想是把共轭性与最速下降方法相结合, 利用已知点处的梯度构造一组共轭方向, 并沿这组方向进行搜索, 求出目标函数的极小点。根据共轭方向的基本性质, 这种方法具有二次终止性。^[8]

非线性共轭梯度法^[8]步骤如下。

第 1 步 在给定范围内求出一个初始点 $x^{(0)} \in \mathbf{R}^n$, $d^{(0)} = -\nabla f(x^{(0)})$, $k = 0$ 。

第 2 步 根据 Goldstein-Armijo 原则确定步长 α_k , $x^{(k+1)}$ 满足约束条件, 令 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ 。

第 3 步 若达到设定迭代次数则终止算法, 否则 $d^{(k+1)} =$

$-\nabla f(x^{(k+1)}) + \beta_k d^{(k)}$, 其中 $\beta_k = \beta_k^{FR} = \frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla f(x^{(k-1)})\|^2}$, 令 $k = k + 1$, 转第 2 步。

2.4 基于非线性共轭梯度法的混沌微粒群算法(NCGPSO)

为了准确、快速地寻找多峰函数的所有极值点, 本文提出的混合优化算法流程如下。

1) 利用混沌序列设置微粒群中微粒的初始位置, 初始化参数包括: 群体规模 N , 惯性权重 ω , 微粒的速度 v , 微粒群算法迭代次数 Itp , 非线性共轭梯度法迭代次数 Itc , 算法总迭代次数 Itd , 精度 ϵ 。

2) 若未达到总迭代次数 Itd 转 3); 否则停止迭代, 输出找到的所有极值点。

3) 计算微粒群中每个微粒的函数值, 得到微粒经历过的最好位置 p_{best} 。

4) 根据式(3)~(4)更新每个微粒的速度和位置, 若未达到微粒群算法迭代次数 Itp 转 3); 否则转 5)。

5) 对每个微粒的最好值 p_{best} , 用非线性共轭梯度法以 p_{best} 为初始点进行搜索, 搜索得到的点记为 p_{best}' , 并用 p_{best}' 代替 p_{best} 。若未达到非线性共轭梯度法迭代次数 Itc 转 5); 否则转 6)。

6) 若 $\|\nabla f(x^{(p_{best})})\| \leq \epsilon$ 或者某微粒收敛到一个极值点不再进化, 则通过极值点的管理决定是否将得到的极值点放入极值点集, 将该微粒通过混沌序列重置。

7) 若达到迭代次数 Iti 利用混沌序列更新微粒群以提高种群多样性, 转 2); 否则直接转 2)。

流程如图 1 所示。

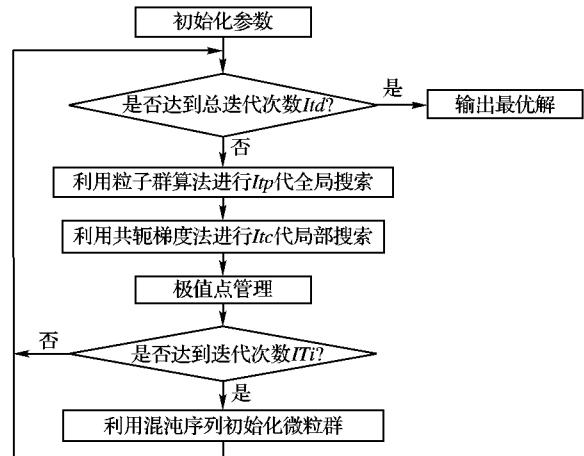


图 1 基于非线性共轭梯度法的混沌微粒群算法流程

收敛判断 极值点可以分为两类: 1) 内部极值点; 2) 边界极值点。对于内部极值点, 其 $\|\nabla f(x)\| = 0$; 对于边界极值, 其 $\|\nabla f(x)\|$ 可能不会等于 0。所以, 对于处理这两种极值点采用不同的方法: 对于内部极值可以通过 $\|\nabla f(x)\| < \epsilon$ 判断其是否收敛, 决定是否将其加入到极值点集中; 对于边界极值, 判断微粒是否停滞在某一个极值点上, 采用的方法是: 如果某一微粒经过梯度下降搜索后, 连续 N 次停留在某一极值点, 就将此点加入到极值点集中。

极值点管理 建立极值点库^[2], 对新加入的极值点与已存在的极值点, 判断它们是否同峰。若是同峰并且大于已存在的极值点的适应值则舍弃该极值点; 若是同峰但小于已存在的极值点的函数值则替代该极值点(最小化问题); 若异峰将其加入到极值点的集合中。

同峰的判断 新发现的极值点与已发现的某个极值点是否同峰一直是求多极值点优化问题的难点。在适应值共享遗传算法中, 通过比较两极值点间的距离与预先设定的小生境半径或峰半径来确定相距较近的极值点是否同峰, 当极值

点分布不均匀时容易出现判断错误。^[2]本文提出一种同峰判断方法,原理如下:设 A, B 为两个极值点,在 A, B 之间插入 $(n-1)$ 个点将 \overrightarrow{AB} 分成 n 等份,计算这 n 段的同峰算子:

$$tf = (f(x_{i+1}) - f(x_i)) / \rho$$

其中: $\rho = \|x_{i+1} - x_i\|$, $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$, d 是维数。如果同峰算子的值出现由负变正,则意味着在这两点之间有山谷存在,那么 A, B 应该是异峰极值点,否则为同峰极值点。

本算法利用微粒群算法在可行域内找到较好的解,再利用非线性共轭梯度法对找到的较好点进行快速、高精度地搜索,这样可以解决微粒群算法收敛速度慢、精度不高等缺点。

3 实验仿真与分析

为验证算法,本文做了下面两组测试实验,实验目的分别是:全局最优解的精度,测试算法是否可以搜寻到多峰函数的所有山峰。

在两个测试中,统一的参数为 $\omega = 0.7298, c_1 = 2$, 非线性共轭梯度法迭代次数 Itc 为 2。

表 1 函数 1 计算结果

序号	NCGPSO			传统混沌微粒群算法		
	变量		函数值	变量		函数值
	x	y		x	y	
1	-2.805118	3.131313	659.999999999979540	-2.755655	3.148154	659.909233753405830
2	-3.779306	-3.283187	659.999999998924300	-3.689165	-3.262838	659.571872944409620
3	3.584436	-1.848121	659.999999996505720	3.546963	-1.764230	659.850445928961560
4	3.000012	1.999996	659.999999994976630	3.031245	2.138370	659.528533885590950

表 2 函数 2 计算结果

序号	NCGPSO			传统混沌微粒群算法		
	变量		函数值	变量		函数值
	x	y		x	y	
1	0.634920	0.634927	3.259986293038022	0.651715	0.636645	3.245174508833912
2	-0.634921	-0.634922	3.259986294268485	-0.632375	-0.635870	3.259608091311898
3	-0.634925	0.634923	3.259986293812987	-0.627349	0.634373	3.257053631873454
4	0.634922	-0.634922	3.259986294296454	0.632542	-0.644381	3.255067235047727

由表 1 中数据可知,本文算法每次都精确找到了全部极值点,与文献[1~2]相比,解的质量得到了提高,而且没有多余和遗漏的极值点。由表 2 中数据可知,本文算法找到函数 2 的四个全局极大值的解的精度比文献[11]的精度要高。并且,由表 1~2 可知,本文的算法比文献[3]中传统混沌微粒群算法找到的解的精度要高。

3.2 测试二

3.2.1 测试函数

为验证算法是否可以搜寻到所有山峰,本文选用下面两个经典函数进行测试:

函数 1 Himmelbau 函数,共有 4 个全局极大值点,32 个局部极值点。

$$\max f(x, y) = 660 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2; \\ x, y \in [-6, 6]$$

函数 2 有无穷多个局部极大值点,其中只有一个全局极大值点在 $(0, 0)$,其值为 1,在该点周围有两圈脊。

$$f(x, y) = 0.5 - \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{1 + 0.001 \times (x^2 + y^2)}; x, y \in [-10, 10]$$

3.2.2 实验结果

实验结果如图 2~3 所示。图 2 为本实验找到的 36 个峰值点的分布图。实验使用的参数是:微粒的个数 $N = 15$, 微粒群算法迭代次数 It_p 为 10, 总迭代次数 It_d 为 100, $\varepsilon = 0.001$ 。图 3 为本实验找到的 220 个峰值点的分布图。实验使

3.1 测试一

3.1.1 测试函数

为检验算法精度,本文选用以下两个函数,对 NCGPSO 和传统混沌微粒群算法进行比较实验:

函数 1 Himmelbau 函数,此函数有 4 个极大值点,理论极值为 660.0。

$$\max f(x, y) = 660 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2; x, y \in [-6, 6]$$

函数 2 是一个典型的具有众多局部最优解的不等高多峰测试函数。共有 4 个全局极大值点,32 个局部极值点。

$$\max f(x, y) = 2 + x \times \sin(4\pi x) - y \times \sin(4\pi y + \pi); \\ x, y \in [-1, 1]$$

3.1.2 实验结果

NCGPSO 实验参数为:微粒群算法迭代次数 It_p 为 10, 微粒的个数 N 为 20, $\varepsilon = 0.001$, 函数 1 总迭代次数 It_d 为 10, 函数 2 总迭代次数 It_d 为 100; 传统混沌微粒群算法^[3]的迭代次数 I 为 1200, 微粒的个数 N 为 20。计算结果如表 1~2 所示。

用的参数是:微粒的个数 $N = 50$, 微粒群算法迭代 It_p 次数为 5, 总迭代次数 It_d 为 200, $\varepsilon = 0.1$ 。本算法迭代 200 次找到了 1 个全局最大值和 219 个局部极值。

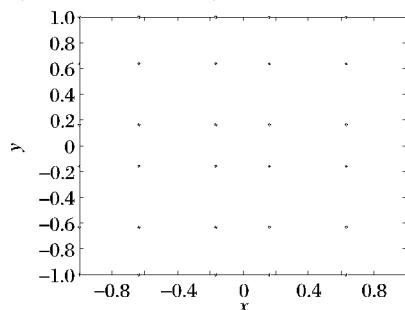


图 2 本文算法解的分布图(函数 1)

图 4~5 是在迭代过程中找到的极值点的个数。由图可知,算法在第二次迭代甚至第一次迭代时就找到了极值点,由此可见算法的收敛速度快,能够较快找到极值点。但从图 4 可知,算法在大概第 60 次迭代的时候找到了所有的极值点,在前 20 次迭代时找到了较多的极值点,但后期找到极值点的速率减慢,这主要是由于在后期找到的极值点与前面找到的极值点相同,这表明在迭代后期具有较多的重复搜索。但这些重复搜索的作用是提高解的精度,因为若两极值点同峰,精度更高的极值将替代精度低的极值。函数 2 具有无穷多个极值点,从图 5 可以看出,在迭代后期算法依然可以找到极值

点,搜索并没有停滞,说明该算法具有较强的全局搜索性,具有寻找所有极值点的能力。如何避免极值点的重复搜索而不丢失极值点,从而经过较少的迭代次数找到所有的极值点,是需要进一步研究和解决的问题。

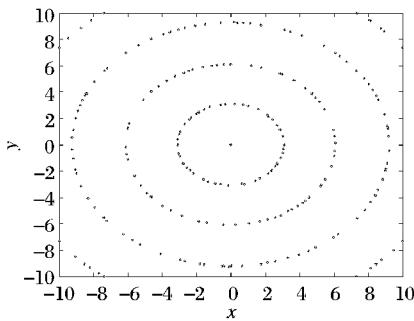


图 3 本文算法解的分布图(函数 2)

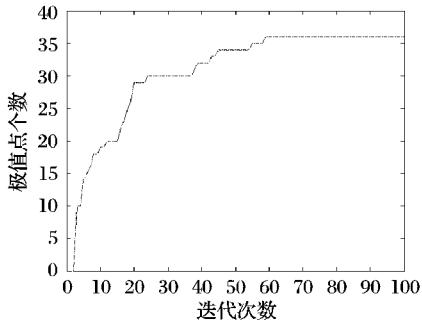


图 4 迭代过程中找到的极值点个数(函数 1)

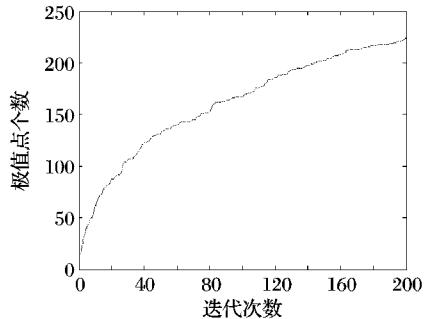


图 5 迭代过程中找到的极值点个数(函数 2)

4 结语

本文提出的基于非线性共轭梯度法的混沌微粒群优化算法综合了微粒群算法、混沌运动、共轭梯度法的特点,利用混

沌运动的遍历性设置微粒的初始位置,并且每隔一定的迭代次数利用混沌序列更新微粒的位置以提高微粒群的全局搜索性能,利用改进微粒群的认知模型实现局部寻优;利用非线性共轭梯度法提高算法的收敛速度和解的精度,可以快速可靠求得多峰函数的极值点。算法不仅在精度上有大幅度提高,在寻找多峰时也取得比较好的效果。对于连续可微函数,非线性共轭梯度法恰好显示了它的搜索速度快、精度高的特点。事实上,只要种群在运动中有某个点落在全局最优值区间,较高精度的最优值就能求得。基于混沌序列的认知模型的微粒群算法能够找到每个全局最优值和局部极值,具有较好的全局搜索性能。从实验结果可以看出,本文提出的算法能快速求得精度非常高的局部最优值。当然,随着函数的性质不同,可能需要调整某些参数。

参考文献:

- [1] 李敏强,寇纪淞.多模态函数优化的协同多群体遗传算法[J].自动化学报,2002,28(4):497-504.
- [2] 王湘中,喻寿益.多模态函数优化的多种群进化策略[J].控制与决策,2006,21(3):285-288.
- [3] 沈洪远,彭小奇,王俊年,等.基于混沌序列的多峰函数微粒群寻优算法[J].计算机工程与应用,2006,42(7):36-38.
- [4] 吴义虎,杨秋实.一种改进的克隆选择算法在多峰值函数优化中的应用[J].计算机工程与科学,2008,30(4):56-59.
- [5] IZUMI K, HASHEM M M A, WATANABE K. An evolution strategy with competing subpopulations [C]// CIRA '97: IEEE International Symposium on Computational Intelligence in Robotics and Automation. Washington, DC: IEEE, 1997: 306-311.
- [6] PORTER F, XUE B. Niche evolution strategy for global optimization [C]// Proceedings of the 2001 Congress on Evolutionary Computation. Washington, DC: IEEE, 2001, 2: 1086-1092.
- [7] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Washington, DC: IEEE, 1995: 1942-1948.
- [8] 李董辉,童小娇,万中.数值最优化[M].北京:科学出版社,2005: 78-85.
- [9] 李莉,李洪奇,谢邵龙.一种有效的多峰函数优化算法[J].计算机应用研究,2008,25(10):2973-2976.
- [10] KENNEDY J. The particle swarm: Social adaptation of knowledge [C]// IEEE International Conference on Evolutionary Computation. Washington, DC: IEEE, 1997: 303-308.
- [11] 徐雪松,诸静.基于免疫原理的多模态函数优化算法[J].南京理工大学学报,2005,29(1):90-93.

(上接第 3266 页)

5 结语

支持向量机能否很好地分类的一个重要因素就是其模型能否有效提取数据的分布与关联信息。数据之间的等价性关系以及概率分布特性都在支持向量机的分类过程中起到了很重要的作用,然而传统 SVM 模型对数据本身信息的获取这方面考虑得却很少。PRSVM 模型将数据的概率分布特性与等价性关系同时融入到支持向量机模型中,很好地刻画了数据间的有效联系,弥补单一参数控制的不足,有效获取并利用了数据信息。实验证明 PRSVM 模型有更好的分类效果和泛化能力。

参考文献:

- [1] CORTES C, VAPNIK V. Support vector networks [J]. Machine Learning, 1995, 20(3): 273-297.
- [2] VAPNIK V N. Estimation of dependencies based on empirical data

- [M]. Berlin: Springer-Verlag, 1982.
- [3] HSU C W, LIN C H. A comparison of methods for multi-class support vector machines [J]. IEEE Transactions on Neural Networks, 2002, 13(2): 415-425.
- [4] HUANG H P, LIU Y H. Fuzzy support vector machines for pattern recognition and data mining [J]. International Journal of Fuzzy Systems, 2002, 4(3): 826-835.
- [5] LEE K Y, DAE-WON K. Possibilistic support vector machines [J]. Pattern Recognition, 2005, 38(8): 1325-1327.
- [6] 梁红霞,闫德勤.粗糙支持向量机[J].计算机科学,2009,36(4):208-210.
- [7] PAL N R, PAL K, BEZDEK J C. A mixed C-means clustering model [C]// Proceedings of the Sixth IEEE International Conference on Fuzzy Systems. Washington, DC: IEEE, 1997: 11-21.
- [8] LIU X, CHENG G, WU J. Analyzing outliers cautiously [J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(2): 432-437.