基于改进量子粒子群的K-means聚类算法及其应用

李 玥[1], 穆维松[1,2], 褚晓泉[1], 傅泽田[2,3†]

(1. 中国农业大学 信息与电气工程学院，北京 100083；2. 中国农业大学 食品与安全北京实验室, 北京 100083；3. 中国农业大学 工学院，北京 100083)

摘 要: 针对传统K-means聚类算法受初始类中心影响导致聚类准确度较低的问题,利用量子粒子群优化算法全局搜索能力强、收敛速度快的优势,提出一种基于改进量子粒子群的K-means聚类算法. 为防止量子粒子群优化算法陷入局部极值,采用具有高斯扰动的局部吸引子以提高种群跳出局部最优的能力;为提高算法的收敛速度,采用加权更新种群平均最优位置以充分发挥精英粒子的优势;通过对收缩-扩张因子和随机变量参数进行交叉实验,选出最佳参数组合策略. 在标准测试函数上的仿真结果表明:改进的量子粒子群优化算法在寻优精度、收敛速度以及稳定性上都有显著提高;通过对比7种聚类算法在UCI数据集上的聚类结果可知,所提出的聚类算法具有更好的聚类性能,可以有效降低K-means对初始聚类中心的依赖. 最后,将该方法应用于我国鲜食葡萄市场客户分类中,以验证该方法的有效性和实用性. 通过实证分析可知,基于改进量子粒子群的K-means聚类算法结构简单、精度高,具有一定的推广性.

关键词: K-means聚类算法；量子粒子群优化算法；聚类中心；聚类分析；客户分类；鲜食葡萄

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2020.1302 **开放科学(资源服务)标识码(OSID):**

引用格式: 李玥, 穆维松, 褚晓泉, 等. 基于改进量子粒子群的K-means聚类算法及其应用 [J]. 控制与决策, 2022, 37(4): 839-850.

K-means clustering algorithm based on improved quantum particle swarm optimization and its application

LI Yue[1], *MU Wei-song*[1,2], *CHU Xiao-quan*[1], *FU Ze-tian*[2,3†]

(1. College of Information and Electrical Engineering，China Agricultural University，Beijing 100083，China；2. Beijing Laboratory of Food Quality and Safety，China Agricultural University，Beijing 100083，China；3. College of Engineering，China Agricultural University，Beijing 100083，China)

Abstract: The original K-means clustering algorithm is seriously affected by initial centroids of clustering and easy to fall into local optima. To overcome these shortages, this paper uses the quantum particle swarm optimization (QPSO) which has power ability of global search and quick convergence rate to optimize the initial clustering centers of the original K-means algorithm. As the QPSO algorithm can easily fall into the local optimum, the local attractor with Gauss disturbance is used to make the population jump out of the local extremum. To improve the convergence speed of the algorithm, the weighted average best position is used to take advantage of the elite particles. The contraction-expansion factors and random variables are combined in order to select the best parameter strategy. The simulation results on various benchmark problems show that the optimization accuracy, convergence speed and stability of the improved optimization algorithm are significantly improved. Experimental results on the typical UCI datasets show that the proposed method is superior to compared algorithms. Finally, this method is applied to the customer classification of table grapes, which shows the effectiveness and practicability of the proposed clustering algorithm. Through the empirical analysis, it is also proved that this model can be promoted and applied.

Keywords: K-means clustering algorithm；quantum particle swarm optimization algorithm；cluster centers；clustering analysis；customer classification；table grapes

收稿日期: 2020-09-17；**录用日期:** 2021-01-19.

基金项目: 现代农业产业技术体系建设专项项目(CARS-29).

责任编委: 刘宝碇.

†通讯作者. E-mail: fzt@cau.edu.cn.

0 引 言

聚类分析是数据挖掘领域中一个非常热门的研究课题,也是一种重要的数据分析技术. 聚类是将物理或抽象的集合分成相似对象类的过程,使同一个簇中对象间具有较高的相似度,而不同簇中对象间差别较大[1-2]. 传统的聚类算法可以分为基于划分、基于层次、基于密度、基于网格、基于模型等几个类别[3]. K-means算法作为一种经典的基于划分思想的聚类算法,具有结构简单、收敛速度快、局部搜索能力强等优点,目前已被广泛地用于统计学、市场营销、客户分类等诸多领域[4-5]. 然而,传统的K-means算法存在对初始聚类中心敏感、全局搜索能力较差、聚类精度低等问题,若随机选取初始聚类中心可能会导致聚类结果陷入局部最优甚至无解[6]. 因此,如何选取一组合理的初始聚类中心,在降低聚类结果波动性的同时,又能得到较高的聚类准确率具有重要的现实意义.

针对传统K-means聚类算法对初始聚类中心取值敏感的问题,当前研究主要分为基于距离和基于密度两类方法[7-8]. 基于距离的方法虽然时间开销较小,但该方法对孤立点过于敏感;而基于密度的方法尽管能够比较准确地反应数据的分布情况,但计算开销较大[9]. 基于此,很多学者对上述问题进行了研究,提出了一系列经典K-means优化方法,例如K-means++和K-medoids[10]. K-means++聚类[11]的核心思想是选取距离尽可能远的数据作为初始聚类中心,较好地解决了聚类结果对初始聚类中心选取过于依赖的问题;K-medoids聚类[12]通过数据样本的中位数选取聚类中心,在一定程度上可以削弱异常值对聚类结果的影响,从而解决了传统K-means算法易陷入局部最优的问题,但该算法时间开销过大,不适合大规模数据聚类. 尽管许多学者对K-means算法的初始类中心敏感问题做出了一些改进,有效提高了K-means算法的准确率,但仍没有解决K-means算法全局搜索能力较差的问题,因此,聚类结果仍可能陷入局部最优.

近年来,群智能优化算法的出现解决了K-means算法对初始中心点过度依赖的问题[13-14]. 该类算法由于具有强大的全局搜索能力而被广泛用于聚类领域,获得了较为理想的应用效果. 文献[15]提出了一种改进的混合粒子群和K-means的聚类算法,通过引入小概率随机变异操作增强种群的多样性,提高了K-means算法的全局搜索能力;文献[16]提出了一种基于改进蜂群算法的K-means聚类算法,有效地消除了初始簇中心对聚类结果的影响,降低了K-means算法陷入局部最优的可能. 粒子群优化算法是一种设定参数少、收敛速度快、适用于大规模高维度数据集的典型群体智能优化算法,但通常无法保证收敛于优化问题的全局最优解,存在搜索精度不高等缺陷[17-19]. 文献[20]受量子力学等相关理论的启发,在粒子群优化算法的基础上提出了具有量子行为和全局收敛性能的量子粒子群优化算法. 量子粒子群优化算法中没有速度更新公式,与粒子群优化算法相比,收敛速度更快,全局寻优能力更强,控制参数更少. 量子粒子群优化算法克服了传统粒子群优化算法中无法保证全局收敛的缺点,是近年来优化技术领域的一个研究热点,但该算法也存在着后期收敛速度较慢、容易早熟收敛的问题[21]. 为了防止量子粒子群优化算法陷入局部极值,文献[22]采用高斯概率分布和混沌变异算子产生随机数来代替量子粒子群优化算法中的参数,在寻优精度和收敛性方面有显著优势;为了提高量子粒子群优化算法的收敛性能,文献[23]提出对特征长度引入权重系数来控制算法的搜索能力,增强了粒子之间的协作性. K-means聚类算法的核心思想是最小化聚类中心到样本点之间的距离和,而量子粒子群优化算法具有全局优化作用,因此通过将K-means聚类算法较强的局部搜索能力与改进量子粒子群更强的全局搜索能力优势相结合,可以获得更好的聚类效果.

鉴于K-means聚类算法和量子粒子群优化算法各自优点,本文首先对量子粒子群优化算法作出改进,以提高算法的寻优精度和收敛速度;然后用改进的量子粒子群优化K-means算法中聚类中心的位置,降低初始聚类中心的影响和陷入局部最优解的可能,改善聚类性能;最后,将改进的聚类算法应用于我国鲜食葡萄市场客户分类中,以验证本文改进方法的有效性和实用性.

1 研究方法

1.1 K-means聚类算法

K-means聚类算法在最小化误差的基础上将数据划分为预定的类数k,采用距离作为相似性评估,利用簇$E_j(j = 1,2,\ldots,k)$的中心e_j表示该簇. 用$\mathrm{dist}(o_i,o_j)$表示两个数据对象o_i与o_j之间的欧氏距离,其计算公式如下:

$$\mathrm{dist}(o_i,o_j)=\sqrt{(o_{i1}-o_{j1})^2+\ldots+(o_{ip}-o_{jp})^2}, \tag{1}$$

其中p为数据对象属性的个数.

使用误差平方和SSE作为度量聚类质量的目标函数,表示簇内样本围绕簇中心的紧密程度. SSE越小,组内样本相似度越高. SSE的计算公式如下:

$$\mathrm{SSE}=\sum_{j=1}^{k}\sum_{o\in E_j}\mathrm{dist}(o,e_j), \tag{2}$$

$$e_j=\frac{1}{n_j}\sum_{o\in E_j}o. \tag{3}$$

其中n_j为第j个簇E_j中样本数据的个数.

1.2 量子粒子群优化算法

粒子群优化算法表达如下:

$$v_{id}^{k+1}=v_{id}^{k}+c_1r_1(p_{id}^{k}-x_{id}^{k})+c_2r_2(p_{gd}^{k}-x_{id}^{k}), \tag{4}$$

$$x_{id}^{k+1}=x_{id}^{k}+v_{id}^{k+1}. \tag{5}$$

其中:x_{id}^{k}为第i个粒子在k时刻的位置;v_{id}^{k}为第i个粒子在k时刻的速度,$i=1,2,\ldots,N$,N为粒子数目,$d=1,2,\ldots,D$,D为粒子维度;c_1和c_2为加速系数,即学习因子;r_1和r_2为[0, 1]内的随机数;p_{id}^{k}为第i个粒子的历史最优位置;p_{gd}^{k}为种群历史最优位置.

粒子轨迹分析表明:如果每个粒子收敛到它的局部吸引子a_i^k,则粒子群优化算法收敛[24]. 其中a_{id}^{k}表达式如下:

$$a_{id}^{k}=\varphi p_{id}^{k}+(1-\varphi)p_{gd}^{k}, \tag{6}$$

$$\varphi=\frac{c_1r_1}{c_1r_1+c_2r_2}. \tag{7}$$

就动力学的角度而言,粒子群优化算法中粒子的收敛过程是以a点为吸引子,随着速度减小不断接近a点,最后收敛到a点. 因此,在整个迭代过程中,a点处实际上存在某种形式的吸引势能场吸引着粒子,使得整个粒子群能保持聚集性. 但是由于在粒子群系统中,粒子的搜索空间是一个有限的区域,不能保证收敛到全局最优.

假设一个粒子群优化算法系统是一个量子系统,每个粒子具量子行为,并通过求解薛定谔方程得到粒子在空间中某一点出现的概率密度函数. 应用蒙特卡罗方法,得到在第$k+1$次迭代x_i的第d维为

$$x_{id}^{k+1}=a_{id}^{k}\pm\frac{1}{2}L_{id}^{k}\ln\left(\frac{1}{u_{id}^{k+1}}\right), \tag{8}$$

$$L_{id}^{k}=2\beta|C_d^k-x_{id}^{k}|, \tag{9}$$

$$C_d^k=\frac{1}{N}\sum_{i=1}^{N}p_{id}^{k}. \tag{10}$$

其中:L_{id}^{k}为特征长度,决定粒子的搜索范围;u_{id}^{k+1}为(0, 1)内均匀分布的随机数;C_d^k为所有粒子最优位置的平均;β为收缩-扩张因子,控制算法的收敛速度. 文献[25]表明:当$0.5<\beta<0.8$时,算法能取得较为满意的结果;当$\beta=0.75$时,算法获得良好的性能. 然而,当β固定时,算法对粒子群规模和最大迭代次数都是敏感的. 如果采用时变的β,则算法性能将获得提高.

2 基于改进量子粒子群优化的K-means聚类算法

2.1 量子粒子群优化算法的改进与重构

量子粒子群优化算法能够以一定的概率出现在整个搜索空间中,具有良好的全局搜索能力. 但量子粒子群优化算法在迭代后期由于粒子的聚集性,种群多样性的损失不可避免,容易在后期陷入局部最优[26-28]. 为了进一步改善量子粒子群优化算法的寻优性能,本文对局部吸引子、特征长度、位置边界以及控制参数的选取进行重新设计,提出一种新的学习策略改进量子粒子群优化算法.

2.1.1 局部吸引子的设计

式(6)表明,粒子的局部吸引子收敛到其个体最优和全局最优的加权平均位置. 事实上,随着粒子收敛到自己的局部随机吸引子,它们的个体最优位置收敛到全局最优位置,量子粒子群优化算法收敛. 由此发现,全局最优位置引导局部随机吸引子的移动,影响粒子当前位置的收敛行为. 如果全局最优位置陷入局部最优,则局部随机吸引子、粒子当前位置都将陷入这一局部极值,这将使算法未成熟收敛. 为防止量子粒子群优化算法在迭代后期因粒子多样性不足引起的陷入局部最优,本文采用高斯分布确定局部随机吸引子,增强局部收敛能力以及跳出局部最优的能力,即

$$ga_i^k=[ga_{i1}^k,\ldots,ga_{id}^k,\ldots,ga_{iD}^k]=\mathrm{Normal}(a_i^k,C^k-p_i^k). \tag{11}$$

由此,获得粒子的位置更新如下:

$$x_{id}^{k+1}=ga_{id}^{k}\pm\beta|C_d^k-x_{id}^{k}|\ln\left(\frac{1}{u_{id}^{k+1}}\right). \tag{12}$$

2.1.2 特征长度的优化

所有粒子最优位置的平均可改写为

$$C_d^k=\frac{1}{N}\sum_{i=1}^{N}p_{id}^{k}=\sum_{i=1}^{N}\left(\frac{1}{N}p_{id}^{k}\right)=\sum_{i=1}^{N}(w_i^kp_{id}^{k}). \tag{13}$$

其中:w_i^k可视为在第k次迭代粒子x_i的权重,$w_i=1/N$. 显然,量子粒子群优化算法在计算种群最优位置中心时对各粒子取相同权重,并未考虑各粒子历史最优位置的适应度差异,难以发挥精英粒子优势. 为了提高量子粒子群优化算法的收敛性能,本文采用加权种群最优位置中心优化种群进化的方式,使精英粒

子在迭代过程中发挥的作用更大一些. 该方法可以有效降低落后粒子的干扰,提高种群搜索能力以加速收敛. 权重设置为粒子历史最优位置适应度占所有粒子历史最优位置适应度之和的比例,即

$$w_i^k = \frac{f(p_i^k)}{\sum_{i=1}^{N} f(p_i^k)}. \tag{14}$$

在量子粒子群优化算法中,一旦在位置边界有局部最优解,种群将很容易陷入局部最优. 随着边界处粒子的增多,种群多样性也会进一步降低. 本文对边界进行变异处理,可以有效克服粒子易在边界聚集,增强种群多样性. 变异处理方式为

$$x_{id} = \begin{cases} X_{\max} - (x_{id} - X_{\max}) \times \text{rand}, \ x_{id} > X_{\max}; \\ X_{\min} + (X_{\min} - x_{id}) \times \text{rand}\ , x_{id} < X_{\min}. \end{cases} \tag{15}$$

2.1.3 控制参数的选取

1) 收缩-扩张因子.

若采用固定的收缩-扩张因子,则算法的鲁棒性会降低. 通常采用自适应变化的收缩-扩张因子,可以在迭代后期改善算法局部搜索的精度. 本文选取3种递减策略用于对参数进行重组,分别为线性递减和非线性递减. 线性递减为

$$\beta = \beta_0 + (\beta_1 - \beta_0)\frac{K_{\max} - k}{K_{\max}}; \tag{16}$$

非线性递减(开口向上)为

$$\beta = (\beta_1 - \beta_0) \times \left(\frac{k}{K_{\max}}\right)^2 - (\beta_1 - \beta_0)\left(\frac{2k}{K_{\max}}\right) + \beta_1; \tag{17}$$

非线性递减(开口向下)为

$$\beta = (\beta_0 - \beta_1) \times \left(\frac{k}{K_{\max}}\right)^2 + \beta_1. \tag{18}$$

其中: $K_{\max}$ 为最大迭代次数; β_0, β_1 为预设值,一般取 $\beta_0 = 0.5, \beta_1 = 1$.

收缩-扩张因子随着迭代次数的变化关系如图1所示.

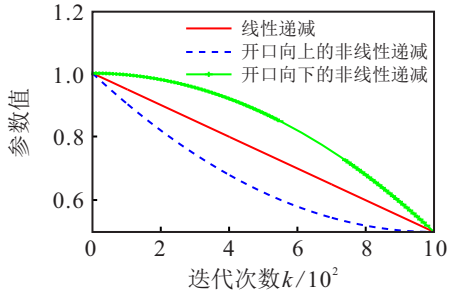


图 1 收缩-扩张因子变化曲线对比

2) 随机变量参数.

随机变量一般设置为 $\ln\left(\frac{1}{u}\right)$,本文采取另外两种变化方式,分别为 $\sqrt{\ln\left(\frac{1}{u}\right)}$ 和 $\frac{1}{\cos\sqrt{u}}$. 不同随机参数对应的函数图像如图2所示.

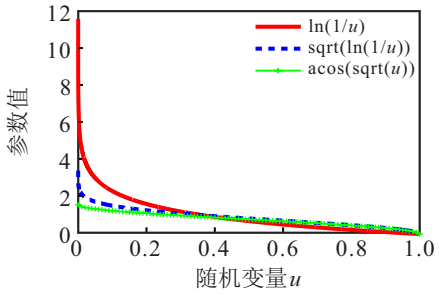


图 2 随机变量变化曲线对比

2.2 K-means聚类中心的确定

假设样本数据规模为 n,每一个数据对象有 p 个特征属性,聚类个数为 k,聚类中心为 $e_j (j = 1, 2, \ldots, k)$,共生成 N 个粒子. 每一个粒子的位置是由 k 个聚类中心组成,其位置编码结构为

$$\text{particle}(i) \cdot \text{location}[\,] = [\bar{x}_{e_1}, \bar{x}_{e_2}, \ldots, \bar{x}_{e_k}]. \tag{19}$$

其中: $\bar{x}_{e_j}$ 为第 j 类数据的聚类中心,是一个 p 维向量. 速度编码结构为

$$\text{particle}(i).\text{velocity}[\,] = [v_1, v_2, \ldots, v_k]. \tag{20}$$

每一个粒子的适应度是一个实数,即

$$f = a\text{SSE}, \tag{21}$$

$$\text{SSE} = \sum_{j=1}^{k} \sum_{o \in E_j} ||o - \bar{x}_{e_j}||. \tag{22}$$

其中: a 为正常数, E_j 为第 j 个类.

本文改进的量子粒子群聚类算法的目标是搜索到使粒子适应度最小时的粒子位置,全局最优位置即为聚类中心.

当全局最优解(聚类中心)确定后,聚类划分由最近邻法则确定,即每个数据对象优先划分到离它最近的类. 数据对象和聚类中心点满足

$$\|o, \bar{x}_{e_j}\| = \min_{j=1,2,\ldots,k} \|o, \bar{x}_{e_j}\|. \tag{23}$$

2.3 算法实施步骤

基本思想: 根据 K-means 算法的聚类原则,利用改进量子粒子群优化聚类中心,可以得到不同聚类数下的聚类划分.

通过采用具有高斯扰动的局部吸引子和加权更新种群平均最优位置等方法,提出一种改进的量子粒子群优化算法(improved quantum-behaved particle swarm optimization, IQPSO). 由于IQPSO算法是一

种随机寻优算法,不受初始解的干扰,且具有较强的全局搜索能力和较快的收敛速度,整个搜索空间内得到使聚类目标函数尽可能小的聚类中心,可以有效地避免K-means聚类算法对初始中心的依赖,提高算法的聚类精度. 将搜索到的全局最优位置(即聚类中心)用作K-means聚类,对整个数据集进行聚类划分时,按照样本与聚类中心欧氏距离最近的原则进行划分. 本文设计IQPSO-K-means (IQPSO-KM)聚类算法实现的具体步骤如下.

step 1: 输入参数:样本数据个数n,样本特征维度p,聚类个数k.

step 2: 初始化粒子群:随机选取k个数据作为一个粒子的初始位置,重复该过程N次,共生成N个粒子;计算初始粒子编码. 包含的参数主要有:群体规模N、维度D、最大迭代次数$T_{\max}$、最大位置边界$X_{\max}$、最小位置边界$X_{\min}$、所有粒子在D维空间中的随机位置等参数.

step 3: 计算所有粒子个体最优位置的平均值和局部吸引子,设计种群收缩-扩张因子和随机变量.

step 4: 计算粒子x_i在D维空间中当前适应度$f(x_i)$.

step 5: 更新个体最优和全局最优:分别比较当前粒子适应度$f(x_i)$与粒子个体最优适应度$f(p_i)$、群体最优适应度$f(p_g)$,分别更新个体和群体最优位置、最优值.

step 6: 更新所有粒子位置并做边界变异处理.

step 7: 检验是否符合结束条件:若不能满足退出条件(未达到最大迭代次数),则返回step 3进行下一次迭代;否则转到step 8,输出全局最优适应度和全局最优位置(最终聚类中心).

step 8: 求出聚类结果:对于每一个数据对象,计算与最终聚类中心的距离,按照最近邻法则确定该数据对象的聚类划分,最后把所有样本数据都分配到k个聚类中心,完成聚类.

3　数值仿真与结果分析

3.1　改进量子粒子群优化算法的性能分析

为了验证IQPSO算法的有效性,本文采用优化问题为0的30维测试函数进行实验. 优化函数包含有许多局部极小值(Rastrigin function (f_1), Griewank function (f_2), Ackley function (f_3))、碗状(Sphere function (f_4), Sum squares function (f_5), Rotated hyper-ellipsoid function (f_6))、山谷状(Rosenbrock function (f_7), Dixon-price function (f_8))和板状(Zakharov function (f_9))函数.

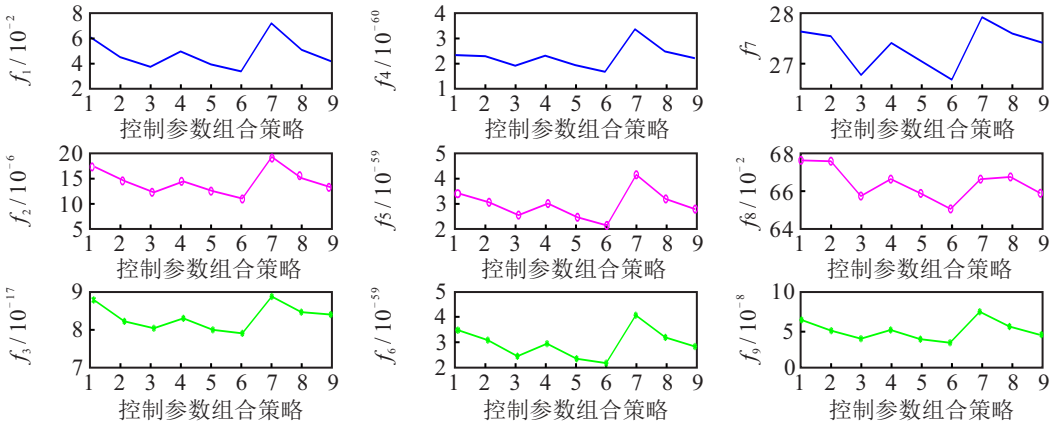
实验中,所有变量维数设为30,种群大小设为400,惯性权重设为1,学习因子设为2,位置范围设为$[-10, 10]$,迭代次数设为1 000.

3.1.1　控制参数组合策略分析

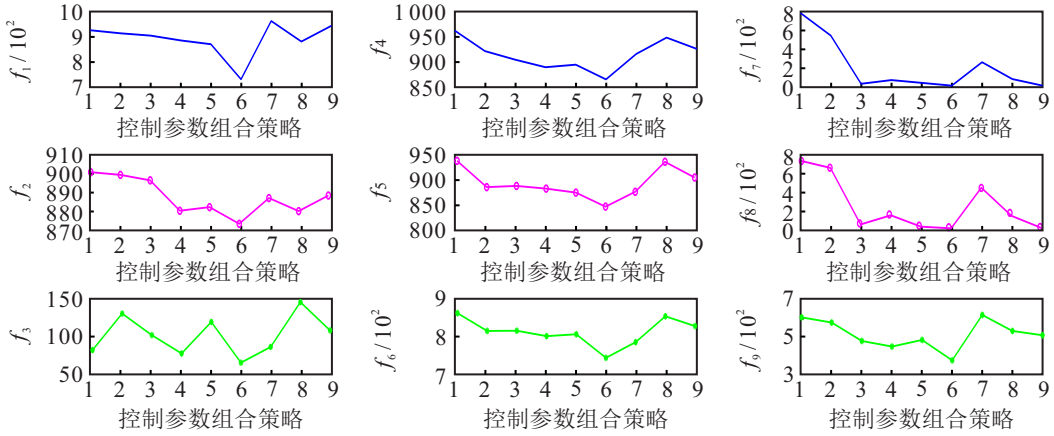
在基于改进局部吸引子和特征长度的量子粒子群优化算法基础上组合以下控制参数(表1)作为最终改进方法的输入参数. 每种方法独立运行30次,对所有测试函数的最优适应度、收敛次数、运行时间、标准差的均值结果进行统计,如图3所示.

表 1　参数组合策略

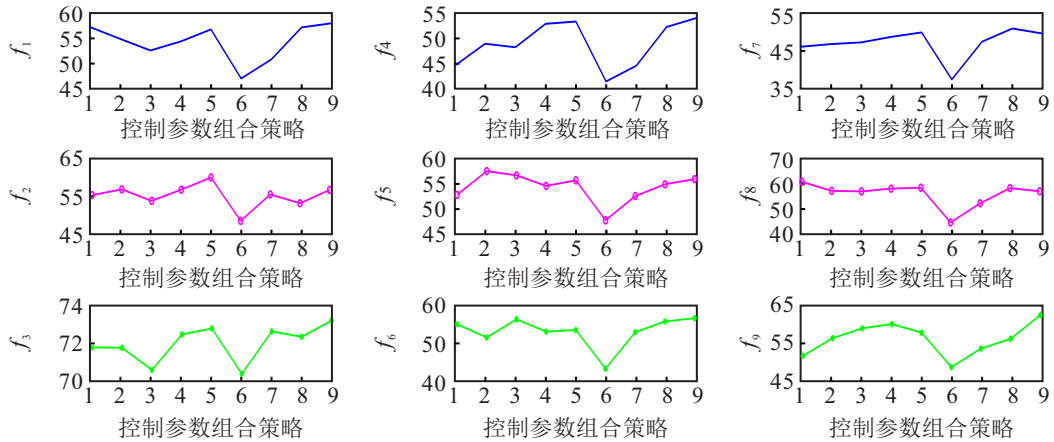
改进方法	收缩-扩张因子	随机变量
IQPSO_1	$\beta_0+(\beta_1-\beta_0)\dfrac{K_{\max}-k}{K_{\max}}$	$\ln\left(\dfrac{1}{u}\right)$
IQPSO_2	$\beta_0+(\beta_1-\beta_0)\dfrac{K_{\max}-k}{K_{\max}}$	$\sqrt{\ln\left(\dfrac{1}{u}\right)}$
IQPSO_3	$\beta_0+(\beta_1-\beta_0)\dfrac{K_{\max}-k}{K_{\max}}$	$\dfrac{1}{\cos\sqrt{u}}$
IQPSO_4	$(\beta_1-\beta_0)\left(\dfrac{k}{K_{\max}}\right)^2-(\beta_1-\beta_0)\left(\dfrac{2k}{K_{\max}}\right)+\beta_1$	$\ln\left(\dfrac{1}{u}\right)$
IQPSO_5	$(\beta_1-\beta_0)\left(\dfrac{k}{K_{\max}}\right)^2-(\beta_1-\beta_0)\left(\dfrac{2k}{K_{\max}}\right)+\beta_1$	$\sqrt{\ln\left(\dfrac{1}{u}\right)}$
IQPSO_6	$(\beta_1-\beta_0)\left(\dfrac{k}{K_{\max}}\right)^2-(\beta_1-\beta_0)\left(\dfrac{2k}{K_{\max}}\right)+\beta_1$	$\dfrac{1}{\cos\sqrt{u}}$
IQPSO_7	$(\beta_0-\beta_1)\left(\dfrac{k}{K_{\max}}\right)^2+\beta_1$	$\ln\left(\dfrac{1}{u}\right)$
IQPSO_8	$(\beta_0-\beta_1)\left(\dfrac{k}{K_{\max}}\right)^2+\beta_1$	$\sqrt{\ln\left(\dfrac{1}{u}\right)}$
IQPSO_9	$(\beta_0-\beta_1)\left(\dfrac{k}{K_{\max}}\right)^2+\beta_1$	$\dfrac{1}{\cos\sqrt{u}}$



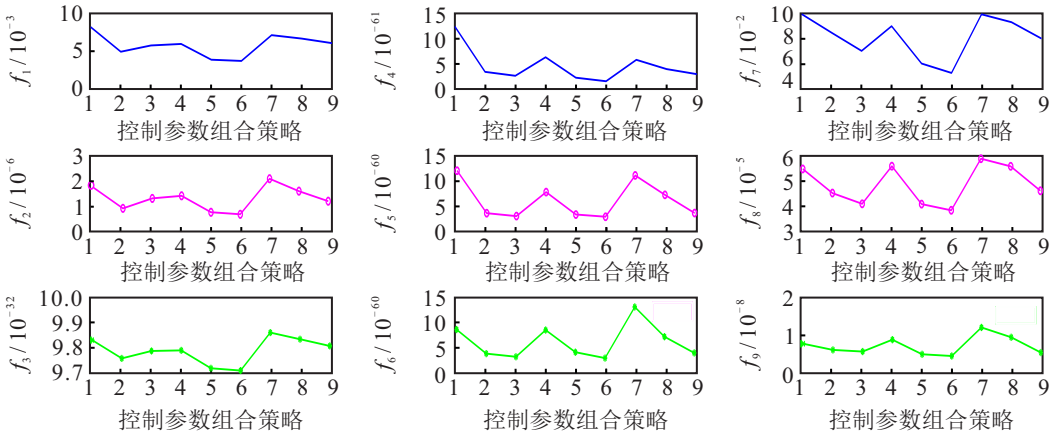
(a) 平均最优适应度



(b) 平均收敛次数



(c) 平均运行时间



(d) 平均标准差

图 3 不同参数组合策略对应的评价指标

由图3可知,从平均最优适应度、收敛次数、运行时间、标准差4个方面看,有许多局部极小值、碗状、山谷状和板状类型的函数的变化趋势基本一致. 从总体上看,第6种组合策略的指标均较低,因此,采取第6种参数组合策略既保证了算法的寻优精度,也保证了收敛速度.

3.1.2　改进算法的寻优性能分析

为了分析IQPSO算法的寻优性能,将其与粒子群优化算法particle swarm optimization, PSO)和量子粒子群优化算法(quantum-behaved particle swarm optimization, QPSO)进行30次独立试验,统计结果如表2所示(加粗部分表示最优值).

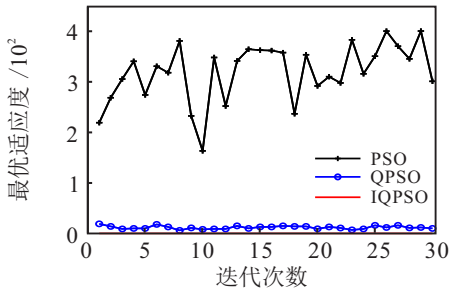
表 2　不同优化算法在所有测试函数上的寻优结果

函数	方法	最优适应度的平均值	最优适应度的最大值	最优适应度的最小值	最优适应度的中位数	平均收敛次数	平均运行时间/s	平均成功率	平均标准差
	PSO	3.201e+02	4.015e+02	1.637e+02	3.365e+02	**664.533**	63.824	1.000	5.651e+01
f_1	QPSO	1.183e+01	1.890e+01	6.070e+00	1.146e+01	994.400	59.606	0.967	3.087e+00
	IQPSO	**3.375e-02**	**3.931e-02**	**2.557e-02**	**3.408e-02**	730.900	**46.976**	**1.000**	**3.618e-03**
	PSO	8.260e-01	1.011e+00	6.030e-01	8.410e-01	879.033	66.866	1.000	1.010e-01
f_2	QPSO	2.349e-02	6.893e-02	0.000e+00	2.213e-02	904.700	59.873	1.000	2.136e-02
	IQPSO	**7.219e-06**	**9.019e-06**	**5.066e-06**	**7.302e-06**	**873.800**	**48.544**	**1.000**	**7.057e-07**
	PSO	6.894e+00	8.335e+00	5.049e+00	6.898e+00	842.067	76.388	1.000	8.140e-01
f_3	QPSO	1.344e-14	2.220e-14	7.994e-15	1.510e-14	185.900	73.786	1.000	4.276e-15
	IQPSO	**8.794e-16**	**8.810e-16**	**8.789e-16**	**8.791e-16**	**66.200**	**70.392**	**1.000**	**9.707e-32**
	PSO	4.786e+01	8.431e+01	1.313e+01	4.092e+01	940.800	57.707	1.000	1.744e+01
f_4	QPSO	2.005e-54	1.379e-53	2.245e-56	8.495e-55	962.234	52.032	1.000	3.167e-54
	IQPSO	**1.673e-60**	**1.991e-60**	**9.484e-61**	**1.672e-60**	**863.300**	**41.422**	**1.000**	**1.202e-61**
	PSO	6.617e+02	1.409e+03	1.671e+02	5.869e+02	928.467	63.652	1.000	3.247e+02
f_5	QPSO	2.003e-53	1.319e-52	1.682e-55	7.616e-54	946.086	58.494	1.000	3.102e-53
	IQPSO	**2.135e-59**	**2.804e-59**	**9.863e-60**	**2.173e-59**	**844.700**	**47.750**	**1.000**	**2.854e-60**
	PSO	7.566e+02	1.555e+03	3.648e+02	6.570e+02	850.433	69.604	0.933	2.803e+02
f_6	QPSO	2.963e-53	1.549e-52	4.110e-55	1.282e-53	880.321	65.103	0.967	4.080e-53
	IQPSO	**2.156e-59**	**2.713e-59**	**1.380e-59**	**2.191e-59**	**745.900**	**43.821**	**1.000**	**2.917e-60**
	PSO	5.374e+04	1.049e+05	1.770e+04	5.204e+04	767.033	55.874	1.000	2.457e+04
f_7	QPSO	2.869e+01	2.876e+01	2.830e+01	2.872e+01	886.400	50.954	1.000	1.015e-01
	IQPSO	**2.668e+01**	**2.779e+01**	**2.653e+01**	**2.679e+01**	**28.533**	**37.431**	**1.000**	**5.274e-02**
	PSO	1.472e+04	5.404e+04	9.708e+02	1.094e+04	725.067	65.191	0.933	1.115e+04
f_8	QPSO	9.667e-01	9.668e-01	9.667e-01	9.667e-01	791.800	62.673	1.000	8.427e-05
	IQPSO	**6.510e-01**	**8.730e-01**	**6.275e-01**	**6.440e-01**	**26.000**	**44.292**	**1.000**	**3.844e-05**
	PSO	1.991e+02	2.781e+02	1.152e+02	1.948e+02	380.367	60.884	1.000	3.411e+01
f_9	QPSO	2.628e-03	7.081e-03	7.347e-04	2.274e-03	699.333	57.214	0.433	1.698e-03
	IQPSO	**3.373e-08**	**3.979e-08**	**1.788e-08**	**3.441e-08**	**378.600**	**48.705**	**1.000**	**4.678e-09**

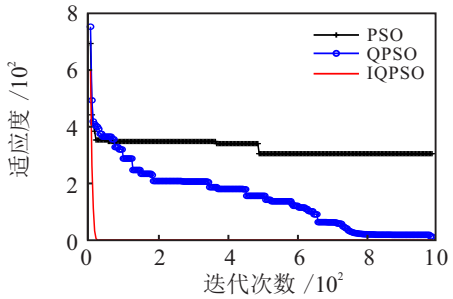
由表2数据可知: 1)为比较不同算法的寻优精度,主要分析最优适应度的平均值、最大值、最小值以及中位数. 对于所有测试函数,IQPSO算法的4个指标均为最小,与目标函数最优解最为接近;IQPSO算法比PSO算法显著提高了寻优精度,这是因为局部吸引子的设计使算法不容易陷入局部最优,增强了种群多样性;QPSO算法相较于PSO算法,具有更好的全局搜索能力. 2)为比较不同算法的收敛速度,主要分析平均收敛次数和平均运行时间. 由于对IQPSO算法的特征长度进行优化,发挥了精英粒子的作用,它的收敛速度最快;PSO算法在收敛次数上优于QPSO算法,这是因为PSO算法利用粒子在解空间中的随机速度改变粒子位置,有更强的随机性. 在算法运行时间方面,IQPSO算法执行最快;相较PSO算法,QPSO算法没有速度项,运行时间较快. 3)为比较不同算法的稳定性,主要分析平均成功率和平均标准

差. IQPSO 算法在所有测试函数上均能够成功收敛, 平均标准差变化趋势与最优适应度平均值相同.

以函数 f_1 为例, 图 4 绘制了不同优化算法运行 30 次最优适应度和某次迭代过程中不同优化算法的适应度变化曲线.



(a) 运行 30 次最优适应度



(b) 某一次迭代适应度

图 4 f_1 函数适应度变化过程

从图 4(a) 中可以看出, IQPSO 算法的最优适应度曲线远比 PSO 算法寻优曲线平滑, 说明 IQPSO 的算法稳定性更好, 寻优精度更明显. 从图 4(b) 中可以看出, IQPSO 算法引入了具有高斯扰动的局部吸引子和粒子加权的特征长度, 可以有效达到全局最优. 综上分析, IQPSO 算法在寻优精度、收敛速度、寻优稳定性上效果均显著提高, 充分说明了 IQPSO 算法的有效性和可行性.

3.2 改进 K-means 算法的聚类结果分析

为了评估 IQPSO-KM 的性能, 与传统 K-means 算法、K-means++ 算法、K-medoids 算法、基于粒子群的 K-means 聚类算法 (PSO-K-means, PSO-KM)、基于量子粒子群的 K-means 聚类算法 (QPSO-K-means, QPSO-KM) 以及文献 [16] 算法进行对比. 采用 3 组典型 UCI 数据集作为实验数据, 考察不同聚类算法的准确率 accuracy、精确率 precision、召回率 recall、F_1 值、轮廓系数 (silhouette coefficient, SC) 以及适应度 fitness. 不同聚类算法的评估指标统计结果如表 3 所示, 直观统计如图 5 所示 (fitness 指标中, 由于 K-means、K-means++、K-medoids 数量级与其他算法相差较大, 绘图仅与 PSO-KM、QPSO-KM 以及文献 [16] 算法进行比较).

表 3 不同聚类算法在典型 UCI 数据集上的聚类结果

数据集	方法	accuracy / %	precision / %	recall / %	F_1	SC	fitness
Iris	K-means	84.666 7	72.820 9	72.449 0	0.781 1	0.678 3	97.325 9
	K-means++	86.678 3	75.374 9	73.072 6	0.814 6	0.702 3	94.534 5
	K-medoids	86.893 3	74.692 6	70.809 5	0.811 5	0.720 1	96.234 5
	文献 [16] 算法	**93.334 2**	79.526 4	83.679 0	**0.847 7**	**0.824 5**	26.326 8
	PSO-KM	88.666 7	77.983 4	80.591 8	0.822 1	0.770 0	33.495 1
	QPSO-KM	88.036 2	80.897 3	82.654 9	0.823 6	0.794 3	31.279 0
	IQPSO-KM	90.763 2	**82.217 5**	**87.952 4**	0.835 4	0.803 3	**25.302 5**
Wine	K-means	65.3393	47.519 0	67.130 0	0.545 6	0.615 2	18 436.952 1
	K-means++	67.989 2	50.845 7	69.463 2	0.597 6	0.660 8	17 558.246 9
	K-medoids	70.223 1	50.390 8	68.324 7	0.583 5	0.655 6	16 701.456 6
	文献 [16] 算法	72.544 9	55.520 3	**79.134 2**	0.635 6	**0.735 2**	96.756 6
	PSO-KM	68.427 0	52.318 6	74.173 6	0.599 6	0.704 5	105.662 5
	QPSO-KM	73.963 1	52.856 9	76.385 4	0.624 3	0.734 1	102.183 2
	IQPSO-KM	**77.755 1**	**55.774 3**	78.980 9	**0.709 8**	0.725 9	**91.595 3**
Haberman	K-means	66.529 4	55.760 2	50.302 4	0.550 4	0.499 1	2 626.410 4
	K-means++	68.534 8	60.870 9	52.711 3	0.565 1	0.595 4	2 586.657 8
	K-medoids	70.538 1	60.987 7	51.324 6	0.557 3	0.527 3	2 599.995 0
	文献 [16] 算法	74.549 1	**67.704 6**	64.410 9	0.626 2	0.650 4	96.875 4
	PSO-KM	72.817 0	63.675 6	62.271 4	0.603 5	0.626 6	104.155 3
	QPSO-KM	73.867 4	65.697 5	66.341 0	0.648 7	0.642 9	100.034 5
	IQPSO-KM	**76.659 7**	66.659 7	**70.372 7**	**0.660 4**	**0.672 8**	**80.583 8**

注: 若聚类算法没有适应度, 则表 3 中适应度值表示为类内误差平方和.

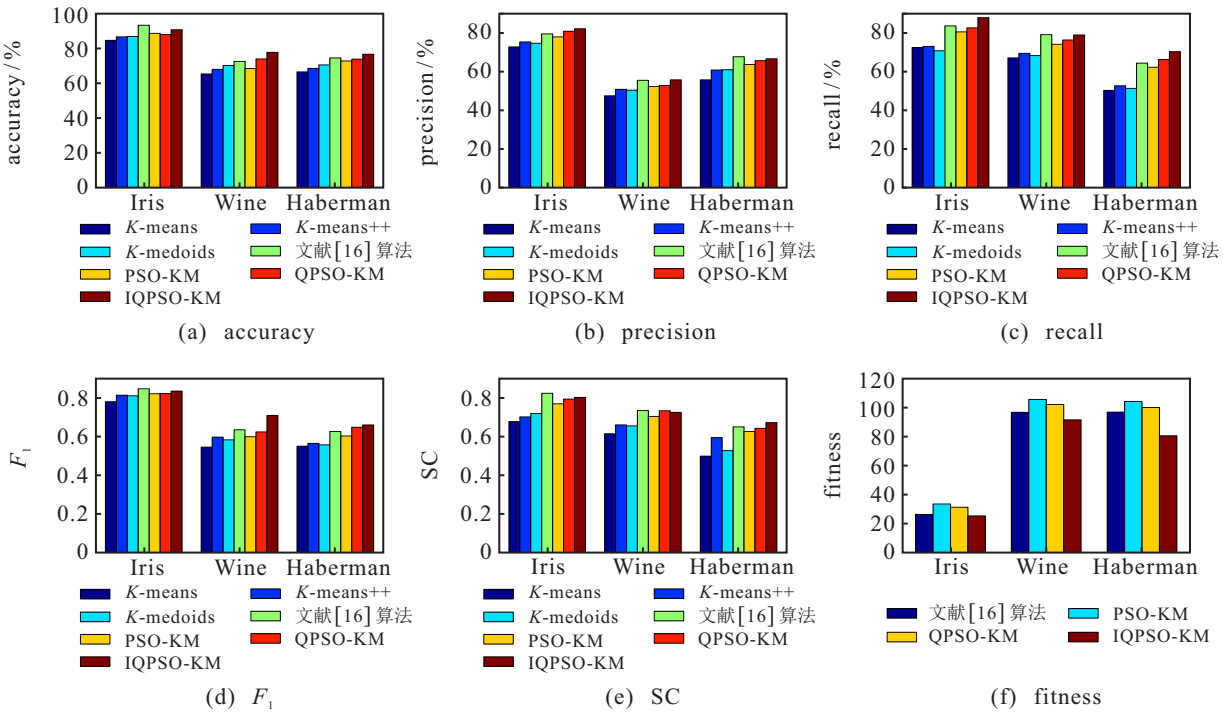


图5　不同聚类算法的聚类性能直观比较

从表3和图5中可知: K-means聚类算法的聚类准确度最低,这是由K-means聚类算法对初始类中心的过度依赖造成的; K-means++和K-medoids聚类算法在处理传统K-means聚类中存在的初始中心点不稳定的问题上均可以得到较好的改善;文献[16]算法采用改进的蚁群算法优化K-means聚类中心的位置,较K-means相关传统改进算法有显著改善,验证采用群智能优化改进聚类算法具有一定的提高; PSO-KM和QPSO-KM聚类算法利用了PSO和QPSO的全局搜索能力,下一代种群具有较大的随机性,也利用了K-means较强的局部搜索能力,在粒子附近又进行了一次精确的局部搜索,所以不容易陷入局部极值; IQPSO-KM较QPSO-KM聚类算法的聚类性能有所提升,这是因为IQPSO-KM聚类算法可以有效地预防早熟收敛现象,使粒子获得重新搜索全局最优解的机会.

3.3　基于改进量子粒子群聚类的葡萄客户分类

在竞争日益激烈的市场环境下,产品竞争力提升依靠突出的产品优势和高度与细分客户相匹配的能力. 客户分类是根据客户的购买需求、价值观等属性,利用相关技术将客户划分为不同客户群体的过程,从而预测不同客户群体的消费行为,为客户提供个性化的服务. 数据挖掘技术中,聚类分析广泛地应用于客户分类领域[29-30]. 最为经典的是基于划分的K-means算法,目前该算法已成为数据挖掘领域中最常用的聚类算法. 为了验证本文提出的改进K-means聚类算法的有效性,将IQPSO-KM算法应用于我国鲜食葡萄市场客户分类中,以达到客户精准分类的目的.

3.3.1　样本数据获取

本文所用真实数据来自2017年国家葡萄产业技术体系在全国进行的客户调查,以鲜食葡萄客户为调研对象,对客户消费价值观做实证分析. 充分考虑样本的分散性和随机性,调查对象涉及到不同的性别、年龄、教育程度、职业、家庭规模、家庭人均月收入以及城市等级. 问卷最终回收3 652份,其中有效样本量为3 230份(占88.44 %). 客户消费价值观特征属性箱线图如图6所示,从图6可以看出变量b、f、g、h存在一定的离群值. 本文数据并未超出正常范围,且逻辑合理,对于个别具有异常值数据的变量暂不做处理.

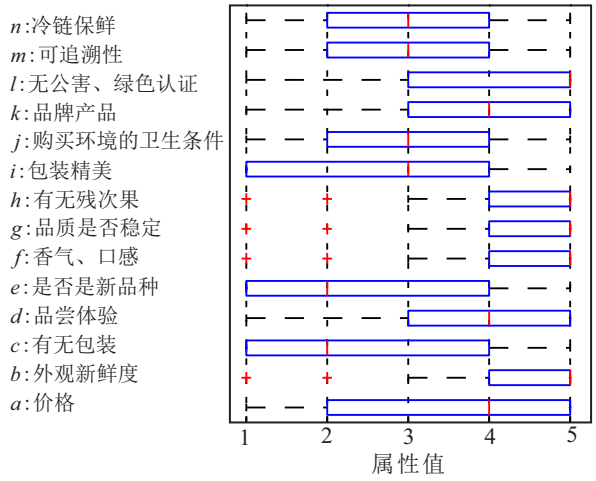


图6　客户消费价值观属性的箱线图

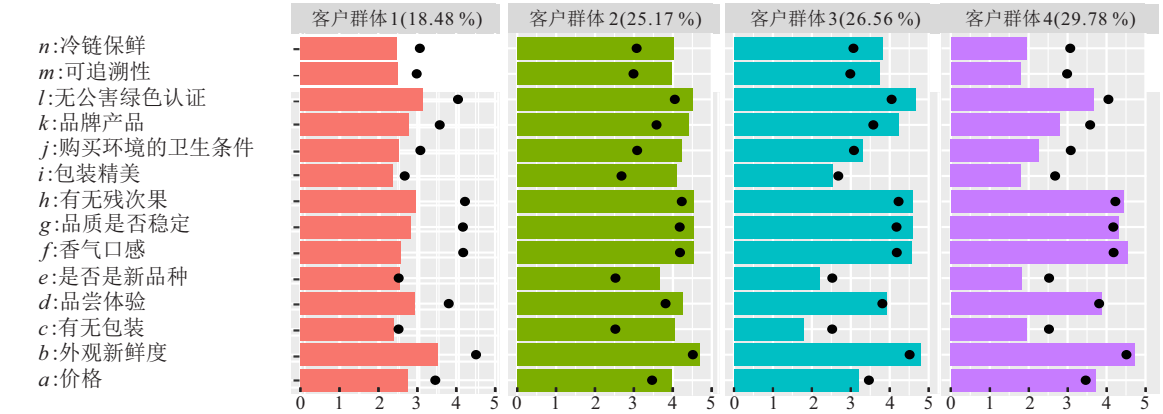
3.3.2 客户分类效果分析

为了确定本文真实葡萄数据集中的客户分类个数,在不同聚类数下利用IQPSO-KM算法对真实数据集进行聚类,并比较最优适应度(表4). 结果显示:当聚类数为4时,最优适应度最小,故将客户群体划分为4类.

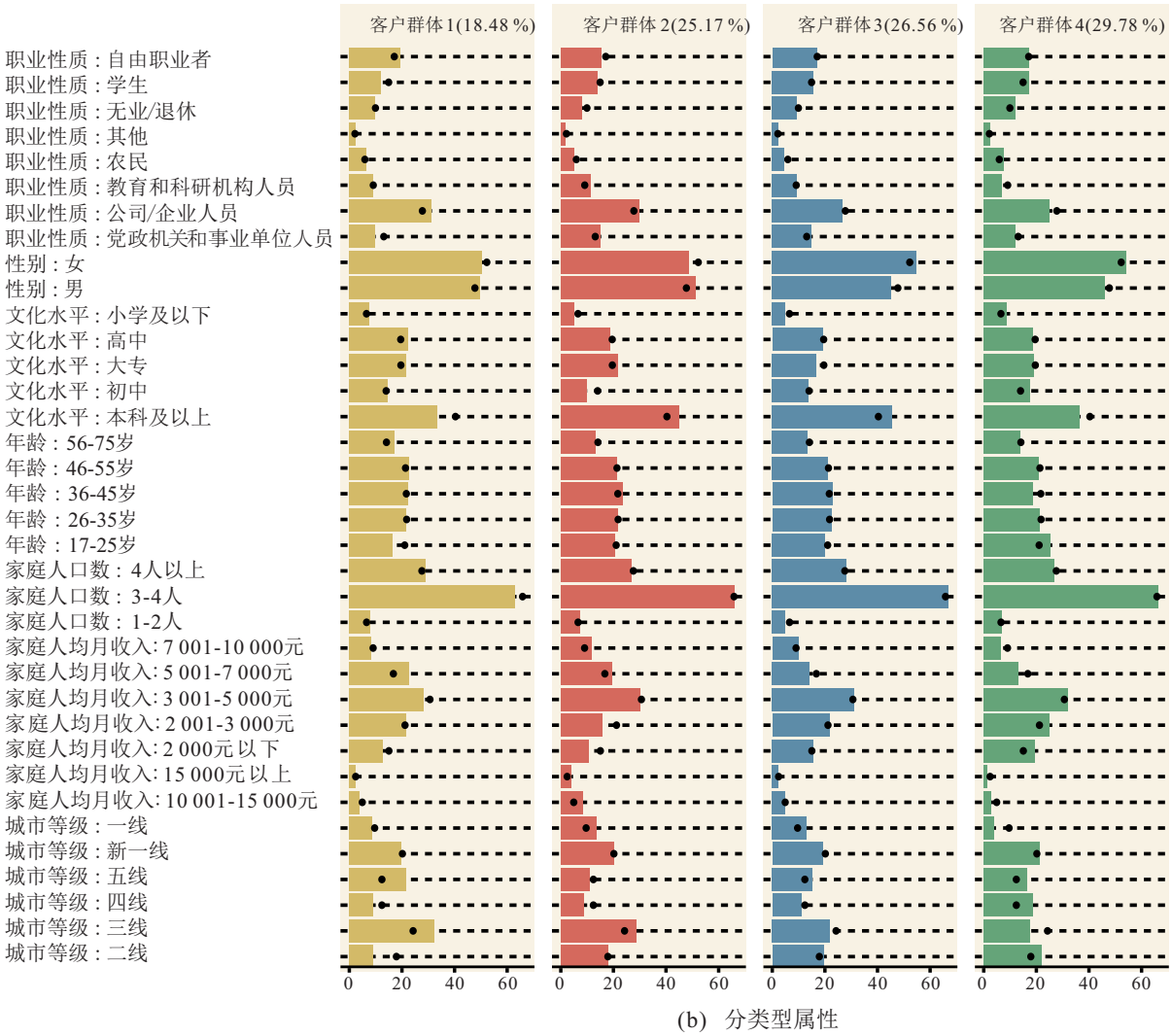
为了更好地描述和总结不同客户群体的消费特征,本研究计算每个客户群体中各数值变量的均值和分类变量的频率,并将其与我国葡萄市场消费水平进行比较(图7).

表4 不同聚类个数对应的目标函数值

聚类个数	聚类结果	聚类个数	聚类结果
2	3 271.333	7	3 154.515
3	3 212.745	8	3 142.956
4	**3 125.385**	9	3 170.892
5	3 151.467	10	3 192.732
6	3 147.576		



(a) 数值型属性



(b) 分类型属性

图 7 葡萄客户分类效果

在图7中,“•”表示我国葡萄市场的平均消费水平. 通过对聚类结果的可视化分析可以清晰地判断出不同类别的客户群体具有不同的消费行为. 因此,验证了本文提出的IQPSO-KM聚类算法具有一定的可行性和有效性,划分结果有助于零售企业对不同的客户群体制定更加合理有效的营销策略,从而更好地进行客户关系管理,以提高企业效益.

4 结　论

1) 本文提出了一种改进的量子粒子群优化算法,采用高斯扰动的局部吸引子,保证了种群多样性,有效防止算法出现早熟收敛;为每个粒子引入权重以改进特征长度,增强粒子挣脱局部极值约束的能力,提高了算法收敛速度;对收缩-扩张因子和随机变量参数进行交叉组合,选择出最佳参数. 通过对不同类型的寻优函数进行仿真可知,IQPSO算法较PSO算法和QPSO算法在寻优精度、收敛速度以及搜索可靠性上都显著提高,充分说明了改进算法的可行性.

2) 将改进的量子粒子群优化算法与K-means算法结合,解决了K-means聚类算法对初始中心点的依赖问题. 在典型UCI数据集上的仿真实验表明,改进的IQPSO-KM算法的聚类准确性显著优于传统K-means聚类算法,使其脱离局部极值获得近似全局最优的聚类划分.

3) 将IQPSO-KM聚类算法应用于我国鲜食葡萄市场客户分类中,验证了本文方法的有效性和实用性. 本文方法为葡萄客户分类提供了积极思路,分类结果可以帮助零售企业为不同客户群体制定个性化营销策略,也为提升产品与客户的需求匹配度提供了重要参考.

参考文献(References)

[1] Jiang X P, Li C H, Sun J. A modified K-means clustering for mining of multimedia databases based on dimensionality reduction and similarity measures[J]. Cluster Computing, 2017, 20(10): 1-8.

[2] 杨华晖, 孟晨, 王成, 等. 基于目标特征选择和去除的改进K-means聚类算法[J]. 控制与决策, 2019, 34(6): 1219-1226.
(Yang H H, Meng C, Wang C, et al. Improved K-means clustering algorithm based on feature selection and removal on target point[J]. Control and Decision, 2019, 34(6): 1219-1226.)

[3] Berkhin P. A survey of clustering data mining techniques[J]. Grouping Multidimensional Data. Berlin: Springer-Verlay, 2006: 25-71.

[4] Chen G C, Liu Y, Ge Z Q. K-means bayes algorithm for imbalanced fault classification and big data application[J]. Journal of Process Control, 2019, 81: 54-64.

[5] Luo F L. An improved K-means algorithm and its application in customer classification of network enterprises[J]. Applied Mechanics and Materials, 2014, 3082(1090): 2124-2127.

[6] 王骏, 王士同, 邓赵红. 聚类分析研究中的若干问题[J]. 控制与决策, 2012, 27(3): 321-328.
(Wang J, Wang S T, Deng Z H. Survey on challenges in clustering analysis research[J]. Control and Decision, 2012, 27(3): 321-328.)

[7] Bai L, Cheng X Q, Liang J Y, et al. Fast density clustering strategies based on the K-means algorithm[J]. Pattern Recognition, 2017, 71(3): 375-386.

[8] 周本金, 陶以政, 纪斌, 等. 最小化误差平方和K-means初始聚类中心优化方法[J]. 计算机工程与应用, 2018, 54(15): 48-52.
(Zhou B J, Tao Y Z, Ji B, et al. Optimizing K-means initial clustering centers by minimizing sum of squared error[J]. Computer Engineering and Applications, 2018, 54(15): 48-52.)

[9] 余小高, 余小鹏. 基于距离和密度的无监督聚类算法的研究[J]. 计算机应用与软件, 2010, 27(7): 122-125.
(Yu X G, Yu X P. On unsupervised clustering algorithm based on distance and density[J]. Computer Applications and Software, 2010, 27(7): 122-125.)

[10] 邓滨玥. K均值优化算法综述[J]. 软件, 2020, 41(2): 188-192.
(Deng B Y. A survey on advanced K-means algorithm[J]. Computer Engineering & Software, 2020, 41(2): 188-192.)

[11] 刘叶, 吴晟, 周海河, 等. 基于K-means聚类算法优化方法的研究[J]. 信息技术, 2019, 43(1): 66-70.
(Liu Y, Wu S, Zhou H H, et al. Research on optimization method based on K-means clustering algorithm[J]. Information Technology, 2019, 43(01): 66-70.)

[12] Park H S, Jun C H. A simple and fast algorithm for K-medoids clustering[J]. Expert Systems with Applications, 2009, 36(2): 3336-3341.

[13] Kuo R J, Mei C H, Zulvia F E, et al. An application of a metaheuristic algorithm-based clustering ensemble method to APP customer segmentation[J]. Neurocomputing, 2016, 205: 116-129.

[14] 陈小雪, 尉永清, 任敏, 等. 基于萤火虫优化的加权K-means算法[J]. 计算机应用研究, 2018, 35(2): 466-470.
(Chen X X, Wei Y Q, Ren M, et al. Weighted K-means clustering algorithm based on firefly algorithm[J]. Application Research of Computers, 2018, 35(2):

466-470.)

[15] 陶新民, 徐晶, 杨立标, 等. 一种改进的粒子群和K均值混合聚类算法[J]. 电子与信息学报, 2010, 32(1): 92- 97.
(Tao X M, Xu J, Yang L B, et al. Improved cluster algorithm based on K-means and particle swarm optimization[J]. Journal of Electronics & Information Technology, 2010, 32(1): 92-97.)

[16] 于佐军, 秦欢. 基于改进蜂群算法的K-means算法[J]. 控制与决策, 2018, 33(1): 181-185.
(Yu Z J, Qin H. K-means algorithm based on improved artificial bee colony algorithm[J]. Control and Decision, 2018, 33(1): 181-185.)

[17] 张宏立, 李瑞国, 范文慧, 等. 基于量子粒子群的全参数连分式混沌时间序列预测[J]. 控制与决策, 2016, 31(1): 52-58.
(Zhang H L, Li R G, Fan W H, et al. Chaotic time series prediction of full-parameters continued fraction based on quantum particle swarm optimization algorithm[J]. Control and Decision, 2016, 31(1): 52-58.)

[18] 张强, 李盼池. 一种自适应多策略行为粒子群优化算法[J]. 控制与决策, 2020, 35(1): 115-122.
(Zhang Q, Li P C. An adaptive multi-strategy behavior particle swarm optimization algorithm[J]. Control and Decision, 2020, 35(1): 115-122.)

[19] 王皓, 欧阳海滨, 高立群. 一种改进的全局粒子群优化算法[J]. 控制与决策, 2016, 31(7): 1161-1168.
(Wang H, Ouyang H B, Gao L Q. An improved global particle swarm optimization[J]. Control and Decision, 2016, 31(7): 1161-1168.)

[20] Sun J, Xu W B, Feng B. A global search strategy of quantum-behaved particle swarm optimization[C]. Conference on Cybernetics and Intelligent Systems. Singapore, 2004: 111-116.

[21] Guan X, Liu J, Huang Q R, et al. Assessing the freshness of meat by using quantum-behaved particle swarm optimization and support vector machine[J]. Journal of Food Protection, 2013, 76(11): 1916-1922.

[22] Coelho L D S. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems[J]. Expert Systems with Applications, 2010, 37(2): 1676-1683.

[23] Xi M L, Sun J, Xu W B. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position[J]. Applied Mathematics and Computation, 2008, 205(5): 751-759.

[24] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73.

[25] Sun J, Fang W, Wu X J, et al. Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection[J]. Evolutionary Computation, 2012, 20(3): 349-393.

[26] 彭越兮, 徐蔚鸿, 陈沅涛, 等. 改进量子粒子群算法的模糊神经网络水质评价[J]. 计算机工程与应用, 2018, 54(11): 211-216.
(Peng Y X, Xu W H, Chen Y T, et al. Improved quantum-behaved particle swarm optimization training fuzzy neural network used in water quality evaluation[J]. Computer Engineering and Applications, 2018, 54(11): 211-216.)

[27] 施展, 陈庆伟. 基于QPSO和拥挤距离排序的多目标量子粒子群优化算法[J]. 控制与决策, 2011, 26(4): 540-547.
(Shi Z, Chen Q W. Multi-objective quantum-behaved particle swarm optimization algorithm based on QPSO and crowding distance sorting[J]. Control and Decision, 2011, 26(4): 540-547.)

[28] 田瑾. 高维多峰函数的量子行为粒子群优化算法改进研究[J]. 控制与决策, 2016, 31(11): 1967-1972.
(Tian J. An improvement of quantum-behaved particle swarm optimization algorithm for high-dimensional and multi-modal functions[J]. Control and Decision, 2016, 31(11): 1967-1972.)

[29] Vijaya J, Sivasankar E. Computing efficient features using rough set theory combined with ensemble classification techniques to improve the customer churn prediction in telecommunication sector[J]. Computing, 2018, 100(8): 839-860.

[30] Holy V, Sokol O, Cerny M. Clustering retail products based on customer behaviour[J]. Applied Soft Computing, 2017, 60(2): 752-762.

作者简介

李玥(1994–), 女, 博士生, 从事数据挖掘、机器学习的研究, E-mail: 18800146617@163.com;

穆维松(1967–), 女, 教授, 博士生导师, 从事农业产业信息管理与智能处理等研究, E-mail: wsmu@cau.edu.cn;

褚晓泉(1994–), 女, 博士生, 从事数据科学与智能系统的研究, E-mail: chuxiaoquan1994@163.com;

傅泽田(1956–), 男, 教授, 博士生导师, 从事信息管理与信息系统、物流与供应链管理等研究, fzt@cau.edu.cn.

(责任编辑: 闫 妍)