基于多类合并的 PSO-means 聚类算法①

林有城, 符 强, 谢文斌, 史马杰, 童 楠

(宁波大学科技学院, 宁波 315212)

摘 要: 针对传统 K-means 算法中对初始化聚类中心敏感, 容易陷入局部极小值等缺点, 提出了一种基于粒子群算法和多类合并方法的新型 K-means 聚类算法. 该算法首先利用改进粒子群算法选取初始聚类中心, 然后利用 K-means 算法进行优化聚类, 最后根据多类合并条件进行聚类合并, 以获取最佳聚类结果. 实验结果证明, 该算法能有效解决传统 K-means 算法存在的缺陷, 具有更快的收敛速度及更好的全局搜索能力, 聚类划分效果更优.

关键词: 粒子群算法; 多类合并; K-means 算法; 适应度方差

K-means Optimization Clustering Algorithm Based on Particle Swarm Optimization and Multi-Groups Merging

LIN You-Cheng, FU Qiang, XIE Wen-Bin, SHI Ma-Jie, TONG Nan

(College of Science and Technology, Ningbo University, Ningbo 315212, China)

Abstract: To deal with the problem of the sensitivity of initialization and premature convergence, this paper proposes a novel K-means optimization clustering algorithm based on particle swarm optimization and multi-groups merging, namely M-PSO-Means. Firstly the algorithm selects the initial cluster center by improving particle swarms clustering algorithm under default number of clustering, then optimizes the clustering, and last carries out cluster merging based on multi-groups merging condition to obtain the best clustering results. The experimental results show that, the algorithm can effectively solve the defects of K-means algorithm, and has a faster convergence rate and better global search ability, as well as better cluster category effect.

Key words: particle swarm optimization(PSO); multi-groups merging; K-means algorithm; fitness variance

传统 K-means 算法具有结构简单及收敛速度快等优点, 在数据挖掘、图像分割、模式识别等诸多领域得到了广泛应用. 然而在实际使用中, K-means 算法也存在对初始值敏感, 易陷入局部最优等问题.

为了克服这些缺点, 文献[1]提出利用灰度直方图的峰数估计最佳聚类数目, 并用 Qstu 法进一步确定聚类中心; 文献[2]分析了 K-means 算法与遗传算法在聚类实现上的差异, 并提出了两者结合能获取更好的聚类结果; 近期也有学者提出利用群体智能算法能有效改善 K-means 算法的聚类效果, 文献[3-4]文提出了基于人工鱼群的优化 K-means 聚类算法; 文献[5]提出了基于改进混合蛙跳的 K-means 聚类算法. 更多研究[6-9]则指出, 利用粒子群算法优化 K-means 聚类算法能取得更好的聚类性能. 上述改进算法的聚类结果都得到了一定的提高, 但由于未能有效解决 K-means 算法中利用欧式距离进行聚类的局限性, 因此对复杂数据集的聚类效果依然不够理想.

本文提出一种基于粒子群和多类合并的 K-means 混合聚类算法. 该算法将粒子群算法与 K-means 算法相结合, 并利用多类合并方法对聚类结果进行再次处理. 实验测试后证明本算法能达到较好的聚类效果.

1 K-means算法

K-means 算法能够使聚类域中所有数据到聚类中心距离的平方和最小. 其聚类原理为: 先取 K 个初始聚类中心, 计算每个数据点到这 K 个中心的距离, 找

① 基金项目:浙江省教育厅科研项目(Y201326770);宁波大学科研基金项目(XYL12009);浙江省教育厅科研项目(Y201326872);浙江省 2011 年度大学生新苗人才计划项目

收稿时间:2013-07-12;收到修改稿时间:2013-09-22

到最小距离把该数据点归入最近的聚类中心，并修改中心点的值为本类所有数据的均值，重复以上操作，直到中心位置不再变化时结束.

K-means 算法的实现步骤如下:

Step1: 从数据集中任意选取 K 个数据点作为初始聚类中心;

Step2: 分别计算每个数据点到 K 个中心的距离，将每个数据点指派到最相近的类;

Step3: 重新计算每个类的平均值，作为类的中心;

Step4: 如果划分结果不再发生变化或者达到最大迭代次数，则结束，否则转 Step2;

K-means 算法实现起来比较简单，其计算复杂度较低，且具有较好的可扩展性. 但是 K-means 算法也存在以下缺点: 聚类数目 K 必须预先给定以及对初始质心严重依赖，对数据输入顺序敏感; 易受噪声和异常数据的影响; 算法中基于欧式距离的划分规则存在较大的局限性.

2 粒子群优化算法(PSO)

粒子群算法是一种有效的全局寻优算法，能沟通过群体中粒子间的合作与竞争产生的群体智能指导优化搜索. 粒子群算法中，每一个优化问题的解都可以看作搜索空间中的一只鸟，即“粒子”. 首先生成初始化种群，即在可行解空间中随机初始化一群粒子，每一个粒子都为优化问题的一个可行解，并有目标函数为之确定一个适应度值. 每个粒子都将在解空间中运动，并由运动速度决定其飞行方向和距离. 通常粒子将随当前的最优粒子在解空间中搜索. 在每一次迭代中，每个粒子都能记住自己搜索到的最优解，记做 $pBest$，以及整个粒子群经历过的最好的位置，即目前整个群体搜索到的最优解，记做 $gBest$，粒子通过 $pBest$ 和 $gBest$ 不断调整自己的位置 x 来搜索新解. 具体根据下面两个公式来更新自己的速度与位置:

$$v_i(n+1)=\omega v_i(n)+c_1\cdot rand_1()\cdot(pBest-x_i(n))+c_2\cdot rand_2()\cdot(gBest-x_i(n)) \quad (1)$$

$$x_i(n+1)=x_i(n)+v_i(n+1) \quad (2)$$

其中，$v_i(n)$ 是当前粒子的速度，$x_i(n)$ 是粒子的当前位置. i=1, 2, …, N, N 是当前空间的维度. $rand_1()$，$rand_2()$ 是[0, 1]之间的随机数，c_1 和 c_2 为学习因子，通常取 $c_1=c_2$ =2, ω 是惯性权重，一般在 0.1 到 0.9 之间取值.

3 基于多类合并的PSO-Means聚类算法(M-PSO-Means)

本文算法将 K-means 聚类算法与改进的粒子群算法相结合，并且引入多类合并的思想进行最优聚类划分. 首先针对粒子群算法所存在的易陷入局部寻优和最优解振荡等缺陷，引入了适应度方差加快收敛速度加以改进，同时引入了聚类合并思想，能够有效解决 K-means 算法和粒子群算法基于欧氏距离划分聚类的缺陷. 主要改进措施如下:

3.1 加权因子

为了消除粒子群算法在寻优过程中出现的在最优解附近“振荡”现象，且 ω 较大粒子群具有较强的全局搜索能力，ω 较小则粒子群倾向于局部搜索. 因此，我们把加权因子做如下改进，速度更新公式的加权因子 ω 由最大加权因子 $\omega_{\max}$ 减少到最小加权因子 $\omega_{\min}$:

$$\omega=\omega_{\max}-run\frac{(\omega_{man}-\omega_{\min})}{runMax} \quad (3)$$

其中，run 为当前迭代次数，$runMax$ 为算法总迭代次数.

3.2 适应度方差

粒子群算法无论是早熟收敛还是全局收敛，粒子群中的粒子都会出现“聚集”现象，要么所有粒子聚集在某一特定位置，要么聚集在某几个特定位置，这主要取决于问题本身的特性以及适应度函数的选择. 粒子位置的一致等价于各粒子的适应度相同. 因此，研究粒子群中所有粒子适应度的整体变化就可以跟踪粒子群的状态，判断算法是否收敛.

适应度方差 σ^2 的公式如下:

$$\sigma^2=\sum_{i=1}^{N}\left(\frac{F_i-F_{avg}}{f}\right)^2 \quad (4)$$

其中，N 为粒子群的粒子数目，F_i 为第 i 个粒子的适应度，f 为粒子群目前的平均适应度，σ^2 为粒子群的群体适应度方差. F_{avg} 是归一化定标因子，其作用是限制的 σ^2 大小. f 可以取任意值，但需注意两个条件: ① 归一化后，整个粒子群 $\left|F_i-F_{avg}\right|$ 的最大值不大于 1; ② f 随算法的进化而变化. 在本文算法中，f 的取值采用如下公式:

$$f=\begin{cases}\max\left\{\left|F_i-F_{avg}\right|\right\}, \max\left\{\left|F_i-F_{avg}\right|\right\}>1\\ 1, others\end{cases} \quad (5)$$

群体适应度方差 σ^2 反映的是粒子群中所有粒子的“收敛”程度. σ^2 越小, 则粒子群趋于收敛; 反之, 粒子群则处于随机搜索阶段.

3.3 聚类合并

由于直接用改进的粒子群聚类算法聚类划分时, 适应度函数的选择十分苛刻, 很难选到合适的函数来得到最优的聚类划分. 并且粒子群算法和K-means算法的基于欧式距离的聚类划分标准存在严重的局限性, 对两个很紧密的聚类划分效果很不理想. 因此, 我们提出一种多类合并的思想, 能够有效的解决上述缺陷, 但是合并的条件必须选择合理才能达到最优的聚类划分.

本文算法选择基于密度的合并条件, 具体合并步骤和判断标准如下:

Step1: 计算预分聚类中的所有两两中心的距离存入一个矩阵中, 并对该矩阵按从小到大排序后生成一个新的排序矩阵range;

Step2: 取出当前聚类结果下的中心距离最小(range矩阵的第一组值)的两个中心;

Step3: 取这两个中心的中值作为预合并中心;

Step4: 以最小中心距离的 1/2 为密度范围, 将Step2 和 Step3 选的三个中心按密度范围计算其密度, 原始中心的密度分别记为Da和Db, 预合并中心的密度记为Dc;

Step5: 判断预合并中心密度(Dc)是否大于等于1/4的原始中心密度之和,公式如下:

$$\mathrm{Dc} \geq \frac{1}{4}(\mathrm{Da}+\mathrm{Db}) \tag{6}$$

若不满足则选择中心距离排序矩阵(range)中下一组值对应的两个初始中心, 并且转 Step3; 满足条件则把两原始聚类合并, 重新划分合并后的聚类;

Step6: 若合并后的聚类数目达到最优数目 K, 则跳出合并步骤, 保存最优聚类结果, 否则转 Step1 继续合并.

3.4 算法实现

3. 4.1 参数编码

在本文算法中, 粒子群算法部分采用的是基于聚类中心的编码方式, 也就是每个粒子的位置是由 m 个聚类中心组成, 粒子除了位置之外,还有速度 V 和适应度值 F. 由于样本向量维数为 d, 因此粒子的位置是 $m\times d$ 维变量, 所以粒子的速度也应当是 $m\times d$ 维变量, 另外每个粒子还有一个适应度值.

因此, 粒子采用下面的编码结构:

$C_{11}C_{12}\text{---}C_{1d}C_{21}C_{22}\text{---}C_{2d}\text{---}C_{m1}C_{m2}\text{---}C_{md}$	$V_1V_2\text{---}V_{m*d}$	F_i

其中, $C_{i1}C_{i2}\text{---}C_{id}$ 表示第 i 个 d 维的聚类中心, $1\leqslant i\leqslant m$, V_{m*d} 表示第 $m*d$ 维速度.

则, 整个粒子群采用下面的编码格式:

$$POP=\begin{Bmatrix} X_{11},X_{12},X_{13},\text{---},X_{1m},\ V_{11},V_{12},V_{13},\text{---},V_{1m},\ F_1 \\ X_{21},X_{22},X_{23},\text{---},X_{2m},\ V_{21},V_{22},V_{23},\text{---},V_{2m},\ F_2 \\ \cdots\cdots \\ X_{N1},X_{N2},X_{N3},\text{---},X_{Nm},\ V_{N1},V_{N2},V_{N3},\text{---},V_{Nm},\ F_N \end{Bmatrix}$$

其中, $X_{ij}=C_{j1}C_{j2}\text{---}C_{jd}$, i 代表了属于第几个粒子.

当聚类中心确定时, 聚类的划分由下面的最近邻法则决定, 若 x_i, c_j 满足:

$$\|x_i-c_j\|=\min\|x_i-c_k\|, k=1,2,\cdots,\ m \tag{7}$$

则 x_i 属于第 j 类. 对于某粒子, 按照以下方法计算其适应度:

$$F_i=\sum_{i=1}^{L}\sum_{j=1}^{m}\|x_i-c_{ij}\|^2 \tag{8}$$

其中 L 为样本数, x_i 为输入样本, F_i 代表第 i 个粒子适应度值.

3.4.2 算法描述及具体流程

M-PSO-Means算法首先把预设的最优聚类数目K增大数倍后赋值给变量 G, 然后把 G 作为类别数目带入优化的粒子群算法进行中心点的寻优, 然后调用K-means 算法优化聚类, 完成较好的预聚类结果, 最后根据特定的合并原则, 把预分的类别进行合并, 直到合并后的类别达到最优的聚类数目K后停止并输出最优的聚类.

算法具体流程如下:

Step1: 给定最佳聚类数目 K, 令 $G=3*K$, 将G作为预聚类数目代入优化的粒子群算法进行中心寻优;

Step2: 种群的初始化: 在初始化粒子时, 先将每个数据点随机指派为某一类, 作为最初的聚类划分, 并计算各类的聚类中心, 作为初始粒子的位置编码, 计算粒子的适应度, 同时作为粒子的个体最优位置, 并随机初始化粒子的速度. 反复进行 N 次, 共生成 N 个初始粒子群;

Step3: 对每个粒子, 比较它的适应度和它经历过的最好位置的适应度, 如果更好, 则更新该粒子的最好位置;

Step4: 对每个粒子, 比较它的适应度和群体所经

历的最好位置的适应度，如果更好，则更新全局最好位置;

Step5: 根据式(1)、(2)调整粒子的速度和位置;

Step6: 对于新一代粒子，按照以下的方法进行优化:

根据粒子的聚类中心编码，按照最近邻法则,来确定对应该粒子的聚类划分;

按照聚类划分，计算新的聚类中心，更新粒子的适应度值,取代原来的编码值. 按照这样的方式划分聚类可以使得算法的收敛速度大大提高;

Step7: 计算粒子群算法的适应度方差值;

Step8: 判断适应度方差是否小于设定的阈值，或算法达到最大迭代次数，若是则结束粒子群寻优，并输出最优中心，否则转 Step3;

Step9: 将粒子群聚类算法得到的中心作为 K-means 的初始中心，利用 K-means 算法进行进一步聚类划分，并输出预聚类划分结果;

Step10: 按 3. 3 所述执行聚类合并步骤，对预划分的聚类进行合并;

Step11: 输出最优聚类结果.

算法的时间复杂度为 $O(m^2)$. 其中 m 为粒子群的种群数.

上述步骤中，在 Step6 对新一代个体进行重新划聚类别时,有可能会有空聚类出现. 如果出现空聚类,则将该类的中心进行如下处理:

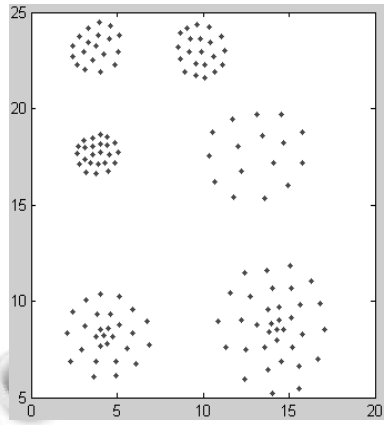
$$C_k = \text{minsam} + (\text{maxsam} - \text{minsa m}) * \text{rand}(1,d) \quad (9)$$

其中，C_k 为该空聚类的中心，minsam 为数据样本坐标点中最小的值，maxsam 为数据样本坐标点中最大的值，rand(1,d) 为随机生成与中心坐标维数相同的一个点.

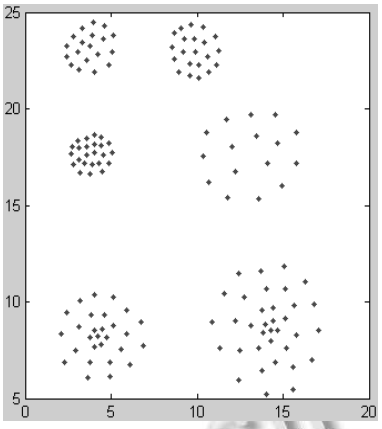
4 实验测试及结果分析

实验选取了三组标准数据集进行测试，可分别划分为 6 个、3 个和 7 个聚类数目. Aggregation1 中各数据群划分较为明显; Aggregation2 中小数据群与较大数据群连接紧密，常规方法容易出现错误划分；而 Aggregation3 为复杂不规则类型，其中的各数据群大小、形状、距离都比较多变，利用现有聚类方法难于进行聚类处理. 三个数据集的结构分别如图 1 所示.

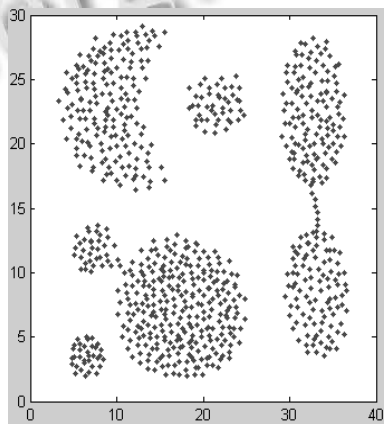
本文算法参数具体设置如下，粒子群的粒子个数 m 设为 50 个，最佳聚类数目 K 根据数据集取值，c_1=c_2 =2，ω 按式(3)进行线性变化，$\omega_{\min}$=0.9，$\omega_{\min}$=0.4，适应度方差阈值 threo 设置为 0.07，算法中粒子群混合聚类部分的循环次数 M 设置为 100 次.



(a) Aggregation1



(b) Aggregation2

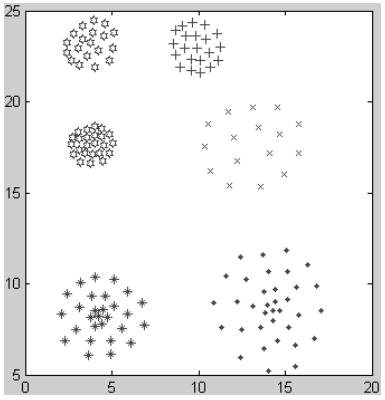


(c) Aggregation3

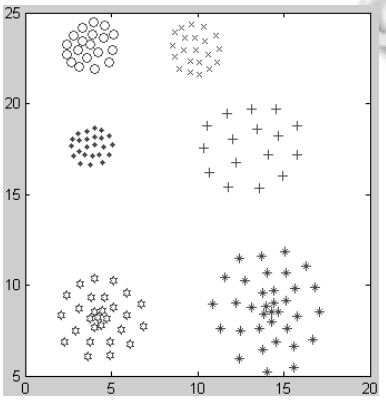
图 1 三组数据集的分布图

为对比本算法的聚类性能，我们在 MATLAB 环境下，分别用 K-means 算法，PSO-Means 算法和本文

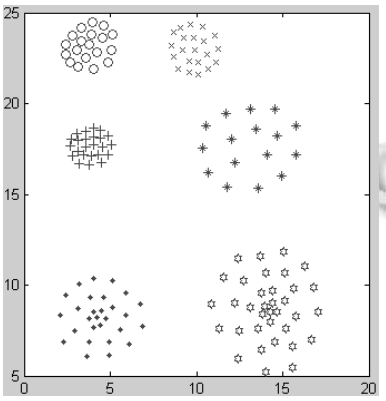
M-PSO-Means 算法对这三组数据集各进行 50 次的程序运行实验. 实验效果总结如下图 2, 3, 4 所示.



(a) K-means 聚类结果



(b) PSO-Means 聚类结果



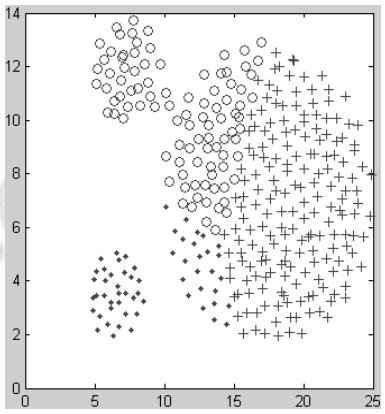
(c) M-PSO-Means 聚类结果

图 2　三种算法对于 Aggregation1 的聚类效果对比

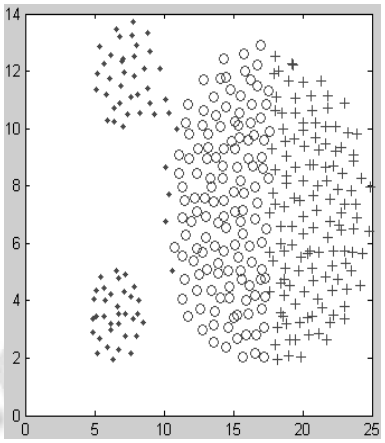
从表 1 中可以看出: 针对数据集 Aggregation1 的测试, M-PSO-Means 算法所需时间最短; 而在其他两个数据集的测试中, K-means 算法具有更快的聚类速度. PSO-Means 算法总体需要更多的运算时间.

表 1　各算法运算时间

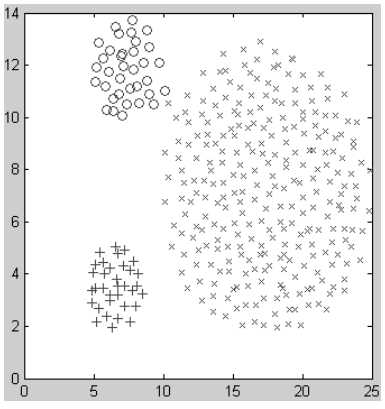
数据集	K-means	PSO-Means	M-PSO-Means
Aggregation1	1.6106	1.7619	0.6781
Aggregation2	0.2928	1.1191	1.3116
Aggregation3	0.9921	2.9686	2.7016



(a) K-means 聚类结果

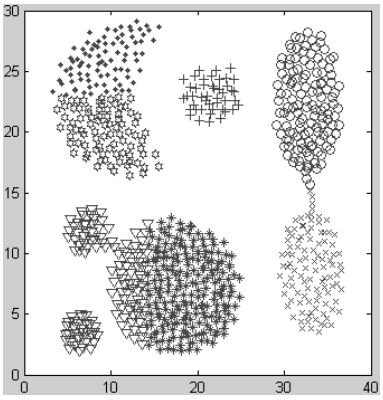


(b) PSO-Means 聚类结果

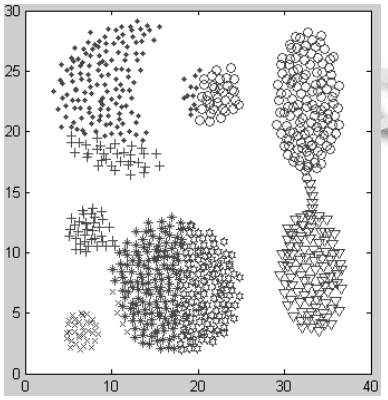


(c) M-PSO-Means 聚类结果

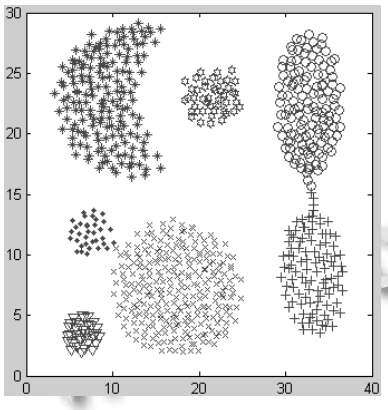
图 3 三种算法对于 Aggregation2 的聚类效果对比



(a) K-Means 聚类结果



(b) PSO-Means 聚类结果



(c) M-PSO-Means 聚类结果

图 4 三种算法对于 Aggregation3 的聚类效果对比

由图 2 可以看出, 对于聚类的类内数据点分布均匀并且类与类的间距较大的数据集 Aggregation1, 不管是 K-means 算法、PSO-Means 算法或者 M-PSO-Means 算法, 聚类结果都比较理想. K-means 算法在收敛速度上相对较快, 但是存在不稳定的情况, 偶尔会出现较差的聚类效果, 其聚类结果容易受初值影响. 而 M-PSO-Means 算法和 PSO-Means 算法则更稳定, 基本都能做到准确聚类.

由图 3、图 4 可以看出, K-means 算法和 PSO-Means 算法对 Aggregation2、Aggregation3 的聚类效果不理想, 而 M-PSO-Means 算法对复杂数据集的聚类效果则更加优越. K-means 算法和 PSO-Means 算法在对大类附近有小类的数据集进行聚类时, 容易把小类划分到大类中, 如图 3 的 a 所示, 无法有效地将小类划分出来, 而 M-PSO-Means 算法则能很好的将小类划分出来. 这是由于 K-means 算法和 PSO-Means 算法采用基于欧式距离的划分标准, 对于 Aggregation2 和 Aggregation3 这种类间距较小的数据集, 聚类结果存在较大缺陷. 而本文提出的 M-PSO-Means 算法则能很好的解决这个缺陷.

可以看出, 相对于 K-means 算法和 PSO-Means 算法, M-PSO-Means 算法对较复杂数据集的聚类效果更优越并且更加稳定, 能做到局部寻优与全局寻优有效的结合, 具有较好的收敛效率.

5 结论

本文提出了一种基于粒子群和多类合并的 M-PSO-Means 聚类算法, 该算法利用粒子群算法寻找初始聚类中心, 并且根据其适应度方差选择 K-means 算法的操作时机, 实现了粒子群算法与 K-means 算法的有效结合, 加快了算法的聚类速度, 最后进行多类合并操作, 实现聚类的最优划分. 理论分析和数据实验结果表明, 本文算法克服了传统 K-means 聚类算法存在的问题, 全局寻优能力优于现有的 K-means 算法和 PSO-Means 算法, 具有较好的聚类效率, 对复杂数据集的聚类划分效果更加优越.

参考文献

1 Yao H, Duan Q, Li D, Wang J. An improved K-means clustering algorithm for fish image segmentation, Mathematical and Computer Modelling, 2013, 58(3): 784–792.

2 Zahraie B, Roozbahani A. SST clustering for winter precipitation prediction in southeast of Iran: Comparison between modified K-means and genetic algorithm-based clustering methods. Expert Systems with Applications, 2011, 38(5): 5919–5929.

(下转第 69 页)

图 6 嵌入水印的文件标记

(2) 实用性: 100%文件能够在不同计算机操作系统, 不能格式中实现水印算法.

(3) 安全性: 100%嵌入水印的文本文件在未安装水印系统中不能打开.

在系统实际应用中, 研究总结近半年来在某单位试运行和反馈情况, 结合软件测评结果, 系统从以下几个标准进行分析:

(1) 安全性: 通过下发密钥配置表完成密钥管理, 通过用密钥对数字水印信息加密和对文本编码循环加密实现文本信息安全保证.

(2) 鲁棒性: 由于加密后水印信息采用循环异或加方法嵌入到文本编码中, 只要有一部分文本编码未被篡改既能提取水印.

(3) 水印容量: 本算法的水印容量主要取决于文本载体编码量, 理论上至多具有与文本等量的数字水印容量.

(4) 适用性: 水印算法主要针对文本 Unicode 编码进行水印嵌入和提取, 能够针对多种格式文本文档, 具有很强的适用性.

5 结论

公文系统安全管理目前主要采用对文件的密钥加密, 在密级控制和文本安全性都有缺陷, 本文提出了基于 Unicode 编码的数字水印分密级公文系统, 即实现了对公文文本编码级的加密同时通过编码数字水印实现分密级管理, 且效果良好, 适用广泛.

设计的系统通过文本 Unicode 编码嵌入使用密钥加密后的水印信息并附加密级编码、奇偶校验码和纠错码等进行编码实现的文本数字水印, 通过数字水印和密钥配置实现涉密文件的分密级管理, 并在文件流转全过程中安全管理.

参考文献

1 蒲天银.安全隔离网闸技术发展探讨.计算机时代,2006,(6):18.

2 蔡谊,沈昌祥.PKI 技术在电子政务中的应用.计算机应用研究,2002,19(10):11–13.

3 雷震声.计算机网络管理及系统开发.北京:电子工业出版社,2002:156–160.

4 Awrangjeb M. An overview of reversible data hiding. ICCIT 2003. Bangladesh. 2003. 75–79.

5 白剑,杨榆,徐迎晖等.基于文本的信息隐藏算法.计算机系统应用,2005,14(4):32–35.

6 陆绿,方勇.基于字符 Unicode 奇偶性的数字水印设计与实现.计算机技术与发展,2010,20(8):176–179.

7 黄国超,王衍波,张凯泽.基于 Unicode 编码的信息隐藏算法研究与设计.计算机技术与发展,2011,21(10):233–236.

8 梁红玉,陈冬梅.扩展汉明码的交织重排算法研究及其实现.计算机应用,2012,32(S1):85–87.

9 冀斌.Windows 平台下应用软件多语言支持.计算机工程,2004,30(12):163–165.

10 房婧婧.基于不可见水印的纸质文档泄密源头管理系统的设计与实现[学位论文].北京:北京邮电大学,2010.

(上接第 165 页)

3 楚晓丽.K-Means 聚类算法和人工鱼群算法应用于图像分割技术.计算机系统应用,2013,22(4):92–94.

4 于海涛,贾美娟,王慧强等.基于人工鱼群的优化 K-means 聚类算法.计算机科学,2012,39(12):60–64.

5 王联国,韩晓慧,宋磊.基于改进混合蛙跳-K 均值聚类算法的无功电压控制分区.传感器与微系统,2013,32(6):18–21.

6 刘衍民,隋常玲,赵庆祯.基于 K-均值聚类的动态多种群粒子群算法及其应用.控制与决策,2011,26(7):1019–1025.

7 鲍新中.基于粒子群的 K 均值算法和粗糙集理论的财务预警.系统管理学报,2012,21(4):461–469.

8 Kalyani S, Swarup KS. Particle swarm optimization based K-means clustering approach for security assessment in power systems. Expert Systems with Applications, 2011, 38(9): 10839–10846.

9 Tsai CY, Kao IW. Particle swarm optimization with selective particle regeneration for data clustering. 2010 Elsevier Ltd. 2011, 38(6): 1–6.