

基于改进粒计算的 K -medoids 聚类算法

潘楚*, 罗可

(长沙理工大学 计算机与通信工程学院, 长沙 410114)

(* 通信作者电子邮箱 panchu2012@163.com)

摘要:针对传统 K -medoids 聚类算法对初始聚类中心敏感、收敛速度缓慢以及聚类精度不够高等缺点,提出一种基于改进粒计算、粒度迭代搜索策略和优化适应度函数的新算法。该算法利用粒计算思想在有效粒子中选择 K 个密度大且距离较远的粒子,选择其中心点作为 K 个聚类初始中心点;并在对应的 K 个有效粒子中进行中心点更新,来减少迭代次数;采用类间距离和类内距离优化适应度函数来提高聚类的精度。实验结果表明:该算法在 UCI 多个标准数据集集中测试,在有效缩短迭代次数的同时提高了算法聚类准确率。

关键词: K -medoids 聚类算法;改进粒计算;粒度迭代搜索策略;优化适应度函数

中图分类号: TP301.6 **文献标志码:** A

Improved K -medoids clustering algorithm based on improved granular computing

PAN Chu*, LUO Ke

(School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha Hunan 410114, China)

Abstract: Due to the disadvantages such as sensitive to the initial selection of the center, slow convergent speed and poor accuracy in traditional K -medoids clustering algorithm, a novel K -medoids algorithm based on improved Granular Computing (GrC), granule iterative search strategy and a new fitness function was proposed in this paper. The algorithm selected K granules using the granular computing thinking in the high-density area which were far apart, selected its center point as the K initial cluster centers, and updated K center points in candidate granules to reduce the number of iterations. What's more, a new fitness function was presented based on between-class distance and within-class distance to improve clustering accuracy. Tested on a number of standard data sets in UCI, the experimental results show that this new algorithm reduces the number of iterations effectively and improves the accuracy of clustering.

Key words: K -medoids clustering algorithm; improved granular computing; granule iterative search strategy; improved fitness function

0 引言

聚类算法是根据数据的相似性将数据对象划分在不同簇中的过程,使得同一簇中的对象具有较高的相似度,不同簇中数据对象不相似。 K -medoids 算法是属于基于划分方法的一种^[1],是 K -means 算法的改进。 K -medoids 算法能有效处理异常数据和噪声数据,具有很好的鲁棒性,所以在聚类算法中得到广泛的应用。但是在现实中对不同数据集的处理还是存在不少问题,如算法对初始化敏感、时间复杂度较高、准确率不高等。因此,最近许多学者提出了很多改进的算法:文献[2]通过一次性计算对象间的距离生成矩阵,以此来初始化中心点和中心点更新,有效地减少计算时间,但性能与传统 K -medoids 相当。文献[3]提出了一种基于初始化中心微调与增量中心候选集的改进算法,时间复杂度稍微降低,但准确率不高。文献[4-5]提出了利用粒计算来初始化 K -medoids 聚类中心,起到一定的优化作用,但其初始聚类中心点有位于同一簇中现象,导致准确率不高。文献[6-7]分别利用密度信息来产生初始中心点、增量式的思想扩大搜索范围,提高了准

准确率但增加了时间复杂度。文献[8-9]分别提出了基于最小支撑树初始化聚类中心点,基于 metric access method 的聚类方法,提高了聚类的准确率但也增加了额外的开销。文献[10-12]结合人工智能仿生算法优化 K -medoids 聚类算法,提高算法全局搜索能力,但同时也牺牲时间复杂度。

上述文献中,虽然对 K -medoids 聚类算法进行了改进。有些算法解决了初始化敏感问题,但增加了时间复杂度;有些算法降低了时间复杂度,但聚类准确率没有明显提高。它们都没能同时很好解决传统 K -medoids 聚类算法对初始化敏感、复杂度较高和聚类准确率不高的问题。鉴于此,本文提出了一种新的改进算法:1)采用改进粒计算来初始化聚类中心,以解决传统算法对初始化敏感的问题;2)提出一种粒度迭代搜索策略来降低算法的时间复杂度;3)采用类间距离和类内距离优化适应度函数来提高聚类准确率。

1 预备知识

1.1 K -medoids 算法简介

K -medoids 聚类算法选取数据集中实际对象来代表簇,每

收稿日期: 2013-12-30; **修回日期:** 2014-03-25。 **基金项目:** 国家自然科学基金资助项目(11171095, 71371065); 湖南省自然科学基金衡阳联合基金资助项目(10JJ8008); 湖南省科技计划项目(2013SK3146); 湖南省研究生科研创新项目(CX2014B386)。

作者简介: 潘楚(1986-),男,湖南怀化人,硕士研究生,主要研究方向:数据挖掘; 罗可(1961-),男,湖南长沙人,教授,博士,主要研究方向:数据挖掘。

个簇使用一个代表对象,其余的对象被分配到与其最为相似的代表对象所在的簇中。划分方法是基于最小化所有对象 p 与其对应的代表对象之间的相异程度之和的原则来进行划分。使用一个绝对误差标准函数^[1]来衡量聚类效果,定义为式(1):

$$E(w) = \sum_{i=1}^K \sum_{p \in c_i} |p - o_i| \quad (1)$$

其中: p 为簇 c_i 的对象, o_i 为聚类中心, p 和 o_i 是多维的。

传统的 K -medoids 聚类算法描述如下。

输入 簇的数目 K , 包括 n 个对象的数据集 SET 。

输出 K 个最优簇集合。

步骤1 从 SET 中随机选择 K 个对象 o_i 作为初始中心点, 记为 w 。

步骤2 把剩余对象指派给离对象最近的中心点所代表的簇, 并计算 $E(w)$ 。

步骤3 随机选择一个非代表对象 o_{random} 代替代表对象 o_i , 更新每个簇中心点并重新聚类划分, 记为 w' , 计算 $E(w')$ 。

步骤4 计算 $\Delta E = E(w') - E(w)$, 若 $\Delta E < 0$, o_{random} 替换 o_i , 此时记 w' 为 w ; 否则不变。

步骤5 重复执行步骤3至步骤4, 直到簇集合中对象不再发生改变, 结束算法。

1.2 粒计算简介

粒计算 (Granular Computing, GrC) 是美国学者 Lin 于 1997 年提出的^[13]。粒就是指一些个体通过不分明关系、相似关系、邻近关系或功能关系等所形成的块^[14]。聚类分析是以“最优”相似度为基础, 在所有可能的粒度中, 求出一个“最优”粒度。从信息粒度的角度看, 聚类就是在一个统一的粒度下分析和处理问题, 把粒度计算的思想运用到聚类算法中, 会得到较理想的结果^[15]。结合粒和数据挖掘的相关概念, 基于粒计算的初始化具体表述如下。

设 $T = (U, B)$ 为聚类空间, U 为论域, B 为属性集合。相似度 $S(x_i, x_j)$ 定义为:

$$S(x_i, x_j) = \frac{1}{1 + \sum_{l=1}^{|B|} w_l |x_{il} - x_{jl}|} \quad (2)$$

其中: $|B|$ 为属性集合个数; x_{il}, x_{jl} 分别为 x_i, x_j 的第 l 属性值; w_l 为属性分辨能力, 定义为:

$$w_l = \frac{\sum_{i=1}^m |x_i|^2}{|U|^2} \quad (3)$$

属性 l 值被划分成区间块 $\{x_1, x_2, \dots, x_m\}$, $|x_i|$ 为 x_i 划分区间的对象个数, $|U|$ 为对象集的对象总数, m 为划分块数。若对象数为 n , 则对象平均相似度定义为:

$$\bar{d} = \sum_{i,j=1}^n [S(x_i, x_j) / n^2] \quad (4)$$

设 U 上 n 个粒子划分为 $\{X_1, X_2, \dots, X_n\}$, 则其中粒子 X_i 的密度表示为:

$$gd(X_i) = |X_i| / |U| \quad (5)$$

其中 $|X_i|$ 表示第 i 个区间里的对象个数。因为 U 中粒子个数为 n , 则粒子的平均密度定义为:

$$\overline{GD} = \sum_{i=1}^n gd(X_i) / n \quad (6)$$

设第 i 个粒子包含 N 个对象, 分别为 $x_{i1}, x_{i2}, \dots, x_{iN}$, 则该粒子的中心点 c_i 定义为:

$$C_i = \left\{ x_{ij} \mid \min_{j=1}^N \left| x_{ij} - \frac{1}{N} \sum_{k=1}^N x_{ik} \right| \right\} \quad (7)$$

粒计算初始化具体步骤为:

1) 根据式(2)和式(4), 分别计算对象间的相似度以及对象平均相似度。

2) 设定一个阈值 d , 其中 d 的范围为 \bar{d} 上下浮动 10% 左右, 具体大小根据实验效果设定。如果 $S(x_i, x_j) \geq d$, 则 $M(i, j) = 1$; 否则为 $M(i, j) = 0$, 其中 M 表示为对象间的模糊相似矩阵。

3) 将对象按矩阵 M 进行归类得到对象的粗粒集 G , 即记录每个对象的相似对象的编号。每个对象与其相似的所有对象的集合称为一个粒子。

4) 根据式(5)计算每个粒子的密度, 按式(6)计算粗粒集 G 的平均密度, 选择 $gd(x_i) \geq \overline{GD}$ 的粒子作为有效粒子放入集合 I 中, 称 I 为有效粒子集。

5) 按式(7)计算有效粒子集 I 中每个粒子的中心, 并计算任意两个粒子间的欧氏距离记录在 D 中。

6) 在 I 中取最大密度粒子的中心作为第一个聚类中心 o_1 , 该粒子记为 Q_1 ; 选择距最大密度粒子最远且密度最大的粒子的中心点作为第二个聚类中心 o_2 , 该粒子记为 Q_2 ; 对于 I 中剩余粒子, 根据 D , 分别求出其中心点到 o_1, o_2, \dots, o_n 的距离为 $d_{i1}, d_{i2}, \dots, d_{in}$, 取 $d_i = \min(d_{i1}, d_{i2}, \dots, d_{in})$, 然后计算 $\max(d_i)$, 对应的粒子中心为 o_i , 该粒子记为 Q_i , 以此类推计算 o_k , 对应粒子记为 Q_k 。

2 算法的改进

2.1 粒计算的改进

1.2 节中算法的第3)步得到的粗粒集 G , 在归类过程出现对象重复归类以及某些粒子雷同现象, 这样就会导致有效粒子中对象冗余和相同粒子出现在 I 中, 从而在第6)步中找出的中心点很有可能位在同一簇中。针对原始算法的缺陷, 首先本文在归类过程中采用一个对象(已经编号)只跟其后的对象逐个比较, 而不与该对象之前的比较的方法, 从而避免了对象重复归类, 这样得到更好的粗粒集; 其次在此粗粒集的基础上进行去重处理, 即去除粒子中对象个数、属性值都相同的多余粒子, 得到对象个数按大小排序且不重复的细粒集 \bar{G} , 从而得到不重复的有效粒子集 I , 这样根据第6)步得出初始化 K 个中心点将位于各异的 K 簇内。

2.2 粒度迭代搜索策略的提出

传统 K -medoids 算法的时间复杂度为 $O(k(n-k)^2)$, 其主要时间开销来源于中心点的更新和重新聚类划分, 传统 K -medoids 算法计算代价较大。为了提高算法的性能, 本文提出了一种粒度迭代搜索策略。该策略是分别在初始中心点 o_1, o_2, \dots, o_k 所对应的粒子 Q_1, Q_2, \dots, Q_k 内进行中心点的更新, 这样避免了中心点在全局对象中随机选择。原因在于 Q_1, Q_2, \dots, Q_k 这些粒子都是有效粒子且高密度, 与初始中心点相似的对象很大程度上属于同一簇, 这样迭代的好处是在有效的范围内寻找最优或者近似最优。该策略将在一定程度上降低算法的时间复杂度。

2.3 适应度函数的改进

传统K-medoids聚类算法的准则函数如式(1)所示,其思想是将类内距离和最小作为衡量聚类质量的标准。而聚类的核心思想是要求类内距离最小,类间距离最大,而式(1)只考虑了类内距离,忽略了聚类总体质量是类内距离和类间距离共同决定的。基于这一思想,本文提出了一种新的准则函数作为K-medoids算法的适应度函数,能有效平衡类内距离和类间距离在聚类中的作用。

设类间距离 $O(w)$ 为不同簇中心点的距离,定义为:

$$O(w) = \sum_{i,j=1}^K |o_i - o_j| \quad (8)$$

改进的适应度函数可以表示为:

$$F(w) = \frac{\sqrt{O(w)}}{\sqrt{E(w)}} \quad (9)$$

改进后的适应度函数将在实验过程中不断地调整函数值,当改进的适应度函数值达到最大值时,新算法将得到最优或近似最优聚类结果。

2.4 改进算法的具体描述

输入 包含 n 个对象的数据集 SET ,簇的数目 K 。

输出 K 个簇类中心及聚类准确率。

步骤1 利用改进的粒计算初始化 K 个聚类中心 o_1, o_2, \dots, o_k ,同时记录中心点所对应的粒子为 Q_1, Q_2, \dots, Q_k 。

步骤2 把剩余对象指派给离对象最近的中心点所代表的簇,记为 w ,并计算 $F(w)$ 。

步骤3 分别在对应的粒子 Q_1, Q_2, \dots, Q_k 中选择一个非代表对象 o_{random} (各不相同)代替代表对象 o_i ,更新每个簇中心点并重新聚类划分,记为 w' ,计算 $F(w')$ 。

步骤4 计算 $\Delta F = F(w') - F(w)$ 。若 $\Delta F > 0$, o_{random} 替换 o_i ,此时记 w' 为 w ;否则不变。

步骤5 重复执行步骤3至步骤4,直到簇集合中对象不再发生改变,结束算法。

2.5 改进算法的分析

改进算法首先采用改进的粒计算有效初始化 K 个聚类中心,使聚类算法在聚类初始就达到准平衡状态;然后充分运用粒子高密度且高相似度的特性,把中心点的更新缩小在粒子的范围内,以提高算法的效率;并且提出新的适应度函数来作为衡量聚类质量的准则,以提高算法准确率。

在算法时间复杂度上,传统K-medoids算法的开销主要来源于中心点的更新和重新聚类划分。对于重新聚类划分,每次必须迭代 $n-k$ 次;而对于中心点的更新,可以在有效区域进行迭代选择,没必要全局搜索。本文提出的粒度迭代搜索策略就是把中心点的更新范围缩小在粒子内。根据改进粒计算得到的有效粒子集,由2.2节分析,考虑允许误差范围内,可以确定有效粒子中的对象个数为 n/k ,那么中心点的更新范围在 $[2, n/k]$ 内,这样中心点的平均更新范围为 $(n-2k)/2k$ 。根据算法流程,中心点更新次数为 $(n-2k)/2k$,重新聚类划分为 $n-k$ 次,中心点的更新和聚类划分共进行 k 次,改进算法的总体时间复杂度为 $O((n-2k)(n-k)/2)$ 。与传统K-medoids聚类算法时间复杂度 $O(k(n-k)^2)$ 相比,虽然没能在数量级上的降低,但是其系数降低到原来的 $1/(2k)$,也是一个不小的进步,将通过实验验证。

3 实验结果与分析

实验环境:硬件:AMD Dual-Core CPU T4400 ++ 2.31 GHz, 2 GB 的内存。软件:操作系统 Windows XP,集成开发软件VC++6.0。实验采用的数据集:本文共选用了UCI中的Iris数据集,Wine数据集,Pima-indians-diabetes数据集(简称为Pid)。

为了验证本文算法的可行性和高效性,采用Partitioning Around Medoids(PAM)算法^[1]、快速K-medoids算法^[2]、文献[5]中算法及本文算法,分别在以上数据集上进行聚类测试,每种算法运行10次计算其平均值。

本文所用Iris、Wine、Pid标准数据集特征如表1所示。

表1 实验中涉及的数据集

数据集	对象数目	属性维度	类别个数	每类个数
Iris	150	4	3	50, 50, 50
Wine	178	13	3	59, 71, 48
Pid	768	8	2	500, 268

在PAM算法和快速K-medoids算法中,只需要设置一个参数——标准数据集的簇个数 K 。然而 K 的设置是根据不同数据集给出不同数值。其中:Iris数据集参数 $K=3$;Wine数据集参数 $K=3$;Pid数据集参数 $K=2$ 。针对文献[5]算法和本文算法,根据各数据集 \bar{d} 的不同,通过实验验证得出参数设置如下:其中Iris数据集参数: $d=0.73$, $K=3$;Wine数据集参数: $d=0.021$, $K=3$;Pid数据集参数: $d=0.20$, $K=2$ 。

首先,选用Iris数据集对改进粒计算有效初始化进行验证,同时证明粒度迭代搜索策略的可行性和科学性。实验结果如表2所示。

表2 Iris数据集初始化结果

粒子编号	对象个数	中心点	粒子包含对象编号
$Q_1 = 2$	69	2	2, 6, 9, 15, 17, 18, 19, 21, 22, 23, 24, 30, 31, 32, 34, 35, 36, 37, 38, 39, 47, 49, 50, 51, 52, 53, 60, 62, 65, 66, 67, 68, 75, 77, 78, 79, 80, 81, 82, 84, 90, 92, 93, 95, 96, 97, 98, 105, 106, 107, 108, 109, 110, 111, 112, 114, 120, 121, 122, 123, 124, 128, 129, 135, 139, 141, 142, 143, 144
$Q_2 = 21$	44	112	21, 23, 24, 35, 36, 37, 38, 39, 49, 50, 51, 52, 53, 54, 65, 66, 67, 68, 69, 75, 78, 80, 81, 82, 84, 95, 96, 97, 98, 108, 110, 111, 112, 113, 114, 122, 124, 125, 128, 129, 140, 141, 142, 143
$Q_3 = 28$	37	43	28, 29, 40, 41, 42, 43, 44, 55, 56, 58, 70, 71, 72, 74, 85, 86, 87, 88, 89, 100, 101, 102, 103, 115, 116, 117, 118, 119, 131, 132, 133, 134, 145, 146, 147, 148, 149

表2中 Q_1, Q_2, Q_3 的粒子编号分别为:2、21、28,它们分别位于Iris标准数据集的第0簇、1簇、2簇。而且与 Q_1, Q_2, Q_3 对应的中心点 o_1, o_2, o_3 分别为:2、112、43,它们同样也对应地位于第0簇、1簇、2簇,这样就有效地避免了文献[5]利用原始粒计算进行初始化,聚类中心点有位于同一簇中的现象。其中 Q_1, Q_2, Q_3 中所包含的对象属于第0簇、1簇和2簇的对

象个数均为30,这样中心点在 Q_1, Q_2, Q_3 中更新,能在更小的搜索范围内找到最优解或者近似最优解,证明新的搜索策略可行。从 Q_1, Q_2, Q_3 粒子中对象个数可以算出改进算法的平均迭代次数为10805,而传统K-medoids迭代次数为64827,改进算法的迭代次数是传统算法的1/5.99,以此验证了2.5节的时间复杂度的理论分析结果。根据实验结果,改进粒计算对Iris数据集初始化后聚类准确率为93.33%,远远超过了PAM算法和快速K-medoids算法和文献[5]算法的最优准确率。实验结果表明改进粒计算达到了有效初始化的目的,同时也验证粒度迭代搜索策略的可行性和科学性。

其次,比较各算法在不同数据集上的准确率,以验证新算法中改进适应度函数的有效性和必要性,并验证新算法的总体高效性,实验数据如表3所示。

表3 各类聚类算法在数据集上的准确率比较 %

聚类算法	Iris	Wine	Pid
PAM 算法	77.56	52.87	49.96
快速 K-medoids	89.32	70.79	62.24
文献[5]算法	90.00	70.79	65.63
本文算法	98.00	85.39	73.46

表3的实验结果表明本文算法在三种不同的数据集中测试,其聚类准确率比其他算法都有很大的提升。其中在Iris数据集中,本文算法收敛最优中心点为:121、36、56,故其聚类准确率高达98.00%。而在Wine和Pid数据集上的聚类准确率虽然不是很高,但相比其他算法都有大幅度的提高,分别收敛近似最优中心点:17、93、170和239、394。实验结果表明:新算法能收敛最优或近似最优中心点;同时相比其他算法,新算法能大幅度提高Wine、Pid这些不规则数据集的准确率,以此验证改进的适应度函数的有效性和必要性。

综上所述,本文列举多种算法在相应的数据集中测试,本文算法利用有效的初始化手段,改进中心点搜索策略和提出新的适应度函数,在降低时间复杂度的情况下又能大幅度提高聚类质量。上述实验结果验证了本文改进算法的先进性。

4 结语

传统K-medoids聚类算法对初始化敏感,时间复杂度较高,聚类准确率不够高。本文提出一种新的改进算法,能有效初始化初始聚类中心点,使其分别位于不同的簇中,有效避免了传统算法对初始化敏感的问题;提出粒度迭代搜索策略和加权优化适应度函数在一定程度上降低了时间复杂度的同时又大幅度提升了聚类准确率。本文得到了理想的实验结果,验证了新算法的可行性和有效性。但是,仔细分析表3,由于欧氏距离作为衡量尺度的缺陷性,随着数据集中数据的维度增大或对象数目的增加,本文算法的聚类准确率也在相应地降低,由此可知算法的健壮性还有待进一步改进。这将是我们将下一步深入研究的方向。

参考文献:

[1] HAN J W, KAMBER M, PEI J. Data mining: concepts and techniques[M]. 3rd ed. FAN M translated. Beijing: China Machine Press, 2012: 293 - 297. (HAN J W, KAMBER M, PEI J. 数据挖掘: 概念与技术[M]. 3版. 范明, 译. 北京: 机械工业出版社,

2012: 293 - 297.)

- [2] PARK H S, JUN C H. A simple and fast algorithm for K-medoids clustering[J]. Expert Systems with Applications, 2008, 36(2): 3336 - 3341.
- [3] XIA N, SU Y, QIN X. Efficient K-medoids clustering algorithm[J]. Application Research of Computers, 2010, 27(12): 4517 - 4519. (夏宁霞, 苏一丹, 覃希. 一种高效的K-medoids聚类算法[J]. 计算机应用研究, 2010, 27(12): 4517 - 4519.)
- [4] DING S F, XU L, ZHU H, et al. Research and progress of cluster algorithms based on granular computing[J]. International Journal of Digital Content Technology and its Applications, 2010, 4(5): 96 - 104.
- [5] MA Q, XIE J. New K-medoids clustering algorithm based on granular computing[J]. Journal of Computer Applications, 2012, 32(7): 1973 - 1977. (马箐, 谢娟英. 基于粒计算的K-medoids聚类算法[J]. 计算机应用, 2012, 32(7): 1973 - 1977.)
- [6] PARDESHI B, TOSHNIWAL D. Improved K-medoids clustering based on cluster validity index and object density[C]// Proceedings of the 2nd IEEE International Advance Computing Conference. Piscataway: IEEE, 2010: 379 - 384.
- [7] LAI P S, FU H C. Variance enhanced K-medoids clustering[J]. Expert Systems with Applications, 2010, 38(1): 764 - 775.
- [8] GAO D Y, YANG B R. An improved K-medoids clustering algorithm[C]// Proceedings of the 2nd International Conference on Computer and Automation Engineering. Piscataway: IEEE, 2010: 132 - 135.
- [9] BARIONI C N M, RAZENTE H L, TRAINA A J M, et al. Acceleration K-medoids-based algorithms through metric access method[J]. Journal of Systems and Software, 2008, 81(3): 343 - 355.
- [10] YAO L, LUO K, MENG Y. Clustering algorithm based on particle swarm optimization[J]. Computer Engineering and Applications, 2012, 48(13): 150 - 153. (姚丽娟, 罗可, 孟颖. 一种基于粒子群的聚类算法[J]. 计算机工程与应用, 2012, 48(13): 150 - 153.)
- [11] MENG Y, LUO K, YAO L, et al. K-medoids clustering algorithm method based on ACO[J]. Computer Engineering and Applications, 2012, 48(16): 136 - 152. (孟颖, 罗可, 姚丽娟, 等. 一种基于ACO的K-medoids聚类算法[J]. 计算机工程与应用, 2012, 48(16): 136 - 152.)
- [12] LI L, LUO K, ZHOU B. K-medoids clustering algorithm based on improved artificial bee colony[J]. Computer Engineering and Applications, 2013, 49(16): 146 - 150. (李莲, 罗可, 周博翔. 一种改进人工蜂群的K-medoids聚类算法[J]. 计算机工程与应用, 2013, 49(16): 146 - 150.)
- [13] WANG G, ZHANG Q, HU J. An overview of granular computing[J]. Transactions on Intelligent Systems, 2007, 2(6): 8 - 26. (王国胤, 张清华, 胡军. 粒计算研究综述[J]. 智能系统学报, 2007, 2(6): 8 - 26.)
- [14] WANG L. Study of granular analysis in clustering[J]. Computer Engineering and Applications, 2006, 42(5): 29 - 31. (王伦文. 聚类的粒度分析[J]. 计算机工程与应用, 2006, 42(5): 29 - 31.)
- [15] XU L, DING S. Research on granularity clustering algorithms[J]. Computer Science, 2011, 38(8): 25 - 28. (徐丽, 丁世飞. 粒度聚类算法研究[J]. 计算机科学, 2011, 38(8): 25 - 28.)