面向软件缺陷数据聚类分析的数据降维处理方法①

万 琳, 范秋灵

(装甲兵工程学院 信息工程系, 北京 100072)

摘 要: 面向软件缺陷数据的聚类分析就是按照一定的准则将不同的软件缺陷数据对象划分为多个类, 使得类内的缺陷数据相似, 类间的缺陷数据相异, 其意义在于发现软件缺陷的分布规律, 有针对性地制定测试方案, 优化测试过程. 针对传统 K-Means 方法聚类结果依赖样本初始空间分布的问题, 提出一种基于 PSO 算法的数据降维处理方法 DRPS. 仿真实验表明, 经过该方法降维处理后数据的聚类准确率及聚类质量都有了一定程度的提高.

关键词: 软件缺陷; 模糊等价矩阵; 聚类分析; 粒子群优化; 数据降维

Data Dimensionality Reduction Method Oriented to Software Defect Data Clustering Analysis

WAN Lin , FAN Qiu-Ling

(Information Engineering Department, Academy of Armored Forces Engineering, Beijing 100072, China)

Abstract: Clustering analysis oriented to software defect data is dividing different software defect data to different clusters according to some criterion. The result of clustering is that defect data in the same cluster is similar while defect data in different clusters is different. It is significant to find the distribution law of software defect make testing scheme and optimize testing process. Due to that the clustering results of K-Means is dependent on distribution of samples, a data dimensionality reduction method based on PSO is proposed. Simulation experiment shows that the clustering accuracy and quality are improved to some extent.

Key words: software defect; fuzzy equivalent matrix; clustering analysis; particle swarm optimization; data dimensionality

聚类分析[1]是数据挖掘领域中一个非常重要的研究内容, 其目的是将数据划分为有意义的若干个类别. 近年来, 软件测试技术飞速发展, 各测评机构通过执行大量的测试任务, 已积累了海量高复杂性的软件缺陷数据. 通过对软件缺陷数据进行聚类分析, 可以发现软件缺陷的总体分布情况, 从而得出软件产品的问题分布规律, 指导软件测试过程.

常用的聚类分析方法是 K-Means 方法, 该方法简单高效, 且伸缩性较好, 易于理解, 但其对初始聚类样本的分布情况比较敏感. 文献 2 表明: 对于 2 维以下的球形样本数据, K-Means 方法的聚类结果相对稳定, 而对于复杂的高维样本数据, 其聚类结果很不稳定. 近年来国内外专家学者也提出了一些新的聚类方法[3-5], 但这些方法仍然没有改变聚类结果对初始样本空间分布情况的依赖性, 有的甚至不惜牺牲数据质量对数据进行压缩, 严重影响聚类效果.

本文提出了一种基于 PSO 算法的数据降维处理方法(A Data Dimensionality Reduction Method Based on PSO, DRPS), 其主要思想是首先对数据进行离散化, 根据离散化后的数据构造其模糊等价矩阵, 并引入 PSO 方法对随机分布的二维坐标进行优化, 使二维样本之间的距离趋于它们之间的模糊等价关系值, 最终得到与原始多维数据等价的二维数据, 从而达到数据降维的目的. 仿真实验表明, 该方法能够较好地降低聚类结果对初始样本数据分布情况的依赖性, 实现任意形状样本簇的聚类, 是可行和有效的.

① 收稿时间:2014-07-04;收到修改稿时间:2014-09-09

1 模糊等价关系构建

为了更清晰、明了地阐述该方法, 在此从 XXX 装备信息管理软件测试过程中产生的软件缺陷数据中提取部分缺陷数据作为实验数据, 采用 DRPS 方法对软件缺陷数据进行降维处理, 实验数据如表 1 所示.

表 1　软件缺陷数据表

编号	缺陷标识	测试类型	缺陷类型	级别
1	WBG-001-1	边界测试	程序问题	一般
2	WBG-002-1	文档审查	文档问题	建议
3	WBG-003-1	功能测试	设计问题	重要
4	WBG-002-2	文档审查	文档问题	建议
5	WBG-003-2	功能测试	程序问题	严重
6	WBG-003-3	功能测试	设计问题	严重
7	WBG-001-2	边界测试	程序问题	一般
8	WBG-002-3	文档审查	文档问题	建议
9	WBG-001-3	边界测试	程序问题	建议
10	WBG-003-4	功能测试	设计问题	严重
11	WBG-001-4	边界测试	程序问题	一般
12	WBG-002-4	文档审查	文档问题	建议

1.1 数据离散化

软件缺陷数据中涉及的数据类型多种多样, 有汉字、字符、整数等等, 复杂多变的数据使得抽象知识的提取非常困难, 很大程度上限制了数据的实用性. 因此, 在进行数据分析之前, 首先要对数据进行离散化编码, 将之转换为可以挖掘的数据类型. 在此, 用信息表 S 表示待挖掘的数据集 T, 则其数学表示形式为多元参数组合 $S=\langle U,A,V,F\rangle$, 各参数含义如表 2 所示.

表 2　信息表 S 各参数含义对照表

序号	参数名称	含义
1	U	待挖掘数据的论域
2	A	待挖掘数据的所有项目集合
3	V	$V=\bigcup_{P\in A}V_P$, V_p 为项目 P 的值域
4	f	$f:U\times A\to V$

根据 V_p 的不同取值类型, 采取不同的编码方式对数据集 T 中的数据进行离散化处理:

1)若 V_p 的取值是有限的、离散的, 且 $0<|V_P|<\infty$, 则将这些取值分别编码为“1”, “2”, ……;

2)若 V_p 的取值是连续的或者无限的, 则将采用等距离划分方法将这些取值划分为有限个区间, 并分别编码为“1”, “2”, …;

3)若某项目取值为空或者某事务不包含该项目, 则编码为“0”.

以表 1 数据为例, 对表中的“测试类型”、“缺陷类型”、“缺陷级别”三个属性分别进行编码, 如表 3 所示.

表 3　缺陷属性编码表

缺陷属性	属性取值	编码
测试类型	文档测试	1
	边界审查	2
	功能测试	3
缺陷类型	程序问题	1
	文档问题	2
	设计问题	3
缺陷级别	严重	1
	重要	2
	一般	3
	建议	4

则离散化后的缺陷数据如表 4 所示.

表 4　离散化缺陷数据表

缺陷编号	测试类型	缺陷类型	缺陷级别
1	2	1	3
2	1	2	4
3	3	3	2
4	1	2	4
5	3	1	1
6	3	3	1
7	2	1	3
8	1	2	4
9	2	1	4
10	3	3	1
11	2	1	3
12	1	2	4

1.2 数据矩阵构建

经过数据离散化处理之后, 杂乱无章的数据结构趋于统一. 为了获得该数据的模糊等价矩阵进行降维, 首先必须构建其数据矩阵.

设项集 $\mathrm{I}=\{i_1,i_2,i_3,\cdots,i_d\}$ 是 d 个项目的集合, $T=\{t_1,t_2,t_3,\cdots,t_N\}$ 是 N 个事务的集合, 则对矩阵 A 的每一个元素$\{a_{ij}\}$进行如下定义:

$$a_{ij}=p_{ij} \tag{1}$$

其中, p_{ij} 为项目 P 的编码值, i=1,2,3,…d, j=1,2,3…,N, 矩阵的每一列表示一个事务(对应于软件缺陷数据的一个缺陷), 每一行表示一个项目(对应于软件缺陷数据的一个缺陷属性).

则表 4 数据依据上述方法生成的数据矩阵为:

$$A=\begin{pmatrix}2&1&3&1&3&3&2&1&2&3&2&1\\1&2&3&2&1&3&1&2&1&3&1&2\\3&4&2&4&1&1&3&4&4&1&3&4\end{pmatrix} \tag{2}$$

1.3 模糊矩阵建立

1) 数据归一化

为了避免后续数据处理过程中某一维或某几维数

据对聚类精度的影响, 加快程序的收敛速度, 首先需要对数据矩阵进行归一化处理, 即在不改变原始数据属性特征的前提下, 将样本数据的取值空间收缩到[0, 1]. 归一方法如下:

$$s_{ij} = \frac{a_{ij} - a_i^{\min}}{a_i^{\max} - a_i^{\min}} \tag{3}$$

其中, $a_i^{\min}$ 为矩阵 A 第 i 行元素的最小值, $a_i^{\max}$ 为矩阵 A 第 i 行元素的最大值.

以式 1-2 为例, 得到归一矩阵 S 为:

$$S = \begin{pmatrix} 0.50 & 0.00 & 1.00 & 0.00 & 1.00 & 1.00 & 0.50 & 0.00 & 0.50 & 1.00 & 0.50 & 0.00 \\ 0.00 & 0.50 & 1.00 & 0.50 & 0.00 & 1.00 & 0.00 & 0.50 & 0.00 & 1.00 & 0.00 & 0.50 \\ 0.67 & 1.00 & 0.33 & 1.00 & 0.00 & 0.00 & 0.67 & 1.00 & 1.00 & 0.00 & 0.67 & 1.00 \end{pmatrix} \tag{4}$$

2) 模糊相似矩阵

模糊相似矩阵[6]是用于存储不同聚类样本之间相似度的 N 维对称矩阵, 取值范围为[0, 1], 本文中利用它来表示数据矩阵中的样本数据之间相似度. 因此, 得到归一化矩阵 S 后, 用最小最大法按照如下方式建立模糊相似矩阵 r:

$$r_{ij} = \sum_{k=1}^{d} (s_{ik} \wedge s_{jk}) / \sum_{k=1}^{d} (s_{ik} \vee s_{jk}) \tag{5}$$

其中, i=1,2,3,…N, j=1,2,3…,N 根据式 1-4 建立的模糊相似矩阵为:

$$r = \begin{pmatrix} 1.00 & 0.63 & 0.56 & 0.63 & 0.57 & 0.44 & 1.00 & 0.63 & 0.86 & 0.44 & 1.00 & 0.63 \\ 0.63 & 1.00 & 0.50 & 1.00 & 0.33 & 0.40 & 0.63 & 1.00 & 0.75 & 0.40 & 0.63 & 1.00 \\ 0.56 & 0.50 & 1.00 & 0.50 & 0.63 & 0.88 & 0.56 & 0.50 & 0.50 & 0.88 & 0.56 & 0.50 \\ 0.63 & 1.00 & 0.50 & 1.00 & 0.33 & 0.40 & 0.63 & 1.00 & 0.75 & 0.40 & 0.63 & 1.00 \\ 0.57 & 0.33 & 0.63 & 0.33 & 1.00 & 0.71 & 0.57 & 0.33 & 0.50 & 0.71 & 0.57 & 0.33 \\ 0.44 & 0.40 & 0.88 & 0.40 & 0.71 & 1.00 & 0.44 & 0.40 & 0.40 & 1.00 & 0.44 & 0.40 \\ 1.00 & 0.63 & 0.56 & 0.63 & 0.57 & 0.44 & 1.00 & 0.63 & 0.86 & 0.44 & 1.00 & 0.63 \\ 0.63 & 1.00 & 0.50 & 1.00 & 0.33 & 0.40 & 0.63 & 1.00 & 0.75 & 0.40 & 0.63 & 1.00 \\ 0.86 & 0.75 & 0.50 & 0.75 & 0.50 & 0.40 & 0.86 & 0.75 & 1.00 & 0.40 & 0.86 & 0.75 \\ 0.44 & 0.40 & 0.88 & 0.40 & 0.71 & 1.00 & 0.44 & 0.40 & 0.40 & 1.00 & 0.44 & 0.40 \\ 1.00 & 0.63 & 0.56 & 0.63 & 0.57 & 0.44 & 1.00 & 0.63 & 0.86 & 0.44 & 1.00 & 0.63 \\ 0.63 & 1.00 & 0.50 & 1.00 & 0.33 & 0.40 & 0.63 & 1.00 & 0.75 & 0.40 & 0.63 & 1.00 \end{pmatrix} \tag{6}$$

3) 模糊等价矩阵

上述方法获得的是模糊相似矩阵, 不满足传递性, 不能如实反映原始数据的聚类特征, 因此, 还需要将矩阵 r 改造成模糊等价矩阵 t. 在本方法中, 采用传递闭包法建立模糊等价矩阵 t, 即按照最短路径原则通过 t 不断自乘的方法寻求两个变量之间的密切关系. 具体处理过程是: 将 t 不断自乘, 直到满足 $t^{2k} = t^k = t$ 为止, 这样便建立了模糊等价矩阵 t.

以式(1)-(6)为例, 本方法建立的模糊等价矩阵为:

$$t = \begin{pmatrix} 1.00 & 0.75 & 0.57 & 0.75 & 0.57 & 0.57 & 1.00 & 0.75 & 0.86 & 0.57 & 1.00 & 0.75 \\ 0.75 & 1.00 & 0.57 & 1.00 & 0.57 & 0.57 & 0.75 & 1.00 & 0.75 & 0.57 & 0.75 & 1.00 \\ 0.57 & 0.57 & 1.00 & 0.57 & 0.71 & 0.88 & 0.57 & 0.57 & 0.57 & 0.88 & 0.57 & 0.57 \\ 0.75 & 1.00 & 0.57 & 1.00 & 0.57 & 0.57 & 0.75 & 1.00 & 0.75 & 0.57 & 0.75 & 1.00 \\ 0.57 & 0.57 & 0.71 & 0.57 & 1.00 & 0.71 & 0.57 & 0.57 & 0.57 & 0.71 & 0.57 & 0.57 \\ 0.57 & 0.57 & 0.88 & 0.57 & 0.71 & 1.00 & 0.57 & 0.57 & 0.57 & 1.00 & 0.57 & 0.57 \\ 1.00 & 0.75 & 0.57 & 0.75 & 0.57 & 0.57 & 1.00 & 0.75 & 0.86 & 0.57 & 1.00 & 0.75 \\ 0.75 & 1.00 & 0.57 & 1.00 & 0.57 & 0.57 & 0.75 & 1.00 & 0.75 & 0.57 & 0.75 & 1.00 \\ 0.86 & 0.75 & 0.57 & 0.75 & 0.57 & 0.57 & 0.86 & 0.75 & 1.00 & 0.57 & 0.86 & 0.75 \\ 0.57 & 0.57 & 0.88 & 0.57 & 0.71 & 1.00 & 0.57 & 0.57 & 0.57 & 1.00 & 0.57 & 0.57 \\ 1.00 & 0.75 & 0.57 & 0.75 & 0.57 & 0.57 & 1.00 & 0.75 & 0.86 & 0.57 & 1.00 & 0.75 \\ 0.75 & 1.00 & 0.57 & 1.00 & 0.57 & 0.57 & 0.75 & 1.00 & 0.75 & 0.57 & 0.75 & 1.00 \end{pmatrix} \tag{7}$$

2 基于PSO算法的数据降维

获得模糊等价矩阵 t 后, 便获得了不同样本之间的模糊等价关系. 如果能找到满足不同样本之间的距离 d_{ij} 与它们之间的模糊等价关系 t_{ij} 之差 c_{ij} 最小的二维

数据, 则这些二维数据在表达数据样本的聚类关系方面与原始高维复杂数据是等价的, 从而可以利用这些二维数据的聚类分析结果代替原始多维复杂数据的聚类分析结果. 本方法引入 PSO 算法进行二维数据的寻优, 使之逼近模糊等价矩阵 t, 实现高维度数据的降维.

PSO 算法[7]是一种源于鸟群行为的群智能搜索方法, 可以通过大量粒子在特定维度的空间内按一定的速度和方向飞行得到目标函数的最优值. 该方法简单高效、易于实现, 具有较好的全局搜索性能, 可以用于寻找样本最优二维坐标值. 但是, PSO 算法中的粒子结构及适应度函数都需要根据数据样本的特点及搜索目标进行设计.

2.1 粒子结构设计

在本方法中, 我们希望得到的是最接近数据样本的模糊矩阵值的最优二维坐标, 比较的基础是样本数据的距离值, 以实数表示最为合适, 因此, 粒子结构采用实数形式进行如下编码: $x_{11}x_{12}x_{21}x_{22}\cdots x_{i1}x_{i2}$, 其中, i=1,2,3,…N, N 为样本的个数, $x_{i1}x_{i2}$ 分别为样本 i 的横坐标和纵坐标.

以表 1 所示的软件缺陷数据为例, 样本个数 N=12, 算法初始数据采用随机方法产生. 由于待寻优的二维数据的目的仅仅是为了能够代替原始空间数据进行聚类分析, 故我们关心的只是数据的分布特征, 而二维数据产生的区间不会影响其分布特征, 因此可根据具体的样本数据量确定随机初始数据的区间. 此例中样本的个数较少, 将随机区间定为(0, 10), 则粒子 i 的初始随机编码如图 1 所示.

x_{11}	x_{12}	x_{21}	x_{22}	x_{31}	x_{32}	x_{41}	x_{42}	x_{51}	x_{52}	x_{61}	x_{62}	x_{71}	x_{72}	x_{81}	x_{82}	x_{91}	x_{92}	x_{101}	x_{102}	x_{111}	x_{112}	x_{121}	x_{122}
0.588	7.632	0.653	6.941	8.587	2.888	8.453	3.159	8.090	7.071	0.664	5.826	7.111	0.911	3.289	4.997	4.725	7.142	2.113	4.547	1.466	7.190	4.231	9.141

图 1　粒子 i 编码结构图

2.2 适应度函数设计

PSO 算法中的适应度函数 f(x)是为了评价粒子群中的每个个体的优良度. 本文中, 寻优的目的是寻找最接近模糊等价矩阵的二维样本, 因此衡量粒子优良度的标准即可设计为二维样本之间的距离 d_{ij} 与其对应的模糊等价关系值 t_{ij} 的逼近程度, 见式(7), f(x)越小, 则说明该粒子的降维效果越好.

$$f(x)=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|d_{ij}-t_{ij}\right| \tag{8}$$

其中, t_{ij} 为模糊等价矩阵 t 中的关系值, d_{ij} 为样本 $i(x_i,y_i)$和样本 $j(x_j,y_j)$之间的二维距离, 定义如下:

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \tag{9}$$

以图 1 中粒子为例, 样本 1 和样本 2 之间的二维距离为:

$$d_{12}=\sqrt{(0.653-0.588)^2+(6.941-7.632)^2}=0.69 \tag{10}$$

其他样本之间的二维距离计算方法同式(10), 样本之间的模糊等价关系值见式(7), 则以 f(x)评价该粒子的适应度值为:

$$\begin{aligned}f(x)&=\frac{1}{12}\sum_{i=1}^{12}\sum_{j=1}^{12}\left|d_{ij}-t_{ij}\right|\\&=\frac{1}{12}(\left|d_{11}-t_{11}\right|+\left|d_{12}-t_{12}\right|+\cdots+\left|d_{112}-t_{112}\right|+\left|d_{21}-t_{21}\right|+\cdots\left|d_{1212}-t_{1212}\right|)\\&=\frac{1}{12}(\left|0\text{-}1\right|+\left|0.69\text{-}0.75\right|+\cdots+\left|3.93\text{-}0.75\right|+\left|0.69\text{-}0.75\right|+\cdots\left|0\text{-}1\right|)=48.5408\end{aligned} \tag{11}$$

2.3 PSO 算法寻优结果

PSO 算法原理描述如下: 一个由 m 个粒子组成的种群在 d 维搜索空间中以一定的速度飞行, 每个粒子 i 都有位置和速度两个属性, 其位置信息表示 d 个样本的一种二维坐标组合, m 个粒子则表示 m 种可能的坐标组合, 每种坐标组合的优良程度通过粒子的适应度函数来衡量, 在粒子飞行过程中, 每个粒子 i 都根据个体最优点 p_{best} 和群体最优点 g_{best} 这两个极值进行迭代, 并按文献 8 中所述的粒子更新方式更新自己的速度和位置, 最终输出群体最优点 g_{best} 的位置信息, 即最优

二维坐标组合.

按照 2.1、2.2 节所述设计应用 PSO 算法, 通过对 n 个初始粒子群的多代寻优之后, 对于表 1 所示的缺陷数据(高维样本数据), 得到其所对应的最优二维坐标值如图 2 所示.

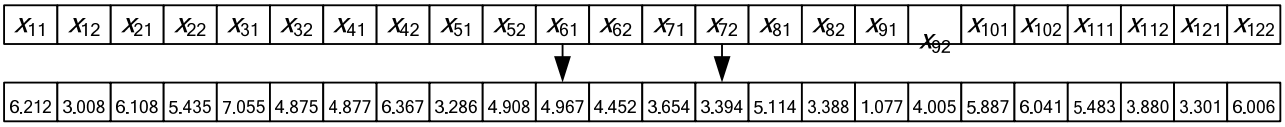


图 2 样本最优二维坐标值

2.4 聚类分析结果

为了了解 DRPS 方法的效果, 我们基于前述例子的原始高维数据和降维后的二维数据分别进行了 K-Means 聚类分析, 结果对比如下:

1) 降维后的二维数据聚类结果

聚类结果集为: X_1=(1,7,9,11), X_2=(2,4,8,12), X_3=(3,5,6,10);

依据表 1 可看出, 聚类边界较清晰, X_1 类为边界测试、程序问题、一般, X_2 类为文档审查、文档问题、建议, X_3 类为功能测试、设计问题、严重. 该聚类结果说明:

① 在对 XXX 装备信息管理软件进行边界测试时发现的程序问题多为一般性的问题, 如日期没有上下限制等, 对软件的实现没有多大影响;

② 文档审查时发现的文档问题多是建议性的, 如文档的格式、书写没有按照规定的要求;

③ 进行功能测试时, 一旦发现设计出了问题, 那一般就是比较严重的, 会影响软件的功能实现.

2) 原始高维数据聚类结果

聚类结果集为: X_1=(1,5,7,11), X_2=(2,4,8,9,12), X_3=(3,6,10);

对照表 1 可看出, 聚类边界模糊, X_1 类中既有边界测试, 又有功能测试; X_2 类中既有文档审查, 又有边界测试; X_3 类虽然全部为功能测试, 但是并不全面.

相比较而言, 原始高维数据的聚类结果很不稳定, 具有一定的局限性, 而经过数据降维处理之后的二维数据得到的聚类结果比较稳定, 且相对准确, 具有更广泛的指导作用.

3 实验及结果分析

为了验证本文提出的数据降维处理方法的有效性与可行性, 笔者用 Matlab2012a 进行了仿真实验, 模拟实现了 K-Means 方法, 并对不同维度数据样本的实验结果进行了比较.

为了体现对比结果的可信性, 本文选取 UCI 机器学习数据库中的 Iris 和 Wine 经典数据集作为仿真实验数据. Iris 数据集是 150 个植物数据样本, 分 3 个类别, 每个类别有 50 个样本, 每个样本有 4 个属性值, 如表 5 所示. Wine 数据集是 178 个葡萄酒数据样本, 分 3 个类别, 每个类别的样本个数分别为 59, 71, 48, 每个样本有 13 个属性值, 如表 6 所示. 实验中各个参数取值如下: 粒子群种群规模 m=30, 迭代次数 t=30, c_1=c_2=2.05, ω_{max}=0.96, ω_{min}=0.01, 聚类个数 K=3.

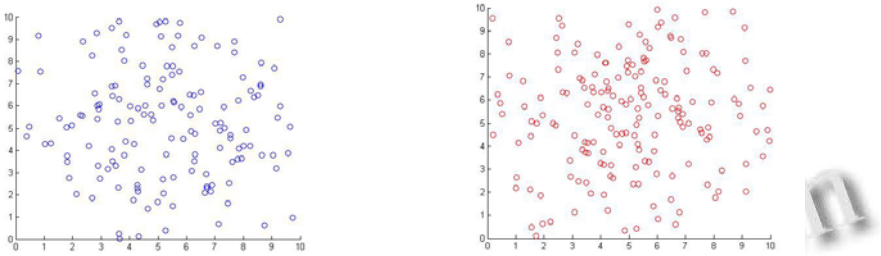
表 5 Iris 样本数据

编号	属性			
	1	2	3	4
1	5.1	3.5	1.4	0.2
2	4.9	3	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5	3.6	1.4	0.2
…	…	...	...	...
149	6.2	3.4	5.4	2.3
150	5.9	3	5.1	1.8

表 6 Wine 样本数据

编号	属性				
	1	2	3	…	13
1	14.20	1.71	2.43	…	1065
2	13.20	1.78	2.14	…	1050
3	13.16	2.36	2.67	…	1185
4	14.37	1.95	2.50	…	1480
5	13.24	2.59	2.87	…	735
…	…	...		...	...
149	13.17	2.59	2.37	…	840
150	14.13	4.10	2.74	…	560

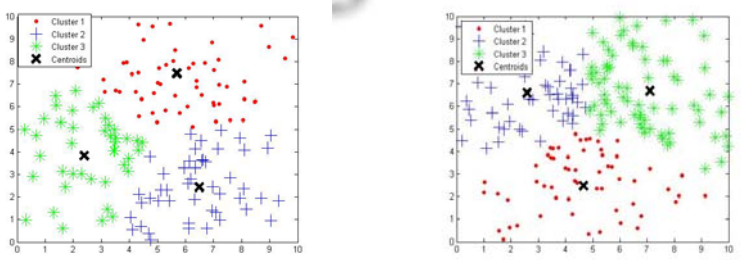
根据第 2 节所述方法进行数据降维后得到的样本二维坐标分布如图 3 所示:



(a)Iris 样本二维坐标分布图　　(b)Wine 样本二维坐标分布图

图 3　样本二维坐标分布图

以此为基础, 进行 K-Means 聚类得到的最终聚类结果如图 4 所示:



(a)Iris 样本聚类结果图　　(b)Wine 样本聚类结果图

图 4　样本聚类结果图

不同维度数据的聚类正确率如表 7 所示.

表 7　不同维度数据的聚类正确率比较

数据维度		高维	二维
		正确率(%)	
Iris	最优值	88.93	91.01
	平均值	81.03	89.34
	最差值	53.22	88.22
Wine	最优值	72.93	76.13
	平均值	69.07	75.72
	最差值	58.22	75.36

衡量样本数据与聚类中心距离的指标 J(C)值如表 8 所示.

表 8　不同维度数据的 J(C)值比较

数据维度		高维	二维
Iris	最优值	97.33	96.85
	平均值	103.42	97.06
	最差值	123.97	97.84
Wine	最优值	16860.33	16946.85
	平均值	17201.42	16960.06
	最差值	18325.97	16970.84

实验结果分析如下:

1) 从表 7 可以看出, 降维之后数据聚类准确率有所提高, 聚类结果比较稳定, 表明数据降维处理可以有效减少聚类结果对样本分布情况的依赖.

2) J(C)是衡量聚类质量的一个比较有效的标准, 它评价的是聚类内部的紧密程度, J(C)值越小, 说明得到的类内部的数据之间相似度越高, 聚类的质量越高. 由表 8 的 J(C)值可以得出, 降维之后的数据聚类质量较好.

4　结语

针对 K-Means 方法聚类结果对样本空间分布情况比较敏感的问题, 本文提出了一种基于 PSO 的数据降维处理方法. 该方法通过构建模糊等价矩阵获得高维样本数据之间的模糊等价关系, 并通过 PSO 方法对随机初始化的二维坐标进行优化, 实现高维数据的降维, 消除聚类结果对样本初始分布情况的依赖. 仿真实验表明, 与高维数据的聚类结果相比, 降维之后的数据

聚类准确率有所提高, 聚类质量有所改善. 但是, 如何提高该方法的效率, 降低复杂度, 是下一步的研究内容.

参考文献

1 Chakrabarti S. Mining the Web: Discovering knowledge from hypertext data. China: Posts & Telecom Press, 2009.

2 Mahajan M, Nimbor P, Varadarajan K. The planar K-means problem is NP-hard. Lecture Notes in Computer Science, 2009, 5431: 274–285.

3 胡艳维,秦拯,张忠志.基于模拟退火与K均值聚类的入侵检测算法.计算机科学,2010,6(6):122–124.

4 何登旭,曲良东.一种新的混合聚类分析算法.计算机应用研究,2009,26(3):879–880.

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

6 钱伟强.一种基于改进粒子群和 K 均值结合的聚类算法[学位论文].西安:西安电子科技大学,2011.

7 刘清.Rough 集及 Rough 推理.北京:科学出版社,2001.

8 汪定伟,王俊伟,王洪峰.智能优化方法.北京:高等教育出版社,2007.