文章编号: 1001-0920 (2012) 04-0603-06

基于聚类融合的混合属性数据增量聚类算法

李桃迎, 陈 燕, 张金松, 秦胜君

(大连海事大学 交通运输管理学院，辽宁 大连 116026)

摘 要: 针对传统增量聚类方法对混合属性数据聚类时存在不稳定、随机性大和准确性不够高的缺点, 提出一种基于聚类融合的混合属性数据增量聚类算法. 该算法以传统增量聚类为基础, 采用多种聚类算法的结果进行融合来代替原有单一划分, 并重新修正了阈值的取值范围. 实验表明, 所提出的算法利用原有数据的特征, 提高了聚类的稳定性和精确性, 具有很好的聚类效果.

关键词: 聚类融合；增量聚类；矢量量化；模糊 k-均值

中图分类号: TP391.4 **文献标识码:** A

Incremental clustering algorithm of mixed numerical and categorical data based on clustering ensemble

LI Tao-ying, CHEN Yan, ZHANG Jin-song, QIN Sheng-jun

(Transportation Management College，Dalian Maritime University，Dalian 116026，China．Correspondent：LI Tao-ying，E-mail：ytaoli@126.com)

Abstract: Traditional clustering methods have disadvantages of unsteadiness, randomness and low accuracy for classifying mixed numerical and categorical data. Therefore, the incremental clustering algorithm of mixed numerical and categorical data based on clustering ensemble is proposed, which adopts the results of several clustering to replace that of single clustering and modifies the design of threshold. The experiment results show that the improved algorithm has higher stability and accuracy by using the characters of existing data, and possess better effectiveness.

Key words: clustering ensemble；incremental clustering；vector quantization；fuzzy k-means

1 引 言

聚类就是将数据点划分成组, 同时满足组内数据点之间的相似性尽可能大, 不同组的数据点之间的相似性尽可能小[1]. 聚类在数据挖掘中具有非常重要的作用, 已广泛应用于模式识别、计算机可视化、模糊控制等领域. 但是, 随着Web技术的出现, 数据和环境无时无刻不在发生变化, 传统的聚类算法已不再适应动态变化的数据, 为此人们提出了增量聚类[2]. 自从Hartigan在文献[3]中提出的算法得以实现[4]以来, 增量聚类便吸引了众人的关注. 增量聚类本质上是维持或者改变 k 个簇的结构问题. Ester等人[5]提出的增量DBSCAN是在DBSCAN的基础上提出的, 因DBSCAN算法具有基于密度的特性, 所以插入或删除一个新的数据点只影响当前聚类中近邻该点的簇. 这种方法的优点是它的聚类结果与非增量聚类的结果相似, 不足之处是只能一个一个地划分数据点, 使得聚类的效率很低. 陈宁等人[6]提出了与增量DBSCAN比较类似的基于网格的增量聚类. Chen等人[6-7]采用批量处理的基于密度的增量聚类, 以批量的形式处理数据, 克服了一个一个处理数据的缺点, 但这种聚类方法由于计算量过大而不能用于大数据集. 刘建晔等人[8]描述了一种高效的基于密度的增量聚类算法, 利用划分和抽样技术处理大数据集, 在划分高维数据时会产生抽样误差. 另外, Sebastian等人[9]采用了基于代表点的连接来处理动态数据流的增量聚类问题, Nikolai等人[10]采用批处理的方法进行大数据集的聚类.

目前, 相关文献中只有Hsu等人[2]针对混合属性数据的增量聚类进行了研究, 借用概念层次树来求解混合型数据的相似度, 但要求用户必须清楚地了解属性的值, 对任意属性(包括分类属性)的取值范围给出大小关系并设置有效的差值, 如若设置不当, 则会导致误差较大.

收稿日期: 2010-10-20；**修回日期:** 2011-01-05.

基金项目: 国家自然科学基金项目(70940008)；高等学校博士学科点专项科研基金项目(200801510001).

作者简介: 李桃迎(1983−), 女, 博士生, 从事数据挖掘、聚类分析等研究；陈燕(1952−), 女, 教授, 博士生导师, 从事数据挖掘、人工智能等研究.

鉴于传统增量聚类对分类属性或混合属性数据聚类时存在结果不稳定、随机性大、准确性不够高的缺点, 本文提出了基于聚类融合的混合数据增量聚类算法. 借鉴已获取数据点的特征, 采用矢量量化的方法划分以增量模式出现的数值型数据, 用分类属性具体值划分以增量模式出现的分类属性数据; 然后用聚类融合算法合并已产生的聚类成员, 以避免单一算法的不稳定性和随机性, 提高聚类的精度和效率.

2 聚类融合

假定数据点集为 $X=\{x_1,x_2,\cdots,x_n\}$, 每个数据点有 m 个属性, 第 i 个属性的权值为 w_i, 该属性有 k_i 个不同的取值, 从而可得到一个如下式所示的划分:

$$R_i=\{C_{i,1},C_{i,2},\cdots,C_{i,k_i}\},\ 1\leqslant i\leqslant m. \tag{1}$$

其中: R_i为根据第 i 个属性的取值得到的划分, $C_{(i,j)}$ 为第 i 个划分的第 j 个簇, 且 $\sum\limits_{i=0}^{m}w_i=1$.

采用不同的聚类算法或对同一算法设置不同的参数, 可以得到多个划分; 然后采用一定的融合策略将不同的划分结果融合成一个, 便实现了聚类融合.下面对分类属性和数值属性聚类成员的生成方法分别进行阐述.

2.1 分类属性聚类成员的产生

将各分类属性按不同的取值进行划分, 从而得到相应的聚类成员 (见表 1).

表 1 实 例

x	a_1	a_2	a_3	a_4
	$w_1=0.2$	$w_2=0.2$	$w_3=0.3$	$w_4=0.3$
x_1	Y	S	B	A
x_2	X	T	B	A
x_3	Y	S	B	C
x_4	X	S	D	A
x_5	Y	T	D	C

从表 1 可以看出, 根据各对象属性值的不同, 可以得到对应属性的划分分别为

$$R_1=\{\{1,3,5\},\{2,4\}\},\ R_2=\{\{1,3,4\},\{2,5\}\},$$

$$R_3=\{\{1,2,3\},\{4,5\}\},\ R_4=\{\{1,2,4\},\{3,5\}\}.$$

聚类成员中各数字分别表示对应的数据点, 因此分类属性的数目即为分类属性的聚类成员数. 处理分类属性的增量模式时, 只需在各属性原有划分中加入新增对象. 同时, 由文献 [11] 可知, 最佳聚类数满足 $k\leqslant\sqrt{n}$, 如果划分簇数不满足 $k\leqslant\sqrt{n}$, 则合并一些属性的取值, 从而保证聚类数不至于过大.

2.2 数值属性聚类成员的产生

数值属性的取值无法枚举, 所以需采用一些算法来获取聚类成员. 本文将数值属性聚类成员的产生分成两类: 一类是有数据基础, 即已经获得一定数目的数据对象, 且这些对象有一定的代表性, 在考虑增量的同时可以利用这些数据对象; 另一类是无数据基础, 即数据对象很少或不具有代表性, 聚类时只能考虑增量形式的数据对象. 下面分别进行讨论.

由于数值属性的取值范围不同, 在进行聚类之前需消除属性之间的量纲.

$$x_{ji}=\left|\frac{x_{ji}^{\text{original}}-\min\limits_t x_{ti}}{\max\limits_t x_{ti}-\min\limits_t x_{ti}}\right|,\ 1\leqslant i\leqslant m. \tag{2}$$

通过方程 (2) 可以消除属性的量纲, 同时将属性的取值范围控制在 $[0,1]$ 之间. 下文所有对数值属性的处理均在无量纲的情况下进行. 对于已有的待聚类的对象, 可采用最常见的聚类算法对数值属性进行聚类. 本文采用模糊 k 均值聚类算法, 其成本函数如下:

$$F(T,W,C)=\sum_{l=1}^{k}\left(\frac{\sum\limits_{j=1}^{n}\sum\limits_{i=1}^{m}\tau_{lj}\omega_i(c_{li}-x_{ji})^2}{\sum\limits_{i=1}^{m}(c_{li}-\overline{x}_i)^2}\right)+\gamma\sum_{i=1}^{m}\omega_i\log\omega_i. \tag{3}$$

其中

$$\sum_{l=1}^{k}\tau_{lj}=1,\ 1\leqslant j\leqslant n,\ \tau_{lj}=\{0,1\};$$

$$\sum_{i=1}^{m}\omega_i=1,\ 1\leqslant\omega_i\leqslant 1;$$

k,n 和 m 分别为簇、对象和属性的数目; x_{ji} 为第 j 个对象的第 i 个属性的值; $C=[c_{li}]$ 为 $k\times m$ 矩阵; C_{lj} 为第 l 个簇中心的第 i 个属性的值; $T=[\tau_{lj}]$ 为 $k\times n$ 的矩阵, 且 τ_{lj} 为第 j 个对象属于第 l 个簇的隶属度; $W=[\omega_i]$ 为 m 维向量, 且 ω_i 为第 i 个属性的权值; γ 为大于 1 的参数; $\overline{x}$ 为所有对象的均值, $\overline{x}_i$ 为 $\overline{x}$ 的第 i 个属性的值, 即 $\overline{x}_i=\dfrac{1}{n}\sum\limits_{j=1}^{n}x_{ji}$, 如果 $n>1$, 则成本函数有效, 否则 $\sum\limits_{i=1}^{m}(c_{li}-\overline{x}_i)^2=0$ 导致 $F(T,W,C)$ 无法获得. 式 (3) 中的分母是一个变量, 并且与各类的均值以及所有类均值差的平方和线性相关.

F 的最小化包含一系列未知解决方法的条件非线性优化问题. 一般而言, 为了求取 T,W 和 C 的值, 需将优化问题转换成部分优化问题. 可借鉴文献 [1] 和 [12] 中的方法, 首先固定 T 和 C, 寻找使 $F(T,W,C)$ 最小的 W; 然后固定 T 和 W, 寻找最适当的 C; 最后固定 C 和 W, 寻找合适的 T. 重复上述步骤, 直到目标函数不再减少.

定理 1 T 和 C 的值保持不变时, F 将是最小值当且仅当下式成立:

$$\omega_t = \frac{\mathrm{e}^{\left(\frac{-\psi_t}{\gamma}-1\right)}}{\sum_{i=1}^{m}\mathrm{e}^{\left(\frac{-\psi_i}{\gamma}\right)}}, \tag{4}$$

其中

$$\psi_t = \sum_{l=1}^{k}\frac{\sum_{j=1}^{n}\tau_{lj}(c_{lt}-x_{jt})^2}{\sum_{i=1}^{m}(c_{li}-\overline{x}_i)^2}. \tag{5}$$

证明　采用文献[1,12]中的方法可以获得如下无条件的最小值优化问题:

$$\min F(\{\omega_i\},\varepsilon) = \sum_{l=1}^{k}\left(\frac{\sum_{j=1}^{n}\sum_{i=1}^{m}\tau_{lj}\omega_{li}(c_{li}-x_{ji})^2}{\sum_{i=1}^{m}(c_{li}-\overline{x}_i)^2}\right)+ \gamma\Big(\sum_{i=1}^{m}\omega_i\log\omega_i\Big)-\varepsilon\Big(\sum_{i=1}^{m}\omega_{li}-1\Big). \tag{6}$$

因为各簇之间是相互独立的, 所以

$$F(\omega_i,\varepsilon) = \frac{\sum_{j=1}^{n}\sum_{i=1}^{m}\tau_{lj}\omega_i(c_{li}-x_{ji})^2}{\sum_{i=1}^{m}(c_{li}-\overline{x}_i)^2}+ \gamma\Big(\sum_{i=1}^{m}\omega_i\log\omega_i\Big)-\varepsilon\Big(\sum_{i=1}^{m}\omega_i-1\Big). \tag{7}$$

$F(\omega_i,\varepsilon)$ 是可导的, 设置导数为零, 即

$$\frac{\partial F(\omega_i,\varepsilon)}{\partial\varepsilon} = \Big(\sum_{i=1}^{m}\omega_i-1\Big)=0, \tag{8}$$

并且

$$\frac{\partial F(\omega_t,\varepsilon)}{\partial\omega_t} = \sum_{l=1}^{k}\frac{\sum_{j=1}^{n}\tau_{lj}(c_{lt}-x_{jt})^2}{\sum_{i=1}^{m}(c_{li}-\overline{x}_i)^2}+ \gamma(1+\log\omega_{lt})-\varepsilon=0. \tag{9}$$

从式(9)可以得到

$$\omega_t = \mathrm{e}^{\left(\frac{-\psi_t+\varepsilon-\gamma}{\gamma}\right)}, \tag{10}$$

其中 ψ_t 如式(5)所示.

将式(10)代入(8)可得

$$\mathrm{e}^{\left(\frac{\varepsilon-\gamma}{\gamma}\right)} = \frac{1}{\sum_{i=1}^{m}\mathrm{e}^{\left(\frac{-\psi_i}{\gamma}\right)}}, \tag{11}$$

将式(11)代入(10)可得(4).

同理, 固定 W 和 C 可以求取 T 的值. 众所周之, 如果第 j 个对象到第 l 个簇的距离最小, 则它将属于第 l 个簇, 即

$$\tau_{lj} = \begin{cases} 1,\ \sum_{i=1}^{m}\omega_{li}(c_{li}-x_{ji})^2 \leqslant \sum_{i=1}^{m}\omega_{zi}(c_{zi}-x_{zi})^2; \\ 0,\ \text{otherwise.}\end{cases} \tag{12}$$

$\tau_{lj}=1$ 表示第 j 个对象完全属于第 l 个簇, 否则表示不属于第 l 个簇. 若固定 T 和 W, 则采用数学平均值的方法可获取 C 的值, 即

$$c_{li} = \frac{\sum_{j=1}^{n}\tau_{lj}x_{ji}}{\sum_{j=1}^{n}\tau_{lj}}. \tag{13}$$

由此定理1得证. □

在获得对现有数据对象的聚类之后, 对于增量的数据对象可采用基于聚类融合的增量聚类算法, 下面将进行详细介绍.

3　基于聚类融合的增量聚类算法

3.1　算法描述

事实上, 可以根据经验或专家意见获得聚类对不同属性的依赖程度, 从而获取固定的权值 w_i. 在对已有数据划分成 k 个簇之后, 可以将新增的数据点划分到已有簇中.

假定新的数据点为 x^*, 则可以根据式(12)和(13)获得新增数据点所属的簇及该簇更新后的中心. 这种方法只适用于历史数据点可以反映全局的情况, 在新增的数据点与历史数据点有明显差异时将不再适用, 这是因为已存在的簇只能反映历史数据点的特征而不能反映新增数据点的特征.

3.2　矢量量化的增量聚类模式

下面用改进的矢量量化的概念来划分新增的数据点. 传统矢量量化的扩展公式如下:

$$c_{li}^{\mathrm{new}} = c_{li}^{\mathrm{old}} + \mu(x_i^* - c_{li}^{\mathrm{old}}). \tag{14}$$

其中: c_{li}^{new} 为新的中心, c_{li}^{old} 为原中心, $\mu(x_i^*-c_{li}^{\mathrm{old}})$ 为发生的偏移.

式(14)不能反映整个数据集的分布. 文献[13]提出用矢量量化的扩展来划分新增数据点, 受此启发, 本文采用矢量量化并利用已知数据点划分新增的数据点. 当新增数据点 x^* 时, 首先计算它与所有现有簇中心的距离

$$d(x^*,c_l) = \sum_{i=1}^{m}\omega_{li}(x_i^*-c_{li})^2, \tag{15}$$

并假定 $d_{\min}(x^*,c_l)$ 为 $d(x^*,c_l)$ 的最小值.

If $d_{\min}(x^*,c_l)<\rho$ then

Set $n=n+1$, $\tau_{l^*}=1$

Else Set $k=k+1$, $n=n+1$, $c_k=x^*$

End if

依据上述描述可知, 让ρ的值尽量大以避免产生过多的簇, 同时让ρ的值尽可能小以避免产生的簇过少, 因此ρ的取值对于聚类非常关键. 下面讨论如何获取适当的ρ. 文献[13]中给出的获取公式如下:

$$\rho=\frac{0.3}{\sqrt{2}}\sum_{i=1}^{m}\sqrt{\omega_i}. \tag{16}$$

式(16)可以有效地解决文中所提出的问题. 但是, 其中的0.3和$\sqrt{2}$是如何获取的, 其原因不详, 如果能给出其证明过程则更佳.

本文充分利用已知数据点, 给出了ρ有意义且简单的形式

$$\rho=\frac{1}{n}\sum_{j=1}^{n}\tau_{lj}d(x_j,c_l)\times\sum_{i=1}^{m}\sqrt{\omega_i}. \tag{17}$$

其中: ρ由两部分组成, 前一部分是n个数据点到它们各自所属簇中心距离的均值, 后一部分是m个权值的算术平方根之和.

如果所有的属性具有相同的权值, 即每个属性的权值都是$i=1/m$, 则式(17)可变形为

$$\rho=\frac{\sqrt{m}}{n}\sum_{l=1}^{k}\sum_{l=1}^{k}\tau_{lj}d(x_j,c_l), \tag{18}$$

其中ρ的值与属性的数目及簇内距离相关.

对于无数据基础的聚类, 由于没有历史数据的经验, 无法根据式(17)获取ρ的值, 只能采用式(16).

此时ρ的取值没有实际意义, 因此当聚类数不满足$k\leqslant\sqrt{n}$时, 可增大ρ的值, 以保证簇数不会过大.

3.3 聚类成员的融合

根据上述方法, 所有数值型属性只产生一个聚类成员, 即总的聚类成员数目是分类属性数加1. 实际上, 也可以对每个数值属性采用上述方法分别产生数据成员, 这样得到的总的聚类成员数将是数据对象的属性数目. 由于方法相同, 本文不再赘述.

不失一般性, 现假设有m个聚类成员, 其中$m-1$个是分类属性产生的聚类成员, 每个成员含有其属性所对应的权重, 一个是数值属性产生的聚类成员, 其权重为产生该聚类成员的所有数值属性的权值之和, 所以满足$\sum_{i=1}^{m}=1$.

此时, 设置阈值$\theta\,(0<\theta\leqslant1)$的值, 寻找所有聚类成员的权值之和大于或等于$\theta$的所有聚类成员组合. 首先设置$\Pi$为空集, 对于任意一个组合, 不失一般性, 假设有t个聚类成员, 从每个聚类成员中各取一个簇, 它们的交集为π_i, 即

$$\pi_i=\overbrace{C_{1,j_1}\bigcap C_{2,j_2}\bigcap\ldots\bigcap C_{t,j_t}}^{t}.$$

如果π_i含有2个或2个以上元素, 则将其放入Π; 否则, 如果π_i为空集或只有一个元素, 则丢弃. 然后寻找下一个满足条件的组合, 直到所有满足条件的组合都遍历完毕.

Π中所有集合(假设集合数为s)的权值都大于或等于θ, 同时这些集合之间可能彼此存在交集. 不妨设$\pi(i,j)=\pi_i\bigcap\pi_j\neq\Phi$, 则将得到一个$s\times s$的三角稀疏矩阵$R$对于所有交集元素数不为零的集合. 将其按

$$\frac{|\pi(i,j)|}{\max\{|\pi_i|,|\pi_j|\}}$$

从大到小的顺序排列, 并设置阈值$\lambda\,(0<\lambda\leqslant1)$的值.

Step 1: 令$\alpha=\dfrac{|\pi(i,j)|}{\max\{|\pi_i|,|\pi_j|\}}$.

Step 2: If $\alpha\succeq\lambda$ then

合并i和j为一个集合π^*, 令$s=s-1$, 将π_i和π_j从Π中删除, 并将π^*添加到Π中, 同时删除R中π_i和π_j所在的行和列, 并添加π^*所在的行和列. 转Step 1.

Else 转Step 3.

End If

Step 3: If $\exists\,\pi(i,j)\neq\Phi$ then

If $|\pi_i|>|\pi_j|$ then $\pi_i=\pi_j-\pi(i,j)$

Else $\pi_j=\pi_i-\pi(i,j)$

End If

Else 转Step 4.

End If

Step 4: If $\exists\,x_i\notin\bigcup_{j=1}^{s}\pi_j$ then

将x_i添加到Π中, 转Step 4.

Else 结束运算.

End If

因为Π中包含所有数据点, 且任意集合间不存在交集, 所以是数据集的一个划分.

对于新增的数据点, 不是将所有数据重新进行聚类, 而是将新增的数据点按数值型属性将其添加到数值型属性产生的划分中. 同时调整该新增数据点所在簇的中心, 并按照分类属性的取值将其添加到分类属性所产生的聚类成员中.

传统聚类的时间复杂度为$O(mn^2)$, 与待聚类数据点数的数目成指数变化, 所以传统的聚类方法需要大量的计算. 当数据量很大时, 可能由于计算量过大而无法获得最终结果. 本文提出的算法时间复杂度与EWKM算法[1]的时间复杂度相同, 都是$O(mnk)$. 同时, 在已知数据点很多的情况下采用k-center初始化模糊k-means方法, 可以减少迭代聚类的次数.

当最终得到的聚类数不满足需求时, 需对结果簇进行合并或分裂. 可采用层次聚类中的分裂、合并

的方法来完成, 直到满足最终目标为止. 对于混合属性数据的增量聚类, 可对数值型属性数据取其均值作为中心点; 而对于分类型属性数据, 则选择出现次数最多的属性值作为中心. 在计算新增数据点与现有中心的距离时, 可根据融合策略对数值型属性采用式(15)的方法将其划入现有数值型属性产生的聚类成员中; 而对于分类型属性, 则将其划入相同取值出现最多的分类属性产生的聚类成员中, 然后再采用聚类的融合策略对聚类结果进行融合, 以完成对聚类结果簇的调整.

4　实例分析

下面采用 UCI 公共数据集中的 Diagnosis 数据集和 Adult 数据集来验证本文提出的基于聚类融合的混合属性增量聚类方法.

4.1　Diagnosis 数据集

Diagnosis 数据集含有 120 个数据对象, 每个对象有 6 个属性, 采用类判断的结果是都有膀胱炎和肾盂肾炎. 膀胱炎的主要症状是尿频、尿急、尿痛、排尿不适等, 而恶心和腰疼的症状相对而言不明显, 所以在判断是否患有膀胱炎时, 大都不采用是否恶心和腰疼这 2 个属性. 初始化算法时, 根据医学中各属性的影响因素, 将属性的权值根据真实情况赋值为 $\omega=\{0.5, 0.25, 0.1, 0.15\}$. 同时根据上述算法对温度这个唯一的数值属性进行聚类, 得到聚类成员按温度的划分{{1-60}, {61-68}, {69}, {70-120}}, 其中的数字表示第几个数据对象. 由于是否患有膀胱炎只能有 2 个取值, 即需要划分成 2 类, 同时结合尿频、尿急、尿痛 3 个属性得到的聚类成员, 利用聚类成员的融合和合并得到最终的聚类结果, 其中有 10 个病人的聚类结果有误, 准确率为 91.67%.

肾盂肾炎常伴随发热、畏寒、筋骨酸痛、头痛、恶心呕吐、食欲不振, 而且发热是肯定存在的症状, 所以在体温即低于 38°C 时肯定不会是肾盂肾炎, 从而只需对温度高于 38°C 的情况进行聚类. 此时, 剩余各属性的权值为 $\omega=\{0.05, 0.35, 0.2, 0.2, 0.2\}$, 最终得到的聚类结果中有 2 个病人的聚类结果有误, 准确率为 98.35%.

4.2　Adult 数据集

Adult 训练数据集有 32 561 个数据对象, 测试数据集有 16 281 个数据对象, 每个对象有 14 个属性, 在训练数据集和测试数据集中去掉含有缺省值的对象之后, 分别有 30 718 和 15 060 个数据对象. 由于各属性对收入的影响不同, 最后选择年龄、学历、职业、种族和每周的工作时间作为聚类的属性, 各属性的权值为 $\omega=\{0.2, 0.3, 0.2, 0.1, 0.2\}$. 同时, 考虑到学历和职位的取值较多, 根据实际情况, 认为 Prof-school, Masters 和 Doctorate 是高学历, 其他为低学历, Exec-managerial 和 Prof-specialty 被认为是高薪职位, 其他则为低薪职位. 最终得到的聚类结果中, 训练数据集和测试集的精确数为 26 373 和 12 939, 所对应的精度为 85.86%和 85.92%.

为了更形象地显示聚类的结果, 图 1 和图 2 以年龄和每周的工作时间为坐标显示了最终聚类的结果.

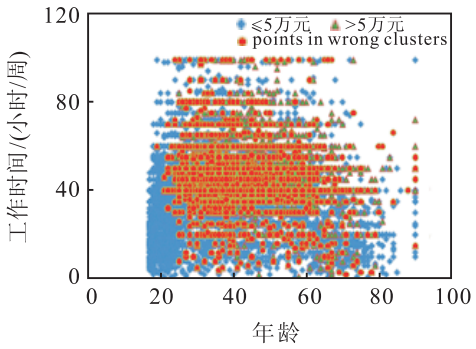

图 1　Adult 训练数据聚类结果

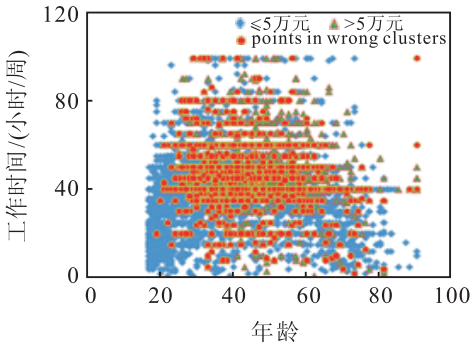

图 2　Adult 测试数据聚类结果

图 1 和图 2 中: 菱形表示实际工资小于或等于 5 万元的人群, 三角形表示实际工资大于 5 万元的人群, 而圆点则表示采用本文算法聚类后被划分到错误簇的点.

由图 1 和图 2 可以看出, 无论年龄和工作时间过大或过小, 其工资基本都在 5 万元以下, 这与实际情况相符. 35 ～ 60 岁的人群, 事业比较稳定, 工作时间不会太短, 同时基本都有自己的家庭, 不会为了增加收入而工作过长时间, 所以每周工作时间基本都在 40 h 上下波动. 年龄过大而仍坚持工作的人, 由于身体原因不会工作太长时间, 但工资依然很高. 虽然年龄和工作时间都与工资有很大的关系, 但由图 1 和图 2 可以看出, 二者并不能完全决定工资的多少. 通过对采用基于聚类融合的混合属性增量聚类的聚类结果进行分析可知, 误差基本都产生在那些学历低、工作时间短、职位高的数据点, 导致产生这些误差的原因有可能是个人工作经验或家族企业等. 与采用文献[2]和[14]中的算法对 Adult 数据集进行聚类的结果

相比, 本文提出的基于聚类融合的混合属性数据增量聚类算法精确度更高, 且算法构造简单、易于计算和实现.

5 结 论

鉴于传统增量聚类算法存在的不足, 本文将聚类融合方法引入混合属性数据增量聚类中, 提出了基于聚类融合的混合属性数据增量聚类算法, 以避免由单一聚类带来的不稳定和随机性的缺陷. 该算法首先采用聚类融合方法对数值型数据和分类型数据产生的聚类成员进行融合; 然后借助簇的合并和分裂方法调整簇的数目, 从而获得聚类的当前结果. 当有新增数据到达时, 采用矢量量化的方法和分类属性实际值将新增数据追加到原有聚类成员中, 并对含有新增数据点的簇进行融合, 直到不再有新的数据点到达为止. 本文利用已有数据的特征, 设计了矢量量化阈值的方法, 从而提高了增量聚类的精度. 最后通过对Diagnosis数据集和Adult数据集进行实例分析, 表明了改进的增量聚类算法能够改善不稳定性和随机性, 具有很好的聚类效果.

参考文献(References)

[1] Jing L, Ng M K, Huang J Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data[J]. IEEE Trans on Knowledge and Data Engineering, 2007, 19(8): 1026-1041.

[2] Hsu C C, Huang Y. Incremental clustering of mixed data based on distance hierarchy[J]. Expert Systems with Applications, 2008, 35(3): 1177-1185.

[3] Hartigan J A. Clustering algorithms[M]. New York: John Wiley & Sons, Inc, 1975.

[4] Carpenter G, Grossberg S. Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures[J]. Neural Networks, 1990, 3(2): 129-152.

[5] Ester M, Kriegel H P, Sander J, et al. Incremental clustering for mining in a data warehousing environment[C]. Proc of the 24th Int Conf on Very Large Data Bases. NY: Morgan Kaufmann, 1998: 323-333.

[6] 陈宁, 陈安, 周龙骧. 基于密度的增量式网格聚类算法[J]. 软件学报, 2002, 13(1): 1-7.
(Chen N, Chen A, Zhou L X. An incremental grid density-based clustering algorithm[J]. J of Software, 2002, 13(1): 1-7.)

[7] 黄永平, 邹力鹃. 数据仓库中基于密度的批量增量聚类算法[J]. 计算机工程与应用, 2004, 40(29): 206-208.
(Huang Y P, Zou L K. An incremental density-based clustering algorithm in a batch mode used in a data warehouse[J]. Computer Engineering and Applications, 2004, 40(29): 206-208.)

[8] 刘建晔, 李芳. 一种基于密度的高性能增量聚类算法[J]. 计算机工程, 2006, 32(21): 76-78.
(Liu J Y, Li F. An efficient incremental algorithm for clustering based on density[J]. Computer Engineering, 2006, 32(21): 76-78.)

[9] Sebastian Lühr, Mihai Lazarescu. Incremental clustering of dynamic data streams using connectivity based representative points[J]. Data & Knowledge Engineering, 2009, 68(11): 1-27.

[10] Nikolai Alex, Alexander Hasenfuss, Barbara Hammerb. Patch clustering for massive datasets[J], Neurocomputing, 2009, 72(7/8/9): 1455-1469.

[11] 杨善林, 李永森, 胡笑旋, 等. K-means算法中的k值优化问题研究[J]. 系统工程理论与实践, 2006, 26(2): 97-101.
(Yang S L, Li Y S, Hu X X, et al. Optimization study on k value of K-means algorithm[J]. Systems Engineering-Theory & Practice, 2006, 26(2): 97-101.)

[12] Chan Y, Ching W, Ng M K, et al. An optimization algorithm for clustering using weighted dissimilarity measures[J]. Pattern Recognition, 2004, 37(5): 943-952.

[13] Lughofer E. Extensions of vector quantization for incremental clustering[J]. Pattern Recognition, 2008, 41(3): 995-1011.

[14] 文益民, 杨旸, 吕宝粮. 集成学习算法在增量学习中的应用研究[J]. 计算机研究与发展, 2005, 42(增): 222-227.
(Wen Y M, Yang Y, Lv B L. Research of the application ensemble learning algorithms to incremental learning[J]. J of Computer Research and Development, 2005, 42(S): 222-227.)