一种基于相似维的高维子空间聚类算法

陈 铭, 吉根林

(南京师范大学计算机科学与技术学院, 江苏 南京 210097)
(江苏省信息安全保密技术工程研究中心, 江苏 南京 210097)

[摘要] 提出了一种基于相似维的子空间聚类算法 SDSCA (Similar Dimension based Subspace Clustering Algorithm). 算法首先通过 Gini值来删除原高维数据空间中的冗余属性, 然后运用相似维概念来寻找彼此相似的属性, 最后在这些相似维所形成的子空间上运用传统聚类算法来进行聚类. 实验结果表明算法是正确的, 并且能够有效地避免冗余属性的干扰.

[关键词] 子空间聚类, 相似维, Gini值
[中图分类号] TP311 [文献标识码] A [文章编号] 1001-4616(2010)04-0119-04

A Subspace Clustering Algorithm for High Dimensional Data Based on Similar Dimension

Chen Ming Ji Genlin

(School of Computer Science and Technology, Nanjing Normal University Nanjing 210097, China)
(Jiangsu Research Center of Information Security & Privacy Technology, Nanjing 210097, China)

Abstract This paper proposes a subspace clustering algorithm—— SDSCA based on similar dimension Firstly the Gini value is used to remove the redundant attributes in the data space After removing the redundant attributes, the similar dimension is used to find the attributes that are close to each other Finally, the traditional clustering algorithms are used on these subspaces that formed by similar dimension The experiment results show that algorithm SDSCA is effective and also reduces the redundant attributes effectively.

Key words space clustering similar dimension, Gini value

聚类是一种用来发现数据分布模式的重要技术, 高维数据聚类是当前聚类领域的研究热点之一. 典型的高维数据包括零售交易数据、文档数据、空间数据、地理数据、多媒体数据、网络访问数据、时间序列数据、基因数据等. 传统聚类方法在处理这些高维数据时会由于高维数据所具有的稀疏性以及处理过程中的“维度效应”现象而失效, 因此有必要针对高维数据研究相应的聚类算法. 现有的高维数据聚类方法主要分为 3种[1]: (1)属性约简方法: 通过特征变换或者特征选择来将高维数据投影到比原数据空间低的特征空间中, 然后对约简后的数据采用传统的聚类方法进行聚类. (2)子空间聚类方法: 这类方法运用特定的策略在原数据空间中寻找部分子空间, 在这些子空间上来进行数据聚类. 如 Clique[2]、Enclus[3]、Doc[4]、Proclus[5]、Orclus[6]等. (3)其他方法: 这类算法通过对传统聚类算法中的相似度度量方式进行改进, 以设计适合度量高维数据的相似度度量准则来对高维数据进行聚类. 此外 FP-tree, SOM 等技术也被应用于高维数据聚类中, 如 Fpsub[7], Vpsom[8].

本文提出了一种基于相似维的子空间聚类算法 SDSCA. 算法首先通过 Gini值来删除原高维数据空间中的冗余属性, 然后运用相似维概念来寻找彼此相似的属性维, 最后在这些相似维所形成的子空间上运用传统聚类算法来进行聚类. 实验结果验证了算法的有效性, 并且能够有效地避免冗余属性的干扰.

收稿日期: 2010-06-10
基金项目: 国家自然科学基金(40871176).
通讯联系人: 吉根林, 博士, 教授, 博士生导师, 研究方向: 数据挖掘技术及其应用. E-mail: glj@njnu.edu.cn

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

1　相关概念

设 $A = \{A_1, A_2, \cdots, A_d\}$ 表示属性集合, DB表示一个 d 维数据对象集合, $|DB| = n$, $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ 表示 DB中的一个数据对象, x_{ij} 表示 X_i 在第 j 个属性上的取值. 设 S_k 为 k 维子空间, $S_k \subset A$, 其中 $k \leqslant d$.

定义 **1**　基本聚类

在 DB的每一属性上, 根据以下策略形成的类称为基本聚类.

(1) 连续型数据, 如果满足 $|x_{ij} - x_{kj}| < \delta$ ($i \neq k$ 且 $i, k \leqslant n$, $j \leqslant d$), 则数据对象 X_i 和 X_k 在第 j 维上相似, X_i 和 X_k 聚成一类.

(2) 类别型数据, 如果满足 $x_{ij} = x_{kj}$ ($i \neq k$ 且 $i, k \leqslant n$, $j \leqslant d$), 则数据对象 X_i 和 X_k 在第 j 维上相似, X_i 和 X_k 聚成一类.

对于每维数据, 根据其类型运用相应策略进行聚类所形成的各个簇称为基本聚类, 用 C_{ik} ($i \leqslant d$, $k = 1, 2, 3, \cdots, m$) 来表示在第 i 维上形成的第 k 个基本聚类, m 是在第 i 维上形成的簇的个数, 并用 $N(C_{ik})$ 表示 C_{ik} 中数据对象个数.

定义 **2**　Gini值

Gini值是度量数据划分或训练元组集 D 的纯度[1], 定义如下:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

其中, p_i 是 D 中元组属于 C_i 类的概率, 并用 $|C_{i,D}| / |D|$ 估计, $0 \leqslant Gini < 1$.

由于 Gini值反映的是数据集的纯度, 如果某一维的 Gini值越小, 则在该维下数据对象的纯度越高, 那么该维下数据对象之间的差异性越小, 这样的属性对最终的子空间形成是无关紧要的, 因此, 给定阈值 α ($0 \leqslant \alpha < 1$) 将 Gini值小于阈值的属性作为冗余属性删去.

定义 **3**　相似维

在数据集 DB中, 给定属性 i 和 j, 在 i 和 j 上分别形成 m 和 n 个基本聚类. 给定相似维阈值 β ($0 \leqslant \beta \leqslant 1$), 如果 $m = n$, 并且 $N\left(\left(C_{ik} \cap C_{jk}\right)\right) / \max(N(C_{ik}), N(C_{jk})) > \beta$ ($k = 1, 2, \cdots, m$), 则称属性 i 和 j 是相似的.

例: 如表 1所示, 数据集有 8个数据对象, 每个数据对象有 3个属性. 在属性 A 上形成 3个基本聚类 $C1 = \{1, 2, 3\}$, $C2 = \{4, 5\}$, $C3 = \{6, 7, 8\}$, 在属性 B 上形成 3个基本聚类 $C4 = \{1, 2\}$, $C5 = \{3, 4, 5\}$, $C6 = \{6, 7, 8\}$, 在属性 C 上形成 3个基本聚类 $C7 = \{1\}$, $C8 = \{2, 3, 4, 5, 6\}$, $C9 = \{7, 8\}$, 每个属性的基本聚类数相等. 若给定相似维阈值 $\beta = 0.6$, 发现 $N\left(C_{Ai} \cap C_{Bi}\right) / \max(N(C_{Ai}), N(C_{Bi})) > \beta$ ($i = 1, 2, 3$), 则属性 A 和属性 B 称作相似维, 而属性 C 和 A、B 之间都不满足相似维条件, 因此不是相似维.

表 **1**　示例数据集 **1**

Table 1　Sample data set 1

ID	属性 A	属性 B	属性 C
1	$C1$	$C4$	$C7$
2	$C1$	$C4$	$C8$
3	$C1$	$C5$	$C8$
4	$C2$	$C5$	$C8$
5	$C2$	$C5$	$C8$
6	$C3$	$C6$	$C8$
7	$C3$	$C6$	$C9$
8	$C3$	$C6$	$C9$

2　算法思想

给定高维数据集 DB, 算法 SDSCA 首先在数据集 DB的每维形成基本聚类, 然后根据基本聚类求得每维的 Gini值, 将 Gini值小于给定阈值的属性从原属性空间中删除, 接着在新的数据集中寻找满足相似维条件的聚类子空间, 最后在找到的子空间上运用传统聚类算法 (本文中, 对连续型数据选用算法 DBSCAN[9], 对类别型数据选用算法 CLOPE[10], 也可以选用其他传统聚类算法) 进行聚类得到结果.

2.1　算法描述

输入: 高维数据源, α, β

输出: 子空间聚类结果

方法:

(1) 在每一维上形成基本聚类, 计算每一维的 Gini值;

(2) 将 Gini值小于阈值 α 的属性从原属性空间中删除, 形成新的属性空间;

(3) 在新的数据集属性空间中, 根据阈值 β 找出满足相似维条件的属性, 利用这些属性形成聚类所需

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

要的子空间, 子空间中的属性两两相似;

(4) 对子空间数据采用传统聚类算法得到最终聚类结果.

2.2 算法举例

如表 2所示, 数据集共有 10个数据对象, 每个数据对象都有 5个属性. 按照算法首先在每维上形成基本聚类: 在 $A1$属性上形成 3个基本聚类 $C1$, $C2$, $C3$; 在 $A2$属性上形成 3个基本聚类 $C4$, $C5$, $C6$; 在 $A3$属性上形成 4个基本聚类 $C7$, $C8$, $C9$, $C10$; 在 $A4$属性上形成 4个基本聚类 $C11$, $C12$, $C13$, $C14$; 在 $A5$属性上形成 2个基本聚类 $C15$, $C16$. 然后求得每维的 Gini值, 取 $\alpha = 0.4$, $\beta = 0.6$, 发现属性 $A5$是冗余属性, 将其从原属性空间中删除. 再依据相似维阈值 β得到 2个聚类子空间 $\{A1, A2\}$和 $\{A3, A4\}$. 最后在这 2个子空间上聚类得到最终聚类结果. 如表 3所示.

表 2　示例数据集 2

Table 2　Sample data set 2

ID	$A1$	$A2$	$A3$	$A4$	$A5$
1	$C1$	$C4$	$C7$	$C11$	$C15$
2	$C1$	$C4$	$C7$	$C11$	$C15$
3	$C1$	$C4$	$C8$	$C12$	$C15$
4	$C1$	$C5$	$C8$	$C12$	$C15$
5	$C2$	$C5$	$C9$	$C12$	$C15$
6	$C2$	$C5$	$C9$	$C13$	$C15$
7	$C2$	$C5$	$C9$	$C13$	$C15$
8	$C3$	$C6$	$C10$	$C13$	$C15$
9	$C3$	$C6$	$C10$	$C14$	$C16$
10	$C3$	$C6$	$C10$	$C14$	$C16$

表 3　示例数据集 2的聚类结果

Table 3　Clustering results of sample data set 2

簇	$\{A1, A2\}$子空间	簇	$\{A3, A4\}$子空间
$C1$	1, 2, 3, 4	$C1$	1, 2
$C2$	5, 6, 7	$C2$	3, 4, 5
$C3$	8, 9, 10	$C3$	6, 7, 8
		$C4$	9, 10

3　实验结果分析

3.1　实验数据及环境

文献[11]中指出, 当一个数据集 DB的维数大于等于 16时, 在数据集上建立的索引将失效, 传统聚类算法效率也将大大降低, 因此可以将维数大于等于 16的数据视为高维数据. 为了验证算法的有效性, 本文分别在 2个 UCI数据集上进行了实验: 第一个为 Soybean, 该数据集含有 47个数据对象, 每个数据对象含有 35个属性; 第二个为 Pendigits, 该数据集含有 3498个数据对象, 每个数据对象含有 16个属性. 实验环境: Windows xp系统, 2.0G主频, 2.0G内存, VC2005.

3.2　实验结果分析

(1) Soybean的数据量很小, 时间消耗可以忽略不计, 因此不给出该数据集的时间效率图, 只给出它的聚类子空间及聚类结果. 为简单起见将 Soybean数据集中第一个属性定为 $A1$, 第二个属性定为 $A2$, 以此类推. 算法发现数据集中含有 16个冗余属性, 在删除这些冗余属性之后寻找满足相似维条件的聚类子空间, 得到 2个维数为 3的子空间 $\{A2, A8, A9\}$、$\{A3, A5, A12\}$和一个维数为 6的子空间 $\{A23, A24, A26, A27, A28, A35\}$. 在这 3个子空间上进行聚类得到如下聚类结果 (括号中为该簇中包含的数据对象, 不被任何簇包含的数据对象作为离群点处理):

聚类子空间 $\{A2, A8, A9\}$:

簇 $C1\{1, 4, 7, 10, 11, 13, 15, 17, 18\}$;

簇 $C2\{2, 6, 9, 12, 14, 16, 19, 20\}$;

簇 $C3\{22, 27, 34, 38, 42, 45\}$;

簇 $C4\{23, 28, 32, 37, 41, 47\}$;

簇 $C5\{30, 31, 33, 36, 39, 40, 43, 46\}$.

聚类子空间 $\{A3, A5, A12\}$:

簇 $C1\{1, 26, 31, 34, 41, 42, 44\}$;

簇 $C2\{2, 3, 4, 5, 6, 7, 8, 9, 10, 33, 35, 36, 37, 38, 39, 43, 47\}$;

簇 $C3\{11, 14, 15, 16, 18, 20\}$;

簇 $C4\{21, 22, 23, 24, 25, 27, 28, 30\}$.

聚类子空间 $\{A23, A24, A26, A27, A28, A35\}$:

簇 $C1\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$;

簇 $C2\{11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$;

簇 $C3\{21, 22, 24, 24, 25, 26, 27, 29, 30\}$;

簇 $C4\{31, 34, 41, 42, 44, 46\}$;

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

簇 *C*5{32, 33, 35, 36, 37, 38, 39, 40, 43, 45, 47}.

(2) Pendigits数据集含有 3 498个数据对象,每个数据对象含有 16个属性. CLIQUE是一个经典的子空间聚类算法,本文算法选择与该算法进行比较. 图 1和图 2为本文算法 SDSCA在该数据集上与算法 CLIQUE比较的运行时间效率图:

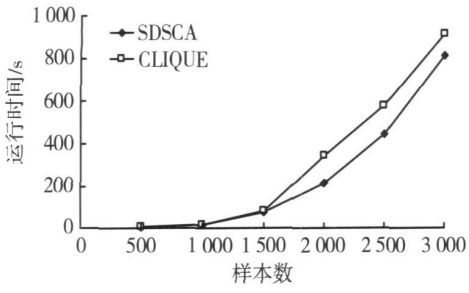

图 1　样本数量与运行时间关系

Fig.1　The relation between sample size and running time

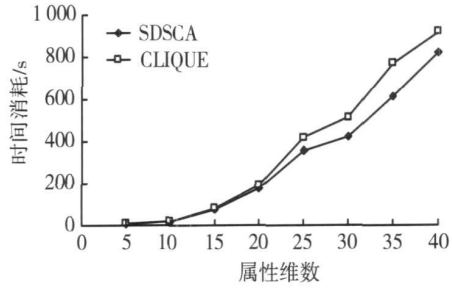

图 2　属性数量与运行时间的关系

Fig.2　The relation between attribute numbers and running time

可以发现,随着样本数的增加和属性数的增加,本文算法的时间消耗少于 CLIQUE 算法的时间消耗. 这是因为算法 SDSCA引入 Gini值对数据集中的冗余属性处理后,数据集的维数得到降低,因此性能优于 CLIQUE算法.

4　结论

通过引入 Gini值来删除原数据空间中的冗余属性,并且通过相似维概念找到聚类子空间,并在找到的子空间上运用传统聚类算法进行聚类. 实验证明了本文算法的有效性,当一个高维数据集中的冗余属性较多时,本文算法能够体现出较大优势.

[参考文献]

[1] Micheline JH. 数据挖掘:概念与技术[M]. 北京:机械工业出版社, 2001.

[2] Agrawal R, Gehrke J, Gunopulos D. Automatic subspace clustering of high dimensional data for data mining applications [C] //Proceedings of the 1998 ACM-SIGMOD International Conference on Management of Data Seattle. Washington: ACM Press, 1998, 6: 94-105.

[3] Chen C H, Fu A W C, Zhang Y. Entropy-based subspace clustering for mining numerical data[C] //Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego: ACM Press, 1999: 84-93.

[4] Procopiuc CM, Jones M, Agarwal P K, et al. A Monte Carlo algorithm for fast projective clustering[C] //Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. Madison: ACM Press, 2002: 418-427.

[5] Aggarwal C C, Procopiuc C, Wolf J L, et al. Fast algorithms for projected clustering[C] //Proceeding of the 1999 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1999: 61-72.

[6] Aggarwal C C, Yu P S. Finding generalized projected clusters in high dimensional spaces[C] //Proceedings of the 2000 ACM SIGMOD International Conference on Management of data. Dallas: ACM Press, 2000: 70-81.

[7] 单世民,王新艳,张宪超. 高维分类属性的子空间聚类算法[J]. 小型微型计算机系统, 2009(10): 2016-2 021.

[8] 刘铭,王晓龙,刘远超. 一种大规模高维数据快速聚类算法[J]. 自动化学报, 2009, 35(7): 859-866.

[9] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial database with noise [C] //KDD-96: Proceedings of the 2nd International Conference on Knowledge Discovering and Data Mining. Piscataway: IEEE Press, 1996: 226-231.

[10] Yang Yiling, Guan Xudong, You Jinyuan. CLOPE: a fast and effective clustering algorithm for transactional data[C] //Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Alberta: ACM Press, 2002: 682-687.

[11] 陈建斌. 高维聚类知识发现关键技术研究和应用[M]. 北京:电子工业出版社, 2008.

[责任编辑:顾晓天]

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net