第 36 卷 第 9 期 计 算 机 工 程 2010 年 5 月
Vol.36 No.9 Computer Engineering May 2010
·软件技术与数据库· 文章编号：1000—3428(2010)09—0065—03 文献标识码：A 中图分类号：TP301.6

基于子空间维度加权的密度聚类算法

黄王非[1]，陈黎飞[2]，姜青山[1,3]
(1. 厦门大学软件学院，厦门 361005；2. 福建师范大学数学与计算机科学学院，福州 360108；3. 成都大学，成都 610106)

摘 要：在高维数据聚类中，受维度效应的影响，现有的算法聚类效果不佳。为此，提出一种适用于高维数据的密度聚类算法 StaDeCon。在经典的 PreDeCon 算法基础上，引入子空间维度权重的计算方法，避免 PreDeCon 算法使用全空间距离度量带来的问题，提高了聚类的质量。在合成数据和实际应用数据集上的实验结果表明，该算法在高维数据聚类上可取得较好的聚类精度，算法是有效可行的。

关键词：聚类；高维数据；子空间；维度加权

Density Clustering Algorithm Based on Subspace Dimensional Weighting

HUANG Wang-fei[1], CHEN Li-fei[2], JIANG Qing-shan[1,3]
(1. School of Software, Xiamen University, Xiamen 361005; 2. School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 360108; 3. Chengdu University, Chengdu 610106)

【Abstract】In clustering of high dimensional data, most of the existing algorithms can not reach people's expectation due to the curse of dimensionality. Based on the classic PreDeCon algorithm, this paper presents the StaDeCon, a density clustering algorithm for high dimensional data, which introduces a measure of subspace dimensional weighting to avoid the problem existing in PreDeCon caused by using full dimensional distance, and in this way, the quality of clustering is improved. Experimental results both on artificial and practical data show that the algorithm is more accurate, and it is effective and feasible.

【Key words】clustering; high dimensional data; subspace; dimensional weighting

1 概述

聚类是数据挖掘的主要任务之一[1]，其目的是寻找数据集的一种划分，使得簇内数据点间的相似度尽可能大，而属于不同簇类的数据点间相似度尽可能小。现已提出了多种聚类算法，然而，在高维数据空间中，这些常用的聚类算法的聚类结果常常不尽如人意。主要原因是受“维度效应”[1-2]的影响，高维空间中存在着大量不相关的属性，令衡量数据点相似度(相异度)的常用方法(如欧氏距离)的有效性大大降低。因此，在高维空间中容易产生“差距趋零”现象[1-2]，即使用这些常用方法时对象间的差异倾向于相等。

一种处理高维数据的方法是维度约简。将高维数据映射到较低维的空间中，以消除那些与簇类不相关的数据维度，从而可以使用传统算法在约简后的子空间内完成聚类任务。根据约简方式的不同，维度约简可分为全局维度约简(GDR)和局部特征约简(LDR)2 类方法[3]，前者包括经典的 PCA[1]等技术，而后者正是近几年提出的子空间聚类算法的基础。

子空间聚类算法可以挖掘存在于不同子空间的簇类，被认为是维度约简方法的一种拓展。它有 2 个主要的任务——在数据集中寻找存在簇的子空间和寻找存在于子空间中的簇。为克服维度效应的影响，文献[4]提出了一种基于子空间选择的密度聚类算法(PreDeCon)，使用加权欧几里德距离进行对象间的相似度度量，从而提高了聚类的精度。PreDeCon 算法根据对象的 ε-邻域为对象选取所在的子空间，而对象 ε-邻域的确定是在全空间进行的，这个过程易受噪声的影响，从而降低了算法的鲁棒性和有效性。本文针对 PreDeCon 算法的不足，对算法进行了改进，给出了一种新的子空间维度权重的计算方法。在合成数据和实际应用数据集上的实验结果表明，改进后的算法进一步提高了高维数据的聚类质量。

2 相关工作

2.1 基于密度的聚类算法

基于密度的聚类算法是一种很常见的聚类算法，包括 DBSCAN, DENCLUE, OPTICS 等[1]。这些算法的原理都是寻找特征空间中被低密度区域分隔开来的高密度区域，通常需要 2 个参数来定义密度的概念：邻域半径 ε 和密度阈值 μ，后者指定邻域内包含的最少的数据点数目。这 2 个参数确定了聚类的密度下限。

2.2 基于子空间选择的密度聚类算法(PreDeCon)

PreDeCon 算法在经典的 DBSCAN 算法的基础上，使用加权欧几里德距离进行对象间的相似度度量。算法首先计算对象各个维度的权重。

假设 D 是有 d 个维度的数据库，$A=\{A_1,A_2,\cdots,A_d\}$ 代表它的属性集。p, q 是 D 中的对象，用 $\pi_{A_i}(p)$ 表示将 p 映射到属性 A_i，$N_\varepsilon(p)$ 表示 p 的 ε-邻域。$N_\varepsilon(p)$ 沿属性 $A_i\in A$ 的变化 $VAR_{A_i}(N_\varepsilon(p))$ 的定义如下：

$$VAR_{A_i}(N_\varepsilon(p))=\frac{\sum_{q\in N_\varepsilon(p)}(dist(\pi_{A_i}(p),\pi_{A_i}(q)))^2}{|N_\varepsilon(p)|}$$

作者简介：黄王非(1985－)，男，硕士，主研方向：数据挖掘；陈黎飞，副教授；姜青山，教授

收稿日期：2009-12-10 **E-mail：**qjiang@xmu.edu.cn

根据 $VAR_{A_i}(N_\varepsilon(p))$ 的大小赋予相应的维度对应的权重，具体的计算方法定义如下：

$$\omega_i=\begin{cases}1 & \text{if } VAR_{A_i}(N_\varepsilon(p))>\delta\\ \kappa & \text{if } VAR_{A_i}(N_\varepsilon(p))\leqslant\delta\end{cases}$$

其中，$\kappa\in R$ 且 $\kappa>>1$。

计算出权重后就可以采用加权欧几里德距离进行相似度度量，具体的度量方法如下：

$$dist_{pref}(p,q)=\max\left\{dist_p(p,q),dist_q(q,p)\right\} \quad (1)$$

$$dist_p(p,q)=\sqrt{\sum_{i=1}^{d}\omega_i(\pi_{Ai}(p)-\pi_{A_i}(q))^2} \quad (2)$$

其中，ω_i 是点 p 的第 i 个维度的权重。

最后，PreDeCon 算法重新定义了核心对象、密度可达、直接密度可达、密度相连等相关概念。基于这些定义，使用了经典的 DBSCAN 算法的策略进行聚类。实验结果表明，与其他自底向上的高维数据聚类算法[1]相比，该算法有效提高了聚类质量和聚类效率。

为确定数据对象所在的子空间，PreDeCon 算法采用了一种“局部性假设”，即根据对象的 ε-邻域所包含的其他对象来计算该对象的子空间(每个维度的权重)。这里，ε-邻域的确定是在全空间进行的。然而，如前所述，在高维空间中，若基于欧氏距离这样的方法衡量数据点间的相似度(相异度)，将导致“差距趋零”现象，削弱了“最近邻”的意义[1-2]，进而影响了 ε-邻域的概念。本文将主要针对这个问题对其进行改进，提出一种基于子空间维度加权的密度聚类算法(subspace Standard deviation weighted Density Connected clustering, StaDeCon)。

3 基于子空间维度加权的密度聚类算法

StaDeCon 算法的流程如图 1 所示。

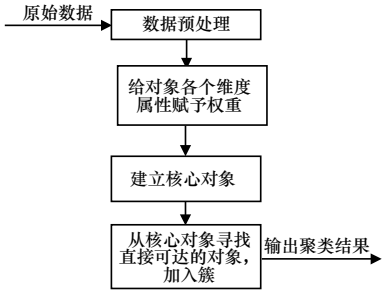

图 1　StaDeCon 算法流程

3.1 相关定义

用 $N_\varepsilon^{\pi_{Ai}}(p)$ 表示 p 在属性 A_i 上的 ε-邻域，可以根据 $N_\varepsilon^{\pi_{Ai}}(p)$ 中包含的点的数量情况来确定 p 在属性 A_i 上的权重。如果 $\left|N_\varepsilon^{\pi_{Ai}}(p)\right|$ 大于某个给定的阈值 $\eta\in N$，则认为 p 在属性 A_i 上的 ε-邻域是密集，赋予该属性维度较大的权重；反之，如果 $\left|N_\varepsilon^{\pi_{Ai}}(p)\right|$ 小于给定的阈值 $\eta\in N$，认为 p 在属性 A_i 上的 ε-邻域是稀疏的，将该属性维度的权重赋予一个较小的值。对于 $\left|N_\varepsilon^{\pi_{Ai}}(p)\right|$ 大于给定的阈值 η (即分布密集)的维度，可以根据 $N_\varepsilon^{\pi_{Ai}}(p)$ 上的标准偏差进一步细化权重的度量，标准差越小，数据的分布就越紧凑，相应地就赋予较大的权重，反之亦然[5]。

定义 1 维度权重

令 $p\in D$, $\eta\in N$, $\kappa\in(0,1)$，δ_i 是 $N_\varepsilon^{\pi_{Ai}}(p)$ 上的方差，点 p 的各维度权重定义如下：

$$\omega_i=\begin{cases}\exp\left(\dfrac{\max\limits_{j=1}^{d}\delta_j-\delta_i}{\max\limits_{j=1}^{d}\delta_j}\right) & \left|N_\varepsilon^{\pi_{Ai}}(p)\right|\geqslant\eta\\ \kappa & \left|N_\varepsilon^{\pi_{Ai}}(p)\right|<\eta\end{cases}$$

该值越大说明该维度对确定点 p 所在的簇越重要。由于高维空间中全空间距离的有效性大大降低[1]，距离能否有效地反映点之间的相似性对于聚类的结果有很大的影响，本文的算法沿用式(1)、式(2)进行对象之间的相似度度量。

定义 2 带权重的 ε-邻域

令 $\delta\in R$，点 p 的带权重的 ε-邻域记作 $N_\varepsilon^w(p)$，定义：

$$N_\varepsilon^w(p)=\left\{x\in D\mid dist_{pref}(p,x)\leqslant\varepsilon_{pref}\right\}$$

定义 3 带权重的核心对象

令 ε_{pref}, $\delta\in R$, $\mu\in N$，点 p 是核心对象当且仅当 $\left|N_\varepsilon^w(p)\right|\geqslant\mu$，用符号 $CORE_{den}^{pref}(p)$ 表示。

定义 4 带权重的直接可达

令 ε_{pref}, $\delta\in R$, $\mu\in N$，点 o 从点 p 直接可达当且仅当满足：(1) $CORE_{den}^{pref}(p)$；(2) $o\in N_\varepsilon^w(p)$；用符号 $DIRREACH_{den}^{pref}(p,o)$ 表示。

3.2 算法过程描述

算法步骤如下：

Step1 给对象各个维度赋予权重

(1)对于对象 p 的各个维度，寻找其 $N_\varepsilon^{\pi_{Ai}}(p)$ 集合并存储。

(2)如果 $\left|N_\varepsilon^{\pi_{Ai}}(p)\right|$ 小于给定的阈值 η，则赋予较小的权重；反之，则计算集合 $N_\varepsilon^{\pi_{Ai}}(p)$ 中各对象的标准差。

(3)重复步骤(1)、步骤(2)，直到所有对象的各个维度都计算完毕。

(4)根据定义 1 计算 $\left|N_\varepsilon^{\pi_{Ai}}(p)\right|$ 不小于给定阈值 η 的维度对应的权重。

Step2 建立核心对象

(1)对各个对象 p 寻找它的 $N_\varepsilon^w(p)$ 集合并存储。

(2)对于给定的阈值 μ，将满足 $\left|N_\varepsilon^w(p)\right|\geqslant\mu$ 条件的对象标记为核心对象。

Step3 从核心对象出发寻找直接可达的对象加入到当前簇中

(1)从对象 p 开始，如果 p 是核心对象，则转步骤(2)；否则，p 将标记为噪声。

(2)产生一个新的簇 ID，并将 $N_\varepsilon^w(p)$ 集合中的对象加入队列 Q。

(3)从 Q 队列中取出第 1 个对象赋予 q，对 $x\in N_\varepsilon^w(q)$，如果 x 是未分类的对象，将 x 加入队列 Q 且赋予当前簇 ID。如果 x 是噪声，则赋予前簇 ID，从 Q 中将 q 移除。

(4)重复执行步骤(3)，直至队列 Q 为空。

(5)如果还有未分类的对象，则从步骤(1)开始执行；否则，结束聚类，输出结果。

4 实验结果与分析

4.1 实验数据

本文在合成数据和真实的高维数据集上测试 StaDeCon 算法的有效性。实验对比的算法为 PreDeCon 算法[4]，以验证本文提出的维度加权方法的有效性。

使用文献[6]提供的方法生成了 3 个合成数据集 DG-1,

DG-2, DG-3，均包含噪声点，用于检验算法抗噪声干扰的性能。数据集的参数如表 1 所示，它们具有不同的维度数、数据点数、簇数目，簇的平均相关维度数据也有差异。每个数据集都包含有一定比例的噪声点。

表 1　合成数据集参数

数据集	数据点数目	维度	簇数目	平均相关维度数	噪声点比例/(%)
DG-1	1 000	20	2	14	5
DG-2	1 000	40	4	20	5
DG-3	2 000	60	6	39	5

本文还使用了经典的测试数据集 KDD CUP 99，该数据集来源于 Internet 中的真实数据，每条数据记录的特征达 41 维(包括若干类属型属性)，是从一个模拟美国空军所属局域网络导出的 9 周内的 TCP/IP 通信数据，该数据集是目前测试网络入侵技术公认的 benchmark 数据，不含有噪声点。鉴于 KDD CUP 99 原始数据集规模较大，在此随机抽取 5 000 条记录(包括 2 000 条正常数据以及 3 000 条异常数据)进行实验。

4.2　实验设置

实验环境为 Pentium 4 CPU 3.00 GHz，1.00 GB 内存，Windows XP 系统，ECLIPSE 3.1。算法用 Java 语言实现。在实验过程中，实验数据均采用最大最小规范化方法规范到 0~1 之间。StaDeCon 算法需要输入的参数：维度属性上邻域半径 ε，维度属性邻域内包含对象数目阈值 η，带权重的邻域半径 ε_{pref}，成为核心对象至少要包含的对象数 μ。参数 ε_{pref} 和 μ 指定了簇必须满足的密度阈值，根据 PreDeCon 给出的方法设置这 2 个参数的取值。

ε, η 这 2 个参数与维度权重的计算有关，直接影响聚类结果的好坏。只有当维度属性上的邻域半径内包含的对象数超过给定的阈值，才认为对象在该维度上的投影被是密集的。显然，这 2 个参数的取值和数据集中对象的个数有关。数据集越大，点在各维度上投影分布的密集度也可能较大，如果 ε 的值一样，越大的数据集 η 的值相应也要设得越大。实验表明，当数据集的大小在 1 000~10 000 之间，ε 设为 0.005，η 的取值区间为[30, 42]，可以获得较好的聚类效果。

4.3　聚类结果及分析

本文采用 $F1$[7]指标评估方法对聚类结果进行评估，$F1$ 指标的定义如下：

$$F1_{ij}=\frac{2\times Precision(i,j)\times Recall(i,j)}{Precision(i,j)+Recall(i,j)}$$

其中，$Precision(i,j)$ 和 $Recall(i,j)$ 分别表示簇的查全率和查准率，由下列 2 式计算得出：

$$Precision(i,j)=\left|c_i\cap l_j\right|\Big/\left|c_i\right|,\quad Recall(i,j)=\left|c_i\cap l_j\right|\Big/\left|l_j\right|$$

其中，类别 $L=\{l_1,l_2,\cdots,l_k\}$ 表示集合中数据点所属的实际类别；簇 $C=\{c_1,c_2,\cdots,c_k\}$ 表示聚类分析得到的类簇结果。

最终，整个聚类结果的 $F1$ 值即为所有簇的加权平均：

$$F1=\frac{\sum_{i=1}^{m}|c_i|\times\max(F_{ij})}{|D|}$$

聚类结果如表 2 所示。由表 2 可以看出在 3 个合成的数据集中 PreDeCon 算法和本文提出的 StaDeCon 算法都能很好地分开各个簇。但是 PreDeCon 算法仍有少部分对象归错类，而 StaDeCon 算法则进一步提高了聚类的精度。在 KDDCUP99 数据集上，PreDeCon 算法的聚类精度只有 78.86%，聚类的质量比较一般，而 StaDeCon 算法对比 PreDeCon 算法在准确率上都有 12%左右的提升。

表 2　算法聚类结果的 *F1* 指标对比　(%)

数据集	PreDeCon	StaDeCon
DG-1	98.25	99.90
DG-2	97.13	99.90
DG-3	96.33	99.90
KDDCUP99	78.86	90.13

图 2 分析了 StaDeCon 算法的聚类精度与算法参数 η 之间的关系。数据集是 KDDCUP99，邻域半径 ε 为 0.005，维度属性邻域内包含对象数目阈值 η 在区间[30,42]内变动。从图中可以看出，StaDeCon 算法的聚类精度对参数的变化不是太敏感，这说明 StaDeCon 算法具有较好的鲁棒性。

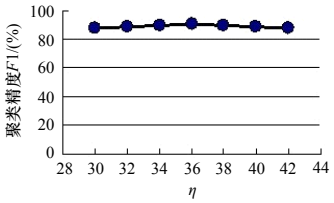

图 2　StaDeCon 算法聚类精度与参数的关系

实验结果表明，本文提出的基于子空间维度加权的密度聚类算法是有效的，它进一步提高了聚类的精度。

5　结束语

针对高维数据聚类中存在的维度效应，全空间的距离变得不再有意义，导致现有一些算法聚类结果不尽如人意。本文在分析 PreDeCon 算法不足的基础上，提出一种基于子空间维度加权的密度聚类算法。StaDeCon 算法不使用全空间的 ε-邻域进行子空间的选择，而是基于各个维度属性上的 ε-邻域确定对象所在的子空间，并利用标准差进一步细化了维度权重的度量。实验结果表明，该算法在高维数据聚类中取得了较好的效果。对算法中各参数对聚类精度的影响、能否动态地改变维度权重及如何更有效地定义相似性度量是下一步的研究内容。

参考文献

[1] Han Jiawei, Kamber M. 数据挖掘概念与技术[M]. 范　明, 孟小峰, 译. 2 版. 北京: 机械工业出版社, 2008.

[2] Hinneburg A, Aggarwal C C, Keim D A. What Is the Nearest Neighbor in High Dimensional Spaces[C]//Proc. of the 26th International Conference on Very Large Databases. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000.

[3] Dash M, Liu Huan. Dimensionality Reduction[C]//Proc. of International Conference on Knowledge Discovery and Data Mining. [S. l.]: John Wiley & Sons, Inc., 2003: 685-690.

[4] Bohm C, Kailing K, Kriegel H P, et al. Density Connected Clustering with Local Subspace Preferences[C]//Proc. of the 4th IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Computer Society, 2004: 27-34.

[5] 陈黎飞, 姜青山, 王声瑞. 基于层次划分的最佳聚类数确定方法[J]. 软件学报, 2008, 19(1): 62-72.

[6] Aggarwal C C, Procopiuc C, Wolf J L, et al. Fast Algorithm for Projected Clustering[J]. IEEE Trans. on Knowledge and Data Engineering, 1999, 28(2): 61-72.

[7] Tan Songbo, Cheng Xueqi, Ghanem M M, et al. A Novel Refinement Approach for Text Categorization[C]//Proc. of the 14th ACM International Conference on Information and Knowledge Management. New York, NY, USA: ACM Press, 2005: 469-476.

编辑　顾逸斐