一种实用高效的聚类算法*

王建会+, 申 展, 胡运发

(复旦大学 计算机与信息技术系,上海 200433)

An Applicable and Efficient Clustering Algorithm

WANG Jian-Hui+, SHEN Zhan, HU Yun-Fa

(Department of Computing and Information Technology, Fudan University, Shanghai 200433, China)

+ Corresponding author: Phn: +86-21-36011651, Fax: +86-21-56691583, E-mail: wangjh888@citiz.net, http://wangjh.myrice.com

Received 2003-05-27; Accepted 2003-09-09

Wang JH, Shen Z, Hu YF. An applicable and efficient clustering algorithm. *Journal of Software*, 2004,15(5): 697~705.

http://www.jos.org.cn/1000-9825/15/697.htm

Abstract: In the research on IR (information retrieval), lots of clustering algorithms have been developed, and in most of them some parameters should be determined by hand. However, it is very difficult to determine them manually without any prior domain knowledge. To solve this problem, an applicable and efficient clustering algorithm is presented. It aims at avoiding any parameter to be determined by hand, and at the same time, improving the efficiency of clustering and the property of IR. The new clustering algorithm is analyzed on several facets and applied later to cluster Chinese documents. The results of the application confirm that the new clustering algorithm is very applicable and efficient.

Key words: IR (information retrieval); clustering; subspace; pattern recognition

摘 要: 在信息处理研究领域,现有的大多数聚类算法都需要人为地给出一些参数.然而,在没有先验知识的情况下,人为地确定这些参数是十分困难的,而且现有的聚类算法的时空效率也有待于进一步提高.为了解决这一难题,首先根据样本分布特性,通过数学分析,得到确定样本空间划分间隔数的数学函数,然后,再根据样本分布特性,采用爬山的策略得到样本类的划分,最后提出了一种实用而高效的聚类算法.从多个角度分析了该算法的性能,并将该算法应用于中文文本聚类.理论分析和应用结果都表明,该算法不仅不需要人为确定参数,同时,还可以提高信息处理的时空效率和性能.

关键词: 信息处理;聚类;子空间;模式识别

中图法分类号: TP18 **文献标识码**: A

随着科学技术的高速发展以及各种资源数量的不断增多,为了提高效率,信息处理已经成为当前最重要的研究内容,其中涉及到信息抽取、自然语言理解、自动聚类和分类、自动摘要、自动标注和主题识别、信息结

* Supported by the National Natural Science Foundation of China under Grant No.60173027 (国家自然科学基金)

作者简介: **王建会**(1972－),男,江苏淮阴人,博士生,主要研究领域为人工智能,自然语言处理;**申展**(1979－),女,硕士生,主要研究领域为全文数据库;**胡运发**(1940－),男,教授,博士生导师,主要研究领域为数据工程,知识工程.

构分析以及文本生成.其中,关于自动聚类方面的研究较为深入,而且,聚类技术已成为信息处理的核心技术.从20 世纪 40 年代至今,国内外的研究者提出了很多聚类算法,如基于层次的算法(CHAMELEON[1],CURE[2],BIRCH[3])、基于平面分割的算法(k-means[4],FREM[5])、基于密度的算法(DENCLUE[6],OPTICS[7],DBSCAN[8])、基于规则和模型的算法[9],以及基于网格和子空间的算法(STING[10],WaveCluster[11],CLIQUE[12]),等等.然而,在这些众多的算法中,大多数算法都需要事先人为地给出一些参数,而且时空效率也有待于进一步提高.然而,在没有先验知识的情况下,人为地确定这些参数是十分困难的.

为了解决这个难题,需要研发新的聚类算法,在保证不降低时空效率和信息处理性能的前提下,力图减少或避免需要事先人为确定的参数.此项研究工作是目前模式识别与人工智能研究领域的前沿课题[13],既具有一定的实际意义,又具有较强的理论意义,并且有利于推动该研究领域的发展,丰富该学术领域的研究.

基于上述研究动机,我们受文献[12]研究工作的启发,采用子空间模型,提出了一种实用而且高效的聚类算法.该算法基于子空间分析,力图避免需要事先人为确定的参数,同时提高时空效率和信息处理的性能.尽管本文得到了文献[12]的启发,但是,本文提出的聚类算法除了与文献[12]中的 CLIQUE 算法有许多的不同点、对其进行了较大的改进以外,自身还有诸多国内外现有的研究工作中所没有的创新内容.首先,CLIQUE 算法采用自底向上的方式,先根据密度阈值,找到一些小的聚类,然后再在其他坐标维上,根据一定的条件,将这些小的聚类逐步合并成较大的聚类,而本文提出的算法采用自顶向下的方式,先将整个样本集作为一个类,在每一个坐标维上,只通过相邻单元之间的比较来寻找对每个已有类的一个划分,最终生成对输入样本集的非对称层次聚类.其次,CLIQUE[12],DENCLUE[6]等现有算法需要指定诸如密度阈值等参数,然而,在缺乏领域知识的情况下,人为指定参数较为困难,而本文提出的算法无须事先指定参数值.另外,CLIQUE[12],DENCLUE[6]等算法对所有的类别都使用一个相同的密度阈值,没有对不同密度的聚类区别对待,不容易识别出样本数量较少的类,而本文提出的算法则不存在这一问题.此外,CLIQUE 算法需要有判断两个聚类是否应该合并的运算,而本文在识别分类的同时,已经找到了聚类之间的划分,因而 CLIQUE 算法的复杂度大于本文算法的复杂度.

本文第 1 节首先给出相关术语的定义,然后给出关于样本分布空间的性质,接着,详细论述本文提出的实用、高效的聚类算法.第 2 节将本文提出的算法应用于中文文本聚类,以考核其性能.最后是结论以及今后的工作.

1 实用高效的聚类算法

1.1 样本分布的性质

定义 1(样本类分布的投影区). 设向量空间的维数为 d,某一个样本类 C 中的所有样本在 i 维上投影区域的下限和上限分别为 D_i^L 和 D_i^U,则 $\prod_{i=1}^{d}[D_i^L,D_i^U]$是向量空间中一个包含该类所有样本的、连续且最小的区域,即有以下条件成立:

① 对于任意的样本 $x=(x_1,x_2,\ldots,x_d)\in C$,有 $x\in\prod_{i=1}^{d}[D_i^L,D_i^U]$.

② 对于任意的样本 $x=(x_1,x_2,\ldots,x_d)\notin\prod_{i=1}^{d}[D_i^L,D_i^U]$,有 $x\notin C$.

③ 不存在任何超立方体区域 $\prod_{i=1}^{d}[D_{i2}^L,D_{i2}^U]\subset\prod_{i=1}^{d}[D_i^L,D_i^U]$能够包含该类的所有样本.

定义 $\prod_{i=1}^{d}[D_i^L,D_i^U]$为该样本类 C 的投影区,D_i^L 和 D_i^U 分别为投影区在向量维 i 上的下界和上界.

定理 1(样本类分布的单调性). 如果一个样本类的所有样本都分布在 $k(2\le k\le d,d$ 为向量空间维数)维向量子空间上的一个投影区内,那么它也应该分布在 $k-1$ 维向量子空间上的一个投影区内.

证明:根据定义 1,对于给定的样本类 C 中的任意样本 $x^{(k)}=(x_1,x_2,\ldots,x_k)\in C$,有 $x\in\prod_{i=1}^{k}[D_i^L,D_i^U]$,则 $D_i^L\le x_i\le D_i^U(1\le i\le k)$.所以,在 $k-1$ 维向量子空间上与其对应的投影 $x^{(k-1)}=(x_1,x_2,\ldots,x_{k-1})\in\prod_{i=1}^{k-1}[D_i^L,D_i^U]$.$x^{(k)}$的上标 k 表

示 $x^{(k)}$是 k 维向量子空间上的向量.

而对于任意的样本$x^{(k-1)}=(x_1,x_2,...,x_{k-1})\notin\prod_{i=1}^{k-1}[D_i^L,D_i^U]$,则至少有一个$j(1\le j\le k-1)$,使得$x_j\notin[D_j^L,D_j^U]$,因而其对应的$x^{(k)}=(x_1,x_2,\ldots,x_k)\notin[D_j^L,D_j^U]\cdot\prod_{i=1,i\ne j}^{k}[D_i^L,D_i^U]=\prod_{i=1}^{k}[D_i^L,D_i^U]$,$x^{(k)}\notin C$,所以 $x^{(k-1)}\notin C$.

此外,假设存在一个超立方体区域$\prod_{i=1}^{k-1}[D_{i2}^L,D_{i2}^U]\subset\prod_{i=1}^{k-1}[D_i^L,D_i^U]$能够包含该类中的所有样本,则在相应的 k 维向量子空间上有$[D_k^L,D_k^U]\cdot\prod_{i=1}^{k-1}[D_{i2}^L,D_{i2}^U]\subset[D_k^L,D_k^U]\cdot\prod_{i=1}^{k-1}[D_i^L,D_i^U]$,即存在一个超立方体区域$[D_k^L,D_k^U]\cdot\prod_{i=1}^{k-1}[D_{i2}^L,D_{i2}^U]\subset\prod_{i=1}^{k}[D_i^L,D_i^U]$能够包含该类中的所有样本,这与定义 1 相矛盾,因而$\prod_{i=1}^{k-1}[D_i^L,D_i^U]$是 k–1 维子向量空间中一个包含该类所有样本的、连续且最小的区域.

根据定义 1,定理 1 得证. □

推论 1(投影区数量的单调性). 一组样本在$k(2\le k\le d)$维向量子空间上的投影区数量,不少于其在 k–1 维向量子空间上的投影区数量.

尽管上述定义和定理与文献[12]中的部分内容相一致,但是,与文献[12]相比,本文给出了详细的形式化定义和证明.

1.2 基于子空间的聚类算法

1.2.1 算法的思路及与相关文献的比较

在信息处理研究中,经常需要对信息样本进行聚类,以便于进行信息抽取,如在主题标注、自动摘要研究中,通过对语句聚类抽取出重要内容,再抽取出主题和生成摘要.根据上述定理 1 和推论 1,本文提出基于子空间的聚类算法,从低维向高维逐维进行聚类.在第 i 维上进行聚类时,将 i –1 维子空间中每个已分好的类再分解成子类,随着由低维向高维逐步进行,形成了对原样本集的自顶向下的层次分类.

正如引言中所述,尽管文献[12]也采用子空间的方法,但是,本文提出的算法与文献[12]中的算法有诸多不同点,如与文献[12]中的算法相比,本文的算法无须人为指定参数,不忽视样本数量少的类,算法复杂度低,采用自顶向下的聚类方式等等.

1.2.2 算法说明

本文提出的基于子空间的聚类算法的策略是,在每一维上,将样本向坐标维上投影,得到间隔内样本数与间隔曲线,在这条曲线上寻找波峰和波谷,从而得到样本类的划分.对于第 i 个山峰,刚过去的山谷是它的第 1 个山谷,即将到达的山谷是它的另一边的山谷,也就是它的第 2 个山谷,也是第 i +1 个山峰的第 1 个山谷.一座山只有 1 个山峰和两个山谷.

首先在第 1 维上寻找分类:① 将第 1 维等分成小间隔,将所有样本投影到第 1 维上,得到每一个间隔内样本数量在第 1 维上的变化曲线;② 对该曲线进行平滑处理,以消除噪声干扰;③ 采用爬山的方法,从第 1 个间隔开始,相邻间隔内的样本数进行比较,如果第 j 个间隔内的样本数小于第 j+1 个间隔内的样本数,则是在爬山,如果第 j 个间隔内的样本数大于第 j+1 个间隔内的样本数,则是在下山,先记录第 1 个山谷,随着 j 的增大,当第 j 个间隔内的样本数分别大于第 j–1 个间隔和第 j+1 个间隔内的样本数时,则到达了山顶,记录下第 1 个山顶的位置 j;④ 开始下山,当第 j 个间隔内的样本数分别小于第 j–1 个间隔和第 j+1 个间隔内的样本数时,则到达了山谷,记录下该山谷的位置 j;⑤ 继续寻找新的山峰和山谷,直到最后一个间隔为止;⑥ 以一个山峰对应一个类的方式,将样本分成类.

在第 i 维上进行聚类时,假设在 i –1 维子空间中已经将所有的样本聚成 k 类,在第 i 维上,对这 k 类中的每一类中的所有样本都进行上述寻找山峰的过程,并将该类样本聚成子类,最终将形成对原样本的自顶向下、非对称的层次分类树.如果在第 i 维上进行聚类时,某个已有类的样本投影曲线只有一个山峰,则在第 i 维上对该类不作任何处理.由于本文提出的基于子空间的聚类算法采用爬山的方式来搜寻类别,因而,本文将其称作 CLIMB(clustering algorithm based on subspace).

1.2.3 进一步提高算法的性能

为了进一步提高性能,还需要考虑下面的几种情形.

首先,上述处理中,是针对第 1 个间隔中的样本数较少的情况,也就是说,处于山谷的情况.在实际应用中有可能相反,第 1 个间隔中的样本数较多,也就是说,处于山峰的情况,此时,处于下山的状态,而且已经找到第 1 个山峰.因为此时已经处于第 1 座山的山顶,而且只有经过山谷才有可能爬到山顶,所以已经走过了第 1 个山谷所在的位置,下一个找到的山谷将是第 2 个山谷.接着,找到下一个山谷的位置 m,将其作为第 2 个山谷的位置.为了保持算法的完整性,本文假设第 1 座山是对称的(因为在本文的聚类算法中,所关心的是样本类的划分,对应的是山的区分,因而第 1 座山的范围本身对结果没有影响,所以这一假设是成立的),也就是说,第 1 个和第 2 个山谷是第 1 座山两边的山谷,这样,可以取第 1 个山谷的位置为 $-m$.后续寻找过程与上述过程相同.

与此相似,当到达最后一个间隔时,有 3 种情况.假设最后一个已经确认的山峰高度(即最后一个山峰中所有间隔内样本数的最大值)为 *Peak*,最后一个已经确认的山谷位置在 n 处.第 1 种情况是,当到达最后一个间隔时,处于爬山状态,即已确认山峰的高度 *Peak* 是前一个山峰的高度,此时,取最后一个间隔内的样本数作为最后一个山峰的高度,并取最后一个山谷的位置为 $\xi+(\xi-n)$(ξ为间隔数).第 2 种情况是,当到达最后一个间隔时,处于下山状态,即已确认山峰的高度 *Peak* 是当前(即最后一个)山峰的高度,而且,最后一个间隔内的样本数大于 *Peak*/2,此时,取最后一个山谷的位置为 $\xi+(\xi-n)$.第 3 种情况是,当到达最后一个间隔时,处于下山状态,即已确认山峰的高度 *Peak* 是当前(即最后一个)山峰的高度,但是,最后一个间隔内的样本数小于 *Peak*/2,此时,取最后一个山谷的位置为 ξ.

第 3 种需要考虑的情形是,如何确定间隔数 ξ.取单个间隔内属于同类的平均样本数为 k,而且共有 n 个待聚类的样本,那么,随着 n 的增大,为了能够有效地区分不同类的样本,为了确保聚类过程收敛,k 必须是满足以下 3 个条件的非降正整数序列:

1. $\lim_{n\to\infty} k=\infty$;
2. $\lim_{n\to\infty} n^{-1}k=0$;
3. $\lim_{n\to\infty} (\log n)^{-1}k=\infty$.

函数关系 $k\propto\sqrt{n}$ 是满足上述条件的一个特解,所以可取 $k=\sqrt{n}$,从而得到等分间隔数 ξ:

$$\xi=n/k=\sqrt{n} \tag{1}$$

另外,研究中还发现,不同的应用需要抽取和保留的信息量是不同的.比如,对于自动摘要和主题标注,只需抽取较为重要的内容,将次要的内容丢掉;而对于聚类和分类来说,所有待聚类或分类的样本都需要保留.为此,引入一个控制最终信息量的调节阈 *Tuner_APP*,在第 i 维上进行聚类时,对于已完成的一个聚类,假设其样本投影在区间 $[D_i^L,D_i^U]$ 内,则仅保留投影在区间 $[D_i^L+Tuner\_APP,D_i^U-Tuner\_APP]$ 内的样本,而丢弃其他的样本.对于聚类和分类来说,取 *Tuner_APP*=0,保留所有的样本;而对于自动摘要和主题标注,则根据具体应用对结果内容详细程度的不同要求,相应地调节 *Tuner_APP*,如果要求取总信息量 δ%生成摘要,或用于进行主题标注,向量空间维数为 d,则取

$$Tuner\_APP=\left\lceil \left(D_i^U-D_i^L\right)\times(1-\delta\%/d)/2 \right\rceil \tag{2}$$

第 5 种需要考虑的情形是,现实世界中的样本在向量空间中可能存在以下 3 种分布情况:

1. 两类样本在任意一个坐标维上的投影区域不相交,或者至多有较小范围的相交;

2. 两类样本在某些坐标维上的投影区域不相交,或者至多有较小范围的相交,而在其他坐标维上相交范围较大;

3. 两类样本在所有坐标维上相交范围都较大.

在第 1 种情况下,可以直接将两类样本分开.对于第 2 种情况,当处理相交范围较大的坐标维时,也就是,当两个或多个山峰连在一起时,则暂时不将其分开,而是作为 1 个山脉,将该山脉中最高的一个山峰峰值作为该整个山脉的峰值,等到处理后续的、投影区域不相交,或者至多有较小范围相交的其他维时,再将其分开.处理的策略是,当一个山谷的深度 *Deep_Valley* 与其相邻的山峰高度 *Peak* 之间的关系是 *Deep_Valley*≤25% *Peak* 时,则将其

作为山脉,或者,将其作为干扰噪声的下波幅,因而又可以起到平滑作用,以控制曲线的光滑程度.此外,将山脉中最高的一个山峰峰值作为该整个山脉的峰值,可以避免山侧面的噪声上波幅干扰,对等间隔内样本数分布曲线进行进一步平滑处理.第 3 种分布情况只可能是一种假想的极端情况.因为在高维向量空间中,如果两类样本在所有坐标维上相交范围都较大,那么它们在各维上的特征几乎就是完全相同的,最终它们总的特征也就是相同的或相近的,而特征相同或相近的样本类应该是同一个类.从另一个角度而言,一个分布范围较广的样本类,其投影分布曲线可能出现局部抖动,可能被误认为是在所有坐标维上相交范围都较大的两类样本.因而在现实世界中,上述 3 种分布中的第 2 种情况才是最为常见的分布情况,第 1 种情况比较少见,而第 3 种情况是高维向量空间中不可能存在的、假想的一种极端情况.在本文的算法中,只需考虑前两种情况.

为了进一步降低噪声干扰,除了上述平滑技术以外,本文还采用了插值平滑技术,对等间隔内样本数分布曲线进行更进一步的平滑处理.本文采用了三值插值,对于第 i 个间隔内的样本数 $Gap\_V[i]$:

$$Gap\_V[i]=(Gap\_V[i-1]+Gap\_V[i+1])/2,\quad 0<i<\xi-1(\xi\text{为间隔数}) \tag{3}$$

1.2.4 基于子空间的聚类算法

算法 1. CLIMB 算法.

输入: 待处理的样本集,其中共计有 n 个样本;

输出: 样本集的层次聚类树.

步骤:

1) 初始化样本类数 *Set_Num*=1,根据文献[14],利用统计方法,从样本集中抽取出特征属性集;
2) 将所有的样本作为一个初始类 *U*[0],根据公式(1)得到等分间隔数 *Div_Gap*;
3) 对特征属性集中的每一个属性 *i* ,依次进行下述操作:
   3.1) *Mid_Set*=*Set_Num*; //将样本类数暂存到*Mid_Set*中
   3.2) 将属性 *i* 对应的维,分割成*Div_Gap*等份;
   3.3) 初始化数组*Gap_V*[*Div_Gap*]为0; //数组*Gap_V*记录投影到相应间隔中的样本数
   3.4) 对已完成的*Set_Num*个样本类中的每一个样本类*U*[*j*],依次进行下述操作:
      3.4.1) 将*U*[*j*]中所有的样本向属性 *i* 对应的维上投影,得到相应间隔中的样本数*Gap_V*;
      3.4.2) 采用公式(3)对从第1个间隔到最后一个间隔中样本数的分布曲线进行平滑处理;
      3.4.3) 初始化间隔游标*t*和山峰记数器*m*为0,山顶判断标记*Is_Peak*=true;
      3.4.4) 如果曲线起点处于山顶,则:
         3.4.4.1) 从第*t*个间隔到最后一个间隔,依次进行如下处理:
            如果(*Is_Peak*=true)或者(*Peak*[*m*]<*Gap_V*[*t*]),则记录第*m*个峰值*Peak*[*m*]=*Gap_V*[*t*],*Is_Peak*=false;
            For (;*Gap_V*[*t*+1]<*Gap_V*[*t*]&*t*<*Div_Gap*−5;*t*++);
            如果没有到达最后一个间隔,则
               如果(*Peak*[*m*]−*Gap_V*[*t*])>*Peak*[*m*]/2,则*Valley*[++*m*]=*t*,*Is_Peak*=true;
               For (;*Gap_V*[*t*+1]>*Gap_V*[*t*]&*t*<*Div_Gap*-5;*t*++);
               从间隔游标*t*处向后,执行循环体(3.4.4.1),寻找下一个山峰;
         3.4.4.2) 如果*m*>0,则第1个山谷*Valley*[0]=−*Valley*[1];
      3.4.5) 如果曲线起点处于山谷,则:
         3.4.5.1) 寻找第1个山谷*Valley*[0]=*t*,使得*Gap_V*[*t*]≥*Gap_V*[0];
         3.4.5.2) 从第*t*个间隔到最后一个间隔,依次进行如下处理:
            For (;*Gap_V*[*t*+1]>*Gap_V*[*t*]&*t*<*Div_Gap*−5;*t*++);
            如果(*Is_Peak*=true)或者(*Peak*[*m*]<*Gap_V*[*t*]),则记录第*m*个峰值*Peak*[*m*]=*Gap_V*[*t*],*Is_Peak*=false;
            如果没有到达最后一个间隔,则

For (;*Gap_V*[*t*+1]<*Gap_V*[*t*]&*t*<*Div_Gap*−5;*t*++);

如果(*Peak*[*m*]−*Gap_V*[*t*])>*Peak*[*m*]/2,则*Valley*[++*m*]=*t*,*Is_Peak*=true;

从间隔游标*t*处向后,执行循环体(3.4.5.2),寻找下一个山峰;

3.4.6) 如果*m*>0(至少有两个山峰),则进行如下的聚类操作,否则,处理下一个样本类:

对于*m*个山峰对应的每一个区间*h*,根据信息量的调节阈*Tuner_APP*,计算出信息区间*h*的范围[*Falve_L*,*Falve_U*];生成一个新的样本集合*U*[*Mid_Set*+*h*],并将其初始化为空;将*U*[*j*]中投影在[*Falve_L*,*Falve_U*]区间内的样本移入*U*[*Mid_Set*+*h*];

更新临时样本类数记录器为*Mid_Set*+=*m*;

如果最后一个山谷距离最后一个间隔还较远,则

取D_i^L=*Valley*[*m*]和D_i^U=*Div_Gap*−1,按公式(2)计算*Tuner_APP*;

如果*Gap_V*[*Div_Gap*−1]>*Peak*[*m*]/2,则

Falve_L=(*Valley*[*m*]+2**Tuner_App*)/*Div_Gap*;

Falve_U=1;

否则,*Falve_L*=(*Valley*[*m*]+*Tuner_App*)/*Div_Gap*;

Falve_U=1−*Tuner_App*/*Div_Gap*;

删除*U*[*j*]中投影不在[*Falve_L*,*Falve_U*]区间内的样本;

否则,*U*[*j*]=*U*[*Mid_Set*−1],删除*U*[*Mid_Set*−1],*Mid_Set*−−; //End of *U*[*j*]

3.4.7) 执行循环体(3.4),处理下一个样本类*U*[++*j*],直到*Set_Num*个样本类都处理完毕;

3.5) *Set_Num*=*Mid_Set*,处理下一个坐标维,直到所有的特征维都处理完为止;

4) 输出样本集的层次聚类树.

CLIMB 算法首先获得特征属性集和特征向量空间,将整个样本集作为一个大类 *U*[0].然后,依次在每一维上进行下述处理.对于每一个已形成的样本类,在当前的向量维上,根据第 1.2.3 节中讨论的策略,寻找样本投影数曲线的山峰(步骤 3.4.1~步骤 3.4.5),并将当前的样本类划分成与山峰相对应的类(步骤 3.4.6).最后,输出样本集的层次聚类树.

1.3 算法性能分析

本文总结了国内外现有的研究工作[1~13],综合而言,信息处理对聚类有 9 项典型的要求.受篇幅所限,本文仅从这几项要求出发,对 CLIMB 算法的性能进行简要分析:

1. 处理噪声数据的能力.在 CLIMB 算法中,采用两种途径来降低噪声干扰的影响,一是通过识别真正有效的山峰和山脉,另一种是采用插值进行预处理,从而提高 CLIMB 算法处理噪声数据的能力.

2. 对于输入样本的顺序不敏感.在 CLIMB 算法中考虑的是每个样本在坐标维上的投影,等到所有的样本均投影完成之后,才寻找一个聚类所对应的山峰,因而输入样本的顺序对 CLIMB 算法的结果没有影响,也就是说,CLIMB 算法对于输入样本的顺序不敏感.

3. 用于决定输入参数的领域知识最小化.CLIMB 算法几乎不需要人为指定参数,为具体应用而引入的调节阈,可以根据具体应用直接计算出来,无须参考领域知识来指定,因而 CLIMB 算法不依赖于领域知识.

4. 复杂度.CLIMB 算法的复杂度为 $\odot(md)$,其中 m 是输入样本数,d 是样本空间的维数,而 CLIQUE 算法需要判断两个聚类是否应该合并的额外运算,因而其复杂度为 $\odot(C^d+md)$[12],其中 C 为常数.由此看来,CLIMB 算法的时空效率优于 CLIQUE 算法.

5. 发现任意形状的聚类.对于类间不完全嵌套缠绕在一起的样本集,CLIMB 算法具有较好的聚类性能,而对于类间完全嵌套缠绕在一起的样本集,则需要对 CLIMB 算法进行进一步的改进,以提高其聚类性能.

6. 可伸缩性,对小数据集和大规模数据都有好的聚类结果.在 CLIMB 算法中考虑的是每个样本在坐标维上的投影,等到所有的样本均投影之后,才寻找一个聚类所对应的山峰,存储空间和运算费用只与等分间隔内的样本数量分布曲线有关,因而,输入样本的数量对 CLIMB 算法的扩展性能没有影响.CLIMB 算法既可以处理小数据集,也可以处理大规模数据.另外,随着输入样本的增多,样本分布特征更加明显,CLIMB 算法可以获得更好

的聚类效果.

7. 高维性.与众多的信息处理算法一样,CLIMB 算法同样也存在“高维诅咒”的问题,而且这也是目前信息研究的关键阻碍.这一问题的解决尚需要广大的研究同仁进一步努力.

8. 可解释性和可用性.由上述算法可以看出,CLIMB 算法的思路很清晰,易于理解,而且其时空效率较好,因而 CLIMB 算法较为实用.

9. 处理不同类型属性的能力.属性类型大致可以分为两大类,一类是数值化的,另一类是非数值化的.大多数的自然语言处理研究,涉及的都是数值化属性,而数据库技术研究既涉及数值化的属性,也涉及非数值化的属性.CLIMB 算法目前只能处理数值化的属性,而对于非数值化的属性,则需要对 CLIMB 算法进行进一步的改进,以提高其聚类性能.

目前较为一致的观点是,还没有一种算法对于所有的应用都具有绝对的优势,不同的应用应选用不同的算法[1~12].CLIMB 算法当然也一样.经过上述分析,我们认为,CLIMB 算法比较适用于涉及数值化属性且类间不嵌套缠绕在一起的聚类应用,现实世界中的大多数应用都属于这类应用,而对于非数值化属性且类间完全嵌套缠绕在一起的样本集,则需要进一步改进 CLIMB 算法,以提高其聚类性能.

2 实验与结果分析

2.1 实验内容与设置

实验语料取自 2002 年的新浪网新闻,采用网络抓取软件从新浪网上抓取相应主题的新闻网页,抽取出新闻主体内容,保存为文本文件.将得到的语料分成 G1 和 G2 两个不同规模的语料库,见表 1.我们之所以取新浪网新闻作为语料,是因为新浪网已经将新闻分好类,可以作为生语料.

Table 1 The testing corpora

表 1 测试文档库

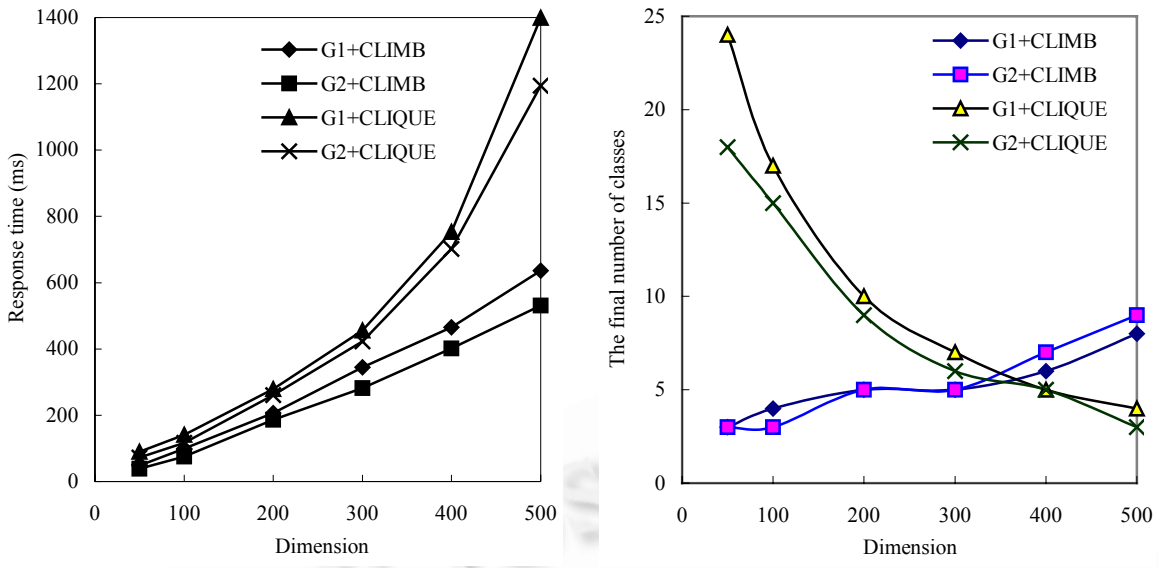
Corpora	G1					G2				
Name of class	Mine	Economics	Environment	Computer	Law	Politics	Electronics	Communication	History	Military
Number of documents	724	680	457	273	272	967	312	221	270	220
Total number	2 406					1 990				

我们之所以采用 G1 和 G2 两个不同规模的语料库进行测试,是为了比较两种算法的准确性以及响应时间与数据集合规模的关系.另外,由于 CLIQUE 和 CLIMB 这两种算法的特殊性,随着维数的增大,两者表现出不同的性能变化规律,维数对这两种算法有很大的影响,因而在不同维数下的性能变化更能反映两者之间的性能差异.而在选择实验维数的范围时,如果维数太小,则不能表达出样本的特性,如果太大,由于出现高维诅咒问题而不利于区分不同聚类算法的性能差异,因而我们认为维数范围取 50~500 较为合适.为此,实验分别取向量空间维数为 50,100,200,300,400,500,然后,分别采用 CLIQUE 和 CLIMB 算法,分别对 G1 库和 G2 库进行聚类,考察两种算法的扩展性能和聚类性能.

实验的硬件环境为 PIII 733MHz 的 CPU 和 256MB 的内存,软件环境为 Windows 2000(Server 版)操作系统,所有代码均用 Visual C++(6.0)实现.

2.2 实验结果与性能分析

图 1 是 CLIQUE 和 CLIMB 算法性能比较实验的结果.图 1(a)的结果与上述性能分析一致,在保持输入样本数量 m 不变和硬件资源充足的前提下,随着样本空间的维数 d 的增大,CLIMB 算法的处理时间几乎呈线性变化,这与其复杂度 $\odot(md)$相一致.而 CLIQUE 算法需要判断两个聚类是否应该合并的额外运算,因而其总处理时间总体上呈现指数变化,这也与其复杂度 $\odot(C^d+md)$相一致.另外,就空间复杂度而言,与 CLIMB 算法相比,CLIQUE 算法需要额外的空间来进行合并运算,而且这一运算所需空间大小不固定,从而造成存储资源的浪费.由此可见,CLIMB 算法具有比 CLIQUE 算法更好的扩展性能,而且效率更高.



(a) Response time versus dimension
(a) 响应时间与维数

(b) The number of clusters versus dimension
(b) 结果聚类数与维数

Fig.1 Comparison of property between CLIMB and CLIQUE algorithms
图 1 CLIMB 和 CLIQUE 算法性能比较

图 1(b)的结果与 CLIQUE 和 CLIMB 两种算法的处理方式一致.CLIQUE 算法采用自底向上的处理方式,先获得小的类,随着维数的增加,在后续处理中,再将这些小类合并成相应的大类,因而随着维数的增加,其结果类数减少.而 CLIMB 算法采用自上而下的处理方式,先将整个样本集作为一个大类,随着维数的增加,在后续处理中,再将这些类分解成相应的小类,因而随着维数的增加,其结果类数增加.从整个实验结果来看,CLIMB 算法具有比 CLIQUE 算法更好的聚类性能,CLIMB 算法的结果聚类数基本上稳定在实际 5 个类的基线左右,而且分类结果基本上与实际的分类一致.单独考虑 CLIMB 算法,可以看到,G1 库的测试结果好于 G2 库的结果.查看分类结果样本的分布,发现其原因是,由于 G2 库中的政治和军事类样本的特征较为相似,因而造成干扰,使得结果性能受到影响.另外,G1 库的语料规模比 G2 库的大,统计信息更加丰富,也是其准确率高于 G2 库的原因.

3 结 论

现有的大多数聚类算法都需要人为地给出一些参数,而且时空效率也有待于进一步提高.然而,在没有先验知识的情况下,人为地确定这些参数是十分困难的.为了解决这一难题,我们提出了一种实用而且高效的聚类算法,无须事先人为确定任何参数,同时提高了时空效率和信息处理结果的性能.通过对该算法的性能进行多角度分析,并将该算法应用于中文文本聚类,有力地证明了该算法不仅不需要人为确定任何参数,而且还具有较优的信息处理时空效率和性能.

然而,通过上面的分析可以看到,尽管 CLIMB 算法在诸多方面性能优越,但是,在少数几个方面仍然需要对其进行改进.今后的研究工作需要考虑如何降低对样本分布空间形状的依赖,同时,我们将和广大研究同仁们一起研究“高维诅咒”问题和非数值化属性空间上的聚类问题.

References:

[1] Karypis G, Han EH, Kumar V. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. Technical Report, #99-007, Department of Computer Science and Engineering, University of Minnesota, 1999.

[2] Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. In: Haas LM, Tiwary A, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Seattle: ACM Press, 1998. 73~84. http://citeseer.ist.psu.edu/guha98cure.html

[3] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS, eds. Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM Press, 1996. 103~114.

[4] Fan M, Meng XF, Translated. Data Mining Concepts and Techniques. Beijing: China Machine Press, 2001 (in Chinese).

[5] Ordonez C, Omiecinski E. FREM: Fast and robust EM clustering for large data sets. In: Kalpakis K, Goharian N, Grossman D, eds. Proc. of the 2002 ACM CIKM Int'l Conf. on Information and Knowledge Management. McLean: ACM Press, 2002. 590~599. http://citeseer.ist.psu.edu/536108.html

[6] Hinneburg A, Keim D. An efficient approach to clustering in large multimedia databases with noise. In: Agrawal R, Stolorz PE, Piatetsky-Shapiro G, eds. Proc. of the 4th Int'l Conf. on Knowledge Discovery and Data Mining (KDD'98). New York: AAAI Press, 1998. 58~65.

[7] Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: Ordering points to identify the clustering structure. In: Delis A, Faloutsos C, Ghandeharizadeh S, eds. Proc. ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM Press, 1999. 49~60.

[8] Ester M, Kriegel H, Sander J, Xu XW. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han JW, Fayyad UM, eds. Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining (KDD'96). Portland: AAAI Press, 1996. 226~231.

[9] Song QB, Shen JY. A Web document clustering algorithm based on association rule. Journal of Software, 2002,13(3):417~423 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/417.pdf

[10] Wang W, Yang J, Muntz RR. STING: A statistical information grid approach to spatial data mining. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. Proc. of the 23rd Int'l Conf. on Very Large Data Bases. Athens: Morgan Kaufmann, 1997. 186~195.

[11] Sheikholeslami G, Chatterjee S, Zhang AD. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In: Gupta A, Shmueli O, Widom J, eds. Proc. of the 24th Int'l Conf. on Very Large Data Bases. New York: Morgan Kaufmann, 1998. 428~439.

[12] Rakesh A, Johanners G, Dimitrios G, Prabhakar R. Automatic subspace clustering of high dimensional data for data mining applications. In: Snodgrass RT, Winslett M, eds. Proc. of the 1994 ACM SIGMOD Int'l Conf. on Management of Data. Minneapolis: ACM Press, 1994. 94~105.

[13] Qian WN, Zhou AY. Analyzing popular clustering algorithms from different viewpoints. Journal of Software, 2002,13(8): 1382~1394 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/1382.pdf

[14] Wang JH, Hu YF. An algorithm to select features based on mutual dependency. Technical Report, No.021011399, Shanghai: Fudan University, 2002 (in Chinese with English abstract).

附中文参考文献:

[4] 范明,孟小峰,译.数据挖掘概念与技术.北京:机械工业出版社,2001.

[9] 宋擒豹,沈钧毅.基于关联规则的 Web 文档聚类算法.软件学报,2002,13(3):417~423.http://www.jos.org.cn/1000-9825/13/417.pdf

[13] 钱卫宁,周傲英.从多角度分析现有聚类算法.软件学报,2002,13(8):1382~1394.http://www.jos.org.cn/1000-9825/13/1382.pdf

[14] 王建会,胡运发.基于互依赖的属性选择算法.科技报告,No.021011399,上海:复旦大学,2002.