文章编号: 1003-0077(2007)06-0065-06

基于 COSA 算法的中文文本聚类

谷 波[1], 李济洪[2], 刘开瑛[1]

(1. 山西大学 计算机与信息技术学院, 山西 太原 030006;
2. 山西大学 计算中心, 山西 太原 030006)

摘 要: 传统聚类算法在计算两个对象间的距离时, 每个属性对距离的贡献相同。COSA (Clustering On Subsets of Attributes) 算法[1]认为在不同的分组中, 每个属性对计算距离所起的作用可能并不相等, 因为不同分组中的对象可能在不同的属性子集上聚集。文献[1]在此基础上定义了新的距离, 并提出了两种 COSA 算法: COSA1 算法是一种分割的聚类算法; COSA2 算法是一种层次聚类算法。为了对比 COSA 距离和传统的欧氏距离在文本聚类中的表现, 本文对中文文本进行了分割聚类和层次聚类的实验。实验结果显示出 COSA 算法较基于欧氏距离的聚类算法有更好的性能, 而且对于属性数的变化, COSA 算法更加稳定。

关键词: 计算机应用; 中文信息处理; 文本聚类; COSA 算法; K-means 算法
中图分类号: TP391 **文献标识码**: A

Chinese Text Clustering Based on COSA Algorithm

GU Bo[1], LI Ji-hong[2], LIU Kai-ying[1]

(1. School of Computer & Information Technology, Shanxi University, Taiyuan, Shanxi 030006, China;
2. Computer Center of Shanxi University, Taiyuan, Shanxi 030006, China)

Abstract: Most traditional clustering algorithms treat each attribute equally. However, COSA [1] (clustering on subsets of attributes) algorithm believes that each separate attribute in different groups may have different weight, and that objects in different groups may cluster in different subsets of attributes. A new distance definition is presented in literature [1], which also presented two COSA algorithms. COSA1 is a partitioning algorithm and COSA2 is a hierarchical cluster algorithm. In this paper, COSA and COSA1 were used for Chinese documents in order to compare the COSA distance and the Euclidean distance. The results show that COSA algorithms achieve better performance and are more robust when the number of attributes changes.

Key words: computer application; Chinese information processing; text clustering; COSA algorithm; K-means

1 引言

聚类是一种无监督的机器学习算法, 它在给定的某种相似性度量下把对象集合进行分组, 使彼此相近的对象分到同一个组内。文本聚类根据文档的某种联系或相关性对文档集合进行有效的组织、摘要和导航, 方便人们从文档集中发现相关的信息。文本聚类方法通常先利用向量空间模型把文档转换成高维空间中的向量, 然后对这些向量进行聚类。由于中文文档没有词的边界, 所以一般先由分词软件对中文文档进行分词, 然后再把文档转换成向量, 最后再进行聚类。

在中文文本聚类中, 有如下一些常用的聚类算法[2]。(1) 层次聚类算法[3]根据给定的距离定义, 计算出每两个对象之间、对象和分组之间以及分组和分组之间的距离, 然后按照距离的大小构建一个聚类层次图。根据算法的起始情况不同, 层次聚类一般分为两种: 自顶向下层次聚类和自底向上层次聚类。较为常用的是自底向上的方法。(2) 分割聚类

收稿日期: 2007-03-12 **定稿日期**: 2007-07-24
基金项目: 国家 863 计划(2006AA01Z142)
作者简介: 谷波(1978—), 男, 讲师, 博士生, 研究方向为自然语言处理; 李济洪(1964—), 男, 硕士, 副教授, 主要研究方向为统计学、自然语言处理; 刘开瑛(1931—), 男, 教授, 博导, 主要研究方向为自然语言处理。

算法是一种划分的方法,这种方法把整个对象集合划分成多个子集。当对象数目较大时,遍历对象集合的全部划分在计算上是不可能的,所以通常采用某种启发式算法对数据进行划分。典型的算法是 K-means [4,5]。(3)基于神经网络的方法。例如基于 SOM 神经网络的文档聚类方法通过大量的训练文本调整神经元的权值,使输出层各节点成为对特定模式类敏感的神经细胞,对应的向量成为各个输入模式类的中心向量。

通常聚类算法在计算距离时,认为所有属性是等权重的。而实际中,每个属性在不同的分组中的影响可能不同,也就是说属性在不同分组中的权重是不同的。例如把词作为属性进行文本聚类,文档集包含两类文本:一类是军事的,另一类是体育的。在聚类时,应当增大军事特征词集在军事文档较多的分组中的权重(对体育亦然),这样才会使更多的文档被分到相应的分组中,从而提高聚类的性能。COSA 就是这样一种基于属性子集的聚类[1,6]的算法,它对传统的距离定义进行了扩展。COSA 算法认为聚类过程中每个分组都可能在不同的属性子集上相近,因而在不同的分组中每个属性的权重并不一定相等。它将每个分组的每个属性权重作为变量加入到聚类准则函数中,然后对准则函数进行极小化从而求得聚类结果。该算法已经被应用在生物领域中,用来发现 DNA 中的基因组[7],但是目前还没有在中文文本聚类中使用该技术的相关文献。我们使用 COSA 算法中文文本进行聚类,并且将 COSA 算法和基于欧氏距离的聚类算法进行了比较。在实验中 COSA 算法取得了较好的结果。

2 COSA 的距离定义

实质上 COSA 算法本身不是具体的聚类算法,其核心是新的距离定义,它在传统的距离定义中增加了对属性权重(属性对聚类的贡献程度)的考虑,本节将简要介绍其距离定义。聚类是将 N 个对象 $\{x_i\}_{i=1}^{N}$(每个对象有 n 个属性),划分到 L 个分组中,使得组内对象比组间对象更加相近,即寻找一个编码函数:$c(i)=l \Rightarrow i \in G_l$,使每个对象 i 映射到一个分组 G_l($1 \le l \le L$)中。极小化如下的聚类准则函数(1)式可以得到一个编码函数。

$$Q = \sum_{l=1}^{L} \frac{W_l}{N_l^2} \sum_{c(i)=l} \sum_{c(j)=l} D_{ij} \qquad (1)$$

这里 W_l是分组 G_l的权重,N_l是分配到第 l 组中的对象的数目,D_{ij}是两个对象之间的距离。通常聚类的每个分组都是平等的,所以 COSA 中没有考虑 W_l。两个对象(i, j)之间的距离可由下式计算:

$$D_{ij} = \sum_{k=1}^{n} w_k d_{ijk} \qquad (2)$$

通常的距离定义中没有考虑 w_k,它表示了第 k 维属性的权重,这也是 COSA 对于传统距离的第一步改进。w_k 满足如下条件,

$$\{w_k \ge 0\}_1^n \quad \text{and} \quad \sum_{k=1}^{n} w_k = 1 \qquad (3)$$

(2)式中的 d_{ijk}可以由式(4)计算。

$$d_{ijk} = \delta_{jk} / s_k \qquad (4)$$

其中 δ_{jk}是两个对象在第 k 维属性上的距离,可以由(5)式(数值属性)或(6)式(类别属性)计算,其中 $I(\cdot) \in \{0,1\}$ 是指示函数,参数为真时取 1,反之取 0。

$$\delta_{ijk} = | x_{ik} - x_{jk} | \qquad (5)$$

$$\delta_{jk} = I(x_{ik} \ne x_{jk}) \qquad (6)$$

s_k是全部对象在第 k 维属性上的平均距离,(4)式中除以 s_k可以对 d_{ijk}进行规范化,s_k由下面(7)式计算。

$$s_k = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_{ijk} \qquad (7)$$

初始时,w_k可以取平均值$\{w_k = 1/n\}_{k=1}^{n}$以表明所有属性权重都相等,这时由(1)式和(2)式,可以变换得到另一种形式的聚类准则函数如下:

$$Q(c) = \sum_{l=1}^{L} W_l \left(\frac{1}{n} \sum_{k=1}^{n} S_{kl} \right) \qquad (8)$$

这里有,

$$S_{kl} = \frac{1}{N_l^2} \sum_{c(i)=l} \sum_{c(j)=l} d_{ijk} \qquad (9)$$

在第 l 个分组内,(9)式度量了组内对象集相对于整个对象集在第 k 维属性上的离散程度,S_{kl}反映出分组内的对象在哪些属性上更加相近。因为属性在不同分组中的权重可能不同,所以为每个分组分别定义属性权重向量。对于分组 G_l的属性权重向量 w_l同样满足如下条件,

$$\{w_{kl} \ge 0\}_1^n \quad \text{and} \quad \sum_{k=1}^{n} w_{kl} = 1 \qquad (10)$$

每个分组有一个 n 维的属性权重向量,L 个分组就形成了 $n \times L$ 的矩阵,这是 COSA 对距离的进一步改进。很显然,若直接将其代入聚类准则函数求解,会将某一属性的权重变为 1(在分组 G_l内,该属性的离散程度最小),而其他属性权重都为零。文献[1]通过引入表示属性权重分布的(11)式(负

熵),达到调节属性权重分布的效果,巧妙解决了上面的问题。负熵类似于信息熵,但是负值,所以当属性权重分布最均匀时,(11)式取最小值。将(11)式加入距离公式得到(12)式。

$$e(w_l) = \sum_{k=1}^{n} w_{kl}\log(w_{kl}) \tag{11}$$

$$D_{ij} = \sum_{k=1}^{n}\{ w_{kl} d_{ijk} + \lambda w_{kl}\log(w_{kl})\} + \lambda\log(n) \tag{12}$$

通过 λ 可以调节负熵 $e(w_l)$ 的本身所占比重,当 λ 增大时,将使分组在更多的属性上聚集。$\lambda\log(n)$ 可使距离依然满足非负性。当分组确定时极小化聚类准则函数 Q,可以得到 w_{kl} 的解如下:

$$w_{kl} = \exp\left(-\frac{S_{kl}}{\lambda}\right) \Big/ \sum_{k=1}^{n}\exp\left(-\frac{S_{kl}}{\lambda}\right) \tag{13}$$

聚类时距离必须在任何两个对象上均有定义。但此时每个分组都有各自的属性权重向量,因此在计算不同分组之间对象的距离时,距离定义不统一。根据两个分组的属性权重向量计算出两个距离然后取最大值,这样就统一了距离的计算,公式如下:

$$D_{ij}' = \max\left(D_{ij}^{\eta}[w_{c(i)}], D_{ij}^{\eta}[w_{c(j)}]\right) \tag{14}$$

其中 D_{ij}^{η} 是以 η 为刻度参数,以 w_k 为权重的 $\left\{d_{ijk}\right\}_{k=1}^{n}$ 的逆指数均值,文献[1]使用 D_{ij}^{η} 计算距离是为了避免算法陷入局部极值。D_{ij}^{η} 计算公式如下:

$$D_{ij}^{\eta} = -\eta\log\left\{\sum_{k=1}^{n} w_k\exp\left(-\frac{d_{ijk}}{\eta}\right)\right\} \tag{15}$$

在实际聚类中,某些属性上常常有一个参照值。比如假设文档在词上的取值被规范在[0,1]内,分组中的多数文档取值都接近 1 的那些词,更可能成为这一分组的特征词,这时的参照值为 1。此时应当考察两个对象在这个属性上,是否同时接近参考值。假设第 k 维属性上的参考值是 t_k,(x_{ik}, x_{jk})分别是对象 i 和 j 在第 k 维属性上相应的值,用(16)～(18)式替换(4)～(6)式便是 COSA 中的标靶聚类,这又是 COSA 对传统距离的改进。

$$d_{ijk}(t_k) = \max\{ d_k(x_{ik}, t_k), d_k(x_{jk}, t_k)\} \tag{16}$$

对于数值属性 d_k 用(17)式计算,对于类别属性使用(18)式。

$$d_k(x, t_k) = |x - t_k|/s_k \tag{17}$$

$$d_k(x, t_k) = I(x \neq t_k)/s_k \tag{18}$$

至此,从聚类的准则函数出发,COSA 逐步对距离的定义进行扩展,并对扩展后出现的问题进行巧妙的解决,最终得到了可计算的引入了属性权重和参考值的距离定义。

3　基于 COSA 的中文文本聚类

在文本聚类中,为了方便处理,通常将文档表示成一个高维空间中的向量,然后再进行聚类。通常用词作为属性(向量空间的维),因此属性的数量会很大,常常大于文档的数目。在这种情况下,通常会有很多与聚类无关的属性。如果我们平等对待所有的属性,这些不相关的属性可能会有负面影响。另一方面,分组内的文档可能仅仅在某些属性子集上比较相近,而非在整个属性集合上相近。COSA 算法正是考虑了上述情况,把分组的属性权重向量作为变量,而达到在不同的属性子集空间上进行聚类的目的,这正符合文本聚类的真实情况。

3.1　向量空间模型

中文的词与词之间没有像其他语言一样的天然间隔,因此首先要对中文文档进行分词,然后再过滤其中的停用词和低频词。按某种方式对词进行排序,每一个词作为一个属性。之后对分词的文档进行词频统计,一篇文档中对应的词出现频率或经过计算后得到的值(通常称为权重,本文为了避免和属性权重相混而称为值),作为这篇文档在该属性上的值。这样一篇文档就对应了一个向量,这就是向量空间模型(Vector Space Model)。实际应用中除了考虑文档中出现的词的频率,还要考虑该词在整个文档集合中的分布情况和所在文档的长度。我们在实验中计算词 w 在文档中的值的时候,采用了常用的 $TF\text{-}IDF$ 公式,如(19)～(21)式。

$$TFIDF(w) = TF(w) \times IDF(w) \tag{19}$$

其中 TF 和 IDF 的计算式分别如下,

$$TF(w) = 0.5 + 0.5 \times freq/fmax \tag{20}$$

$$IDF(w) = \log(N/DF(w) + 0.01)/\log(N + 0.01) \tag{21}$$

$freq$ 是词 w 在当前文档中出现的频率,$fmax$ 是当前文档中出现的词的最大频率。N 是文档总数,$DF(w)$ 是出现 w 的文档数。这样得到的值既考虑了 w 在当前文档中出现的频率,又考虑了 w 在文档集合中的分布,除以 $fmax$ 可以在一定程度上消除文档长度的影响。(20)式和(21)式中的常数 0.5 和 0.01 是用来对数据起平滑作用的。经过(19)式计算得到的词的值是一个在 0 到 1 之间的数。在一些相同的词上都取很大的值的文档更加趋于同一类,所以我们在实验中进行了标靶聚类,将 1 作为每

个属性的参考值。

3.2 COSA算法

文献[1]中有COSA算法的详细推导和描述，其中包含两个算法：给定聚类数目进行分割聚类的COSA1算法；通过计算出COSA距离进而进行层次聚类的COSA2算法。这里需要说明的是，COSA1是一个迭代求解属性权重和聚类的过程，即先给定分组求出属性权重，然后根据新的属性权重利用某种聚类算法在进行聚类划分，然后在新的分组下求再属性权重，循环一直进行直至收敛或达到指定次数。算法中的第六步需要引用另外的聚类算法来得到一个聚类结果，然后根据这个结果修正属性的权重。我们结合常用的K-means算法用C++实现了COSA1算法。COSA1算法描述如下：

1. Read λ, α, cluster number L, and set $\eta=\lambda$, initialize each cluster's weight vector $\mathbf{W}=\{w_{kl}=1/n\}$, $(1\leq k\leq n; 1\leq l\leq L)$. ($\eta$和$\lambda$含义同上述公式一样；$\alpha$控制$\eta$增长速率)
2. Randomly choose objects as clusters' centers.
3. Loop1 {
4. Loop2 {
5. Compute distances $D_{ij}'[\mathbf{W}]$ between each object and each center with equations (14) and (15).
6. Update clusters using the K-means algorithms.
7. Update clusters' centers.
8. } End loop2 until centers stabilize.
9. Compute new weights with equations (9) and (13).
10. Set $\eta=\eta+\alpha\lambda$.
11. } End loop1 until $\mathbf{W}$ stabilizes.

在层次聚类中，每一个对象或分组是一个实体，因此距离D_{ij}应该能在所有对象之间进行计算并且应用在层次聚类算法当中。然而在COSA1中，距离的计算与分组的属性权重向量w_k有关，自下而上聚类，当单个对象自成一类时，w_k无从计算。在COSA2中，通过为每一个对象寻找K个最近的对象形成一个分组，从而得到相关的属性权重向量的值。设$KNN(i)$是与对象i最近的K个对象的集合，则有：

$$KNN(i)=\{j\mid D_{ij}\leq d_i(K)\} \tag{22}$$

上式中$d_i(K)$是$\{D_{ij}\}_{j=1}^{N}$经过升序排列后的第K个值。每个对象在属性i上的离散程度就可以用下式来计算。

$$S_{ki}=\frac{1}{K}\sum_{j\in KNN(i)}d_{ijk} \tag{23}$$

由(23)式，可以推导出对象i的在第k维属性上的权重w_{ki}如下：

$$w_{ki}=\exp\left(-\frac{S_{ki}}{\lambda}\right)\Big/\sum_{k=1}^{n}\exp\left(-\frac{S_{kl}}{\lambda}\right) \tag{24}$$

计算所有对象之间距离的COSA2算法如下：

1. Initialize: read η, α and set $\eta=\lambda$; $\mathbf{W}=\{1/n\}$.
2. Loop{
3. Compute distances D_{ij}' equations (14) using (15).
4. Compute the K nearest neighbors for each point - equation (22).
5. Compute weights $\mathbf{W}=\{w_{ki}\}$ - equation (23) & (24).
6. Update $\eta=\eta+\alpha\lambda$.
7. } End loop until W stabilizes.
8. Output the distances $\{D_{ij}=D_{ij}[W]\}$.

由上述算法得到距离$\{D_{ij}\}$，然后就可以进行层次聚类。

4 实验结果及分析

实验所用的三个文档集全部来自网络，为了方便计算，我们对下载的分类语料库进行一些抽取。文档集一来自http://www.nlp.org.cn/ 由李荣陆提供，我们从中抽取了5类，每类500篇；文档集二来自http://www.inforsec.org.cn/tansongbo/ 由谭松波提供，我们从中抽取了5类，每类1 000篇；文档集三来自http://www.sogou.com/labs/dl/c.html 由搜狐研发中心提供，我们从中抽取了5类，每类10篇。在此对他们表示感谢。

4.1 评价指标

我们使用常用的聚类评价指标纯度和熵值对我们的实验结果进行评价[8,9]。假设C是聚类算法形成的文档分组集合，$C=\{c_1,c_2,\cdots,c_k\}$；集合X是手工标注的文档类别集合，$X=\{X_1,X_2,\cdots,X_q\}$。对于含有n_i个文档数的分组c_i，纯度计算公式如下：

$$S(c_i)=\frac{1}{n_i}\max(n_i^j) \tag{25}$$

这里n_i^j是分组c_i和类别X_j交集的文档数。整个聚类结果的纯度可以计算为：

$$purity=\sum_{i}^{k}\frac{n_i}{n}S(c_i) \tag{26}$$

这里k是分组个数。纯度用来刻画整个聚类结果相对于手工分类结果的准确程度，所以纯度越高

聚类效果越接近手工分类结果。另外一个常用的聚类评价指标是熵值,熵值刻画了聚类结果相对于手工分类结果的混乱程度。所以熵值越大聚类结果越偏离于手工分类的结果。含有 n_i 个文档的分组 c_i 的熵值计算如下:

$$E(c_i) = -\frac{1}{\log q_j}\sum_{j=1}^{q}\frac{n_i^j}{n_i}\log\frac{n_i^j}{n_i} \qquad (27)$$

上式中 q 是手工分类结果的类别总数。整个聚类结果的熵值计算如下:

$$Entropy = \sum_{i=1}^{c}\frac{n_i}{n}E(c_i) \qquad (28)$$

4.2　分割聚类实验结果

为了在实验中观察属性数(即词的个数)变化对聚类结果的影响(即聚类所用词的个数),我们在数目不同数的属性集上进行了 COSA1 和基于欧氏距离的 K-mean 算法的比较。由于 K-means 算法是一种随机确定初始点的分割聚类算法,所以为了减少随机选取初始点对聚类的影响,我们在每组数据上进行五次聚类实验,然后取平均值作为最后的结果。实验结果见表 1～表 3。

表 1　数据集 1 实验结果

指标 / 属性数	纯　度		熵　值	
	cosa 距离	欧氏距离	cosa 距离	欧氏距离
1 064	0.719	0.699	0.446	0.426
1 789	0.759	0.675	0.419 6	0.463
2 405	0.705	0.697	0.456	0.429
3 166	0.746	0.675	0.417	0.455
4 829	0.775	0.610	0.396	0.503

表 2　数据集 2 实验结果

指标 / 属性数	纯　度		熵　值	
	cosa 距离	欧氏距离	cosa 距离	欧氏距离
509	0.555	0.585	0.485	0.449
1 295	0.580	0.585	0.457	0.453
1 899	0.601	0.583	0.445	0.450
2 878	0.607	0.573	0.438	0.531
3 626	0.620	0.563	0.408	0.553

从以上结果可以看到总体上 COSA1 的纯度高于基于欧氏距离的 K-means 算法,熵值小于基于欧氏距离的 K-means算法。另外可以看到,即使不进行特征的选择 COSA1 的表现仍然很好。总体上随着属性的增加,COSA1 算法性能基本不变或略有增加,而传统的 K-means 算法性能下降较大,表明在文本聚类中属性数的大小对于 COSA1 的影响很小。这说明 COSA1 算法比传统聚类算法更适合在没有经过特征选择的高维空间中进行聚类。

表 3　数据集 3 实验结果

指标 / 属性数	纯　度		熵　值	
	cosa 距离	欧氏距离	cosa 距离	欧氏距离
339	0.520	0.520	0.653	0.640
801	0.560	0.300	0.661	0.899
1 086	0.600	0.400	0.590	0.825
1 539	0.640	0.300	0.528	0.899
2 583	0.660	0.280	0.565	0.918

4.3　层次聚类实验结果

为了详细的观察 COSA 聚类的结构,我们进行了基于 COSA2 的层次聚类实验。基于欧氏距离的层次聚类使用统计软件 R 中自带的算法包,COSA2 层次聚类算法可从以下网站获得 http://www-stat.stanford.edu/～jhf/COSA.html.。这两个算法都运行在 R 系统下。为了使得聚类层次图显示清晰,我们从文档集 1 中,每类文档随机抽取出 50 篇,共计 250 篇作为层次聚类的数据。最终的聚类层次图如图 1 和图 2 所示。

图 1 和图 2 中的每一个点都代表一个文档。我们将 250 篇文档按照手工分类的类别顺序从左到右排列,也就是从第一篇文档开始每 50 篇文档属于同一类别。从图 1 和图 2 可以看出基于 COSA2 的层次聚类结果明显优于基于欧氏距离的层次聚类。从图 1 看,在不同的属性数下,基于 COSA2 的层次聚类基本给出了比较清晰的聚类层次图,这说明基于 COSA2 的层次聚类受属性数的影响不大。从图 2 上看,基于欧氏距离的层次聚类很难发现清晰的分组,特别是当属性数增加时(图 1c 和图 2d),所有的文档几乎都连接到了一起,这说明属性数的变化对基于欧氏距离的聚类影响很大。

5　总结和展望

从实验结果可以看出: 不论是在分割聚类中,还是在层次聚类中,基于 COSA 的文本聚类明显优

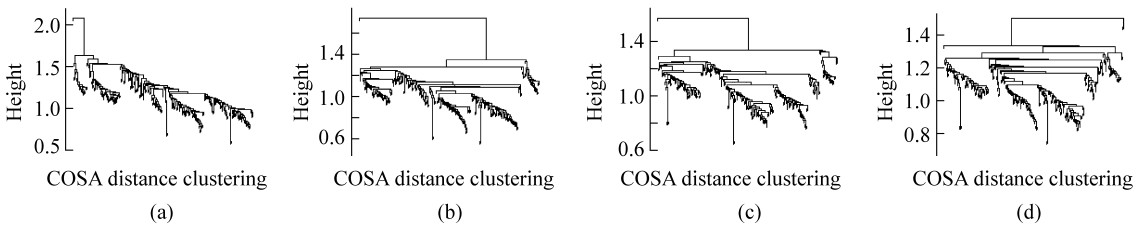

图 1 基于 COSA 距离的聚类层次图(图中 a ,b ,c ,d 分别表示在属性数为 706 ,1 500 ,2 350 和 3 270 时的结果)

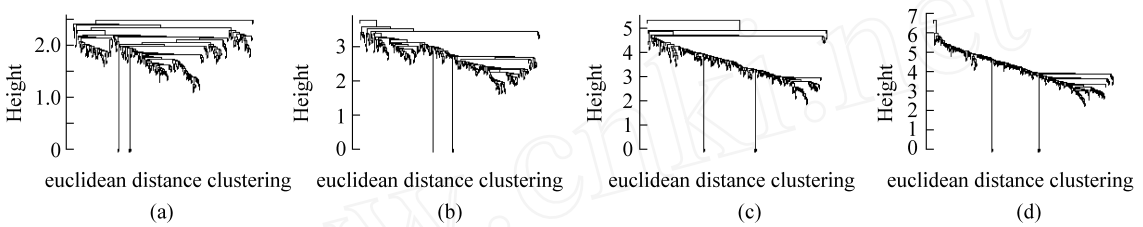

图 2 基于欧氏距离的聚类层次图(图中 a ,b ,c ,d 分别表示在属性数为 706 ,1 500 ,2 350 和 3 270 时的结果)

于传统的文本聚类。而且 COSA 的聚类更加稳定,不容易受到属性维数变化的影响。这是因为 COSA 和传统的聚类算法不同,它考虑了不同的分组可能聚集在不同的属性子集上,且引入了属性权重向量,并在这一基础上进行聚类准则函数的极小化。这样做可以减少那些与聚类不相关的词所带来的负面影响,更符合文本聚类的实际情况。因此在文本聚类实验中,COSA 聚类算法表现了良好的效果。

COSA 改进了聚类中的距离定义,它把属性权重作为变量,使得每个属性在不同分组中所起作用不同。由于 COSA1 算法是一个聚类和求解属性权重向量的迭代过程,所以除了 K-means 它还可能和其他的聚类算法相结合,我们下一步将继续研究 COSA1 和其他聚类算法的结合,从而看其是否能对其他的聚类算法的性能有所提高;另外 COSA 算法对每个聚类的分组都有一个属性权重向量,我们将进一步研究如何利用 COSA 算法对文本集合进行特征词的自动提取。

参考文献:

[1] Jerome H Friedman, Jacqueline J Meulman. Clustering objects on subsets of attributes [J]. J R Statist Soc B, 2004, 66(4): 1-25.

[2] 刘远超等. 文档聚类综述 [J]. 中文信息学报, 2006, 20(3):55-62.

[3] 曼宁 D. C. 统计自然语言处理基础[M]. 苑春法,李庆中, 王昀等译. 第一版. 北京: 电子工业出版社,2005.

[4] 孙即祥,等. 现代模式识别[M]. 长沙: 国防科技大学出版社, 2002.

[5] 范明, 孟晓峰. 数据挖掘概念与技术[M]. 北京: 机械工业出版社. 2001,8.

[6] Lance Parsons, Ehtesham Haque Ehtesham, Huan Liu. Evaluating subspace clustering algorithms [J]. In Workshop on Clustering High Dimensional Data and its Applications, SIAM Int. Conf. on Data. Mining: 48-56, 2004.

[7] John Allen. Bridging Microarray Platforms To Extend the Utility of Gene Expression Profile [D], Science School of Informatics University of Edinburgh, 2004.

[8] Steinbach M, Karypis G, Kumar V. A comparison of Document Clustering Techniques [A]. Department of Computer Science and Engineering, University of Minnesota. Technical Report # 00-034, 2000.

[9] Zhao Y, Karypis G. Criterion Functions for Document Clustering Experiments and Analysis [A]. Technical Report # 01-40, Department of Computer Science, University of Minnesota, Minneapolis, MN, 2001.