第32卷 第5期 北京工业大学学报 Vol.32 No.5
2006年 5月 JOURNAL OF BEIJING UNIVERSITY OF TECHNOLOGY May 2006

分层子树合并聚类算法

李玉鑑
(北京工业大学 计算机学院 多媒体与智能软件技术北京市重点实验室, 北京 100022)

摘 要: 为了解决传统分层合并聚类算法可能产生不唯一的二叉树结果问题,提出了分层子树合并聚类算法,其基本思想是通过在数据集的最小树中分析 θ-极大紧邻子树然后合并它的顶点集,该算法每步可将多个对象聚类,计算结果用多叉树表示. 在理论上证明了该树在不计分支次序时是唯一的,并且通过计算实验说明,在样本中存在较多距离彼此相等的点对时,该树所描述的聚类结果要明显比传统分层合并聚类算法用二叉树描述的聚类结果更为合理.

关键词: 分层合并聚类算法; 分层子树合并聚类算法; 最小树; 极大紧邻子树; 聚类
中图分类号: TP 18 **文献标识码**: A **文章编号**: 0254 - 0037(2006)05 - 0442 - 05

对数据样本进行聚类,在本质上就是把其中相似的样本分别形成一类的过程. 聚类技术在语音和图像处理、生物信息计算以及数据挖掘等领域有着非常广泛的应用[1-4]. 经典的聚类技术主要包括 K-均值算法,分层合并聚类算法和基于最小树的算法[5-7]. 一些较新的聚类方法可以在文献[8-13]中找到.

在分析样本集中的层次聚类结构时,通常采用传统的分层合并聚类算法,其基本思想是先让每个样本自成一类,然后每步把 2 个距离最近的类合并成 1 个新类,直到满足某个终止条件或只有 1 个类. 虽然传统的分层合并聚类算法能够给出一个用二叉树表示的分层聚类结构,但不能保证结果的唯一性,在理论上不够完美,在应用上可能产生冲突. 文献[14]提出一种智能控制方法,通过合理选择距离测度对传统分层合并聚类的过程进行控制,尽管可以根据需要对聚类结果进行一定的调整和优化,但是在理论上不能保证聚类结果的唯一性. 本文的目的在于解决这一问题.

1 最小树和极大紧邻子树的定义及其性质

用 $G=(V, E)$表示顶点集为 V,边集为 E 的图,用 $|V|$和 $|E|$分别表示图 G 的顶点数和边数. $\forall e\in E$,其长度(或权重)记为 $W(e)>0$. 如果 $V_1\subset V$, $V_2\subset V$,则用 $E[V_1, V_2]$表示 G 中一个端点属于 V_1 且另一个端点属于 V_2 的所有边的集合. 如果 $V'\subset V$, $E'\subset E$,那么 $G'=(V', E')$称为 G 的子图;用 $G[V']$表示 G 的点诱导子图,其中顶点集为 V',边集为 E 中连接 V'的顶点的所有边;用 $G[E']$表示 G 的边诱导子图,其中顶点集为 E'的所有顶点,边集为 E'.

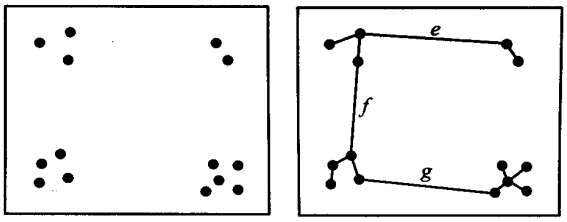
$G=(V, E)$的子树是指 G 的一个连通且无回路的子图 $T=(V_T, E_T)$,易知 $|V_T|=|E_T|+1$. 如果 $|V_T|=1$,则 T 称为单点树;如果 $V_T=V$,则 T 称为 G 的支撑树,其长度定义为: $W(T)=\sum_{e\in E_T}W(e)$

不难看出,当且仅当 G 是连通图时,才有支撑树. 用 $T(G)$表示图 G 的所有支撑树的集合. 如果连通图 G 的支撑树 T 的长度 $W(T)$满足: $W(T)=\min\{W(T')|T'\in T(G)\}$;那么 T 称为 G 的最小树. 值得一提的是,最小树可能不唯一.

对任意一个具有某种距离 ρ 的集合 $X=\{x_1, x_2, \cdots, x_n\}$来说,也可以定义最小树的概念. 令顶点集 $V=X$,边集 $E=\{(x_i, x_j)|x_i, x_j\in X, i\neq j\}$,其中$(x_i, x_j)$的长度 $W(x_i, x_j)=\rho(x_i, x_j)$,如果把 X 关于

收稿日期: 2005-07-11.
基金项目: 北京市自然科学基金资助项目(4052005).
作者简介: 李玉鑑(1968 -), 男, 湖南邵东人, 副教授.

距离 ρ 的诱导图定义为 $G_\rho(X)=(V,E)$，那么 X 的最小树可定义为 $G_\rho(X)$ 的最小树. 例如，图 1(b)是连接图 1(a)中所有数据点的最小树，其中边长采用欧氏距离计算.



(a) 数据点集　　(b) 最小树

图 1　数据点集及其最小树表示

Fig.1　A dataset and its minimal spanning

在最小树的基础上，可以进一步建立极大紧邻子树的概念. 如果把空集记为 $\emptyset$，并定义 $\max\emptyset=0$，$\min\emptyset=+\infty$，那么图的极大紧邻子树可定义为：

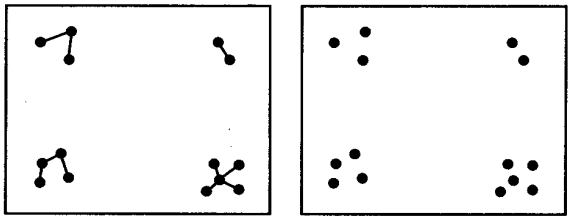
定义 1　如果 $T=(V_T,E_T)$ 是图 $G=(V,E)$ 的 1 棵子树，那么 T 称为 G 的 θ-极大紧邻子树 ($\theta\geqslant 0$)当且仅当它满足下面 3 个性质.

1) 最小性：T 是点导出子图 $G[V_T]$ 的最小树；

2) 紧邻性：$\max\{W(e)\mid e\in E_T\}\leqslant\theta$；

3) 极大性：$\min\{W(e)\mid e\in E[V_T,\overline{V}_T]\}$，其中 $\overline{V}_T=V-V_T$.

如果 $X=\{x_1,x_2,\cdots,x_n\}$ 是 1 个定义了某种距离 ρ 的集合，则把 X 的 θ-极大紧邻子树定义为 $G_\rho(X)$ 的 θ-极大紧邻子树. 显然，θ-极大紧邻子树也可能不唯一. 图 1(b)中的最小树是极大紧邻子树的 1 个例子，如果去掉其中的 e、f、g 3 条边，得到图 2(a)中的 4 棵极大紧邻子树；去掉图 2(a)中的所有边，得到图 2(b)中的单点极大紧邻子树.



(a) 4棵极大紧邻子树　　(b) 单点极紧邻子树

图 2　极大紧邻子树举例

Fig.2　Examples of maximal θ-distant subtrees

最小树和极大紧邻子树的概念是本文建立分层子树合并聚类算法的基础，其中最小树的 2 个重要性质描述在定理 1 和 2 中，θ-极大紧邻子树的 1 个重要性质描述在定理 3 中.

定理 1　设 T_1 和 T_2 是 $G=(V,E)$ 的任意 2 棵最小树，如果把它们的边分别记为 $e_1^1,e_2^1,\cdots,e_m^1$ 和 $e_1^2,e_2^2,\cdots,e_m^2$，则存在 $1,2,\cdots,m$ 的 1 个排列 $j_1,j_2,\cdots,j_m$，使得 $W(e_k^1)=W(e_{j_k}^2)$，$k=1,2,\cdots,m$.

证明：不妨设 $T_1\neq T_2$，令 $e\in T_1\setminus T_2$，则 T_2+e 包含唯一的 1 个回路 $C(e)$，并且 $C(e)$ 上至少有 1 条边 $e'\in T_2\setminus T_1$. 假定 $W(e)<W(e')$，则支撑树 $T'=T_2+e-e'$ 的长度 $W(T')<W(T_2)$，这与 T_2 是最小树矛盾，从而 $W(e)\geqslant W(e')$. 假定 $W(e')<W(e)$，则支撑树 $T''=T_1+e'-e$ 的长度 $W(T'')<W(T_1)$，这与 T_1 是最小树矛盾，从而 $W(e')\geqslant W(e)$. 综上所述 $W(e')=W(e)$，因此可以把 e' 定义为 e 的对应边.

由于 T' 是 G 的最小树，所以可令 $T_2=T'$ 重复上述过程直到 $T'=T_1$. 于是，不难在 $T_1\setminus T_2$ 和 $T_2\setminus T_1$ 的所有边中建立一一对应关系并使得对应边相等，从而在 T_1 和 T_2 的所有边中建立一一对应关系并使得对应边相等. 证毕.

定理 2　设 T_1 和 T_2 是 $G=(V,E)$ 的任意 2 棵最小树，它们都包含 $p-1$ 条大于 $\theta\geqslant 0$ 的边，从 T_1 和 T_2 中去掉 $p-1$ 条边后，如果把它们产生的 p 个连通分支分别记为 $T_k^1=(V_k^1,E_k^1)$ 和 $T_k^2=(V_k^2,E_k^2)$ $(k=1,2,\cdots,p)$，则存在 $1,2,\cdots,p$ 的 1 个排列 $j_1,j_2,\cdots,j_p$，使得：$V_k^1=V_{j_k}^2$，$k=1,2,\cdots,p$.

证明：因为 $V=\bigcup_{k=1}^{p}V_k^1=\bigcup_{k=1}^{p}V_k^2$ 且 $V_i^1\cap V_j^1=V_i^2\cap V_j^2=\emptyset$ $(1\leqslant i\neq j\leqslant p)$，所以对任意 V_l^1，存在 V_m^2 使得 $V_l^1\cap V_m^2\neq\emptyset$ $(1\leqslant l,m\leqslant p)$. 采用反证法来证明 $V_l^1=V_m^2$.

假定 $V_l^1\neq V_m^2$，则 $V_l^1-V_m^2\neq\emptyset$ 或 $V_m^2-V_l^1\neq\emptyset$. 不妨设 $V_l^1-V_m^2\neq\emptyset$，则存在顶点 $u\in V_l^1-V_m^2$ 和顶点 $v\in V_l^1\cap V_{m'}^2$，使得 $(u,v)\in E_l^1$ 且 $(u,v)\notin E_{m'}^2$，显然 $W(u,v)\leqslant\theta$. 设 $u\in V_{m'}^2$ $(m'\neq m)$ 且 (u',v') 是在 T_2 中连接 $T_{m'}^2$ 和 T_m^2 的边. 令 $T'=T_2+(u,v)-(u',v')$，则 T' 也是 G 的支撑树. 因 $W(u',v')>$

θ,故 T'的长度 $W(T')<W(T_2)$,这与 T_2 是 G 的最小树矛盾,从而 $V_l^1=V_m^2$.

因此可以把 V_m^2 定义为 V_l^1 的对应顶点集. 根据 V_l^1 的任意性知,存在 $1,2,\cdots,p$ 的 1 个排列 $j_1,j_2,\cdots,j_p$,使得:$V_k^1=V_{j_k}^2$, $k=1,2,\cdots,p$. 证毕.

定理 3 设 $G=(V,E)$是连通图,如果 $T=(V_T,E_T)$是 G 的最小树,那么对任意 $\theta\geqslant 0$,从 T 中去掉所有大于 θ 的边后,把产生的 p 个连通分支分别记为 $T_k=(V_k,E_k)(k=1,2,\cdots,p)$,则 T_k 都是 G 的 θ-极大紧邻子树,且在不记次序时$\{V_k\}_{k=1}^{p}$形成 V 的唯一划分,即 $V=\bigcup_{k=1}^{p}V_k$, $\forall 1\leqslant i,j,k\leqslant q, i\neq j, V_i\cap V_j=\varnothing$, $V_k\neq\varnothing$,同时点导出子图 $G[V_k](k=1,2,\cdots,p)$的所有最小树都是 G 的 θ-极大紧邻子树.

证明:可由定理 2 及 θ-极大紧邻子树的定义直接得到,详细证明略.

2 分层子树合并聚类算法

定理 3 表明,在获得 1 个连通加权图 $G=(V,E)$的某棵最小树 T,并合理选择不同大小的 θ 值后,就可以通过从 T 中切割所有长度大于θ 的边来确定图G 的 1 组 θ-极大紧邻子树,同时获得 V 的 1 个划分. 前面已经指出,对任意 1 个具有某种距离 ρ 的集合$X=\{x_1,x_2,\cdots,x_n\}$来说,都可以定义 1 个 X 关于距离 ρ 的诱导图$G_\rho(X)$. 由于 $G_\rho(X)$是 1 个加权图,因此可以直接利用定理 3 进行顶点集划分,从而形成 X 的聚类. 显然只需选择不同的 θ 值,同时合并相应 θ-极大紧邻子树的顶点集,即可在不同的层次上形成 X 的聚类结构. 基于这一思想,我们对传统的分层合并聚类算法进行改进.

对传统的分层合并聚类算法来说,核心思想是每步只将 2 个距离最近的对象聚类,因此在数据集中存在较多距离彼此相等的对象时,就存在多种可以选择的情况,从而不能保证计算结果的唯一性. 可是,如果事先在数据集的最小树中分析由所有最短边形成的极大紧邻子树,那么由于极大紧邻子树的顶点集可以形成数据集的唯一划分,因此每步合并这些极大紧邻子树的顶点集就能保证聚类结果的唯一性. 在这里,把通过反复合并极大紧邻子树的顶点集进行聚类的方法称为分层子树合并聚类算法.

在进行子类合并时,传统的分层合并聚类算法通常可选择 3 种不同距离,从而产生 3 个不同的变种,分别称为单边连接、完全连接和平均连结算法,其中所使用的距离对应于最小距离($d_{\min}$)、最大距离($d_{\max}$)和平均距离(d_{avg}). 如果 $V\subseteq X$ 且 $V'\subseteq X$, ρ 是定义在 X 上的 1 种距离,那么这 3 种距离可描述为:

$$d_{\min}(V,V')=\min_{x\in V,x'\in V'}\rho(x,x');\ d_{\max}(V,V')=\max_{x\in V,x'\in V'}\rho(x,x');$$

$$d_{\text{avg}}(V,V')=\frac{1}{|V||V'|}\sum_{x\in V}\sum_{x'\in V'}\rho(x,x').$$

参照传统分层合并聚类算法的 3 个变种,可以把分层子树合并聚类算法统一描述如下:

1) 令 $C_i=\{x_i\}$, $i=1,2,\cdots,n$, $\Omega=\{C_1,C_2,\cdots,C_n\}$,并选择距离 d 为$d_{\min}$、$d_{\max}$或 d_{avg}三者之一;

2) 计算图 $G_d(\Omega)$中的最短边长 $\theta=\min\{d(C_i,C_j)\mid C_i\in\Omega,C_j\in\Omega,i\neq j\}$;

3) 计算图 $G_d(\Omega)$中所有最短边长的集合 $E_S=\{(C_i,C_j)\mid d(C_i,C_j)=\theta,C_i\in\Omega,C_j\in\Omega,i\neq j\}$;

4) 把 E_S 中的所有边枚举为$e_1,e_2,\cdots,e_m$;

5) 构造一个新图 $G=(V,E)$,其中 $V=\Omega$, $E=\varnothing$;

6) 在不产生回路的条件下把 $e_1,e_2,\cdots,e_m$ 依次加入G 中;

7) 把 G 的所有连通分支记为 $T_k=(V_k,E_k)(k=1,2,\cdots,p)$;

8) 计算新的聚类 $C_k=\bigcup_{C\in V_k}C(k=1,2,\cdots,p)$,然后令 $\Omega=\{C_1,C_2,\cdots,C_p\}$;

9) 如果 $p>1$,返回 2);否则停止.

在上述算法中,第 6 步在不产生回路的条件下把 $e_1,e_2,\cdots,e_m$ 依次加入G 中,其理论依据是计算最小树的 Kruskal 算法[15],这样可以保证第 7 步的连通分支 $T_k=(V_k,E_k)(k=1,2,\cdots,p)$都是 θ-极大近邻子树,从而能够直接利用定理 3 保证计算结果的唯一性.

根据所选距离 d 是$d_{\min}$、$d_{\max}$还是 d_{avg},也可类似地把分层子树合并算法分别称为单边连接、完全连

接和平均连结算法. 根据定理 3,这 3 种算法的最终计算结果都将是唯一的. 计算结果的唯一性使分层子树合并算法比传统的分层合并聚类算法更优越,因为后者不能保证聚类结果的唯一性. 此外,分层子树合并算法的聚类结果一般用多叉树表示,而传统的分层合并聚类算法的聚类结果一般用二叉树表示.

3　计算实验

为了验证分层子树合并聚类算法的有效性和合理性,选取了 1 组 2 维数据进行计算实验,数据集用 DS 表示,其中共包含 144 个样本,且存在较多距离彼此相等的点对(见图 3). 如果把单个样本和整个数据集都算作聚类,那么从人的感知来看,在 DS 中应存在 4 个不同层次的聚类.

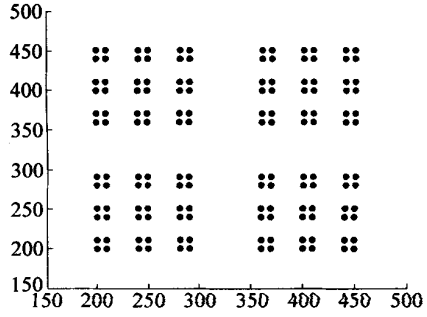
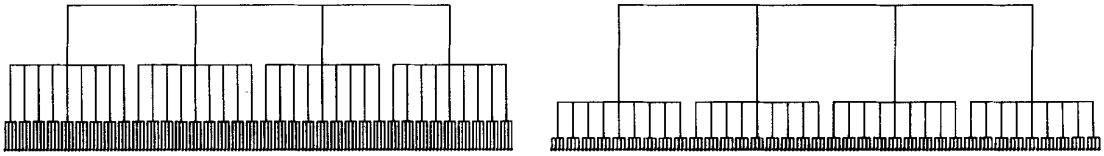


图 3　1 组 2 维数据,共 144 个样本,简称 DS

Fig. 3　A 2-D dataset containing 144 points, abbreviated as DS

如果采用分层子树合并聚类算法对 DS 聚类,那么单边连接和平均连结算法的聚类结果可分别用图 4(a)、(b)中的多叉树表示. 不难看出,虽然 2 棵树的分支长度存在着差别,但是它们的拓扑结构是相同的,而且恰好都分为 4 个不同层次,与人类的感知相同. 此外,完全连接算法的聚类树与它们的拓扑结构也相同,只是分支长度略有差别.

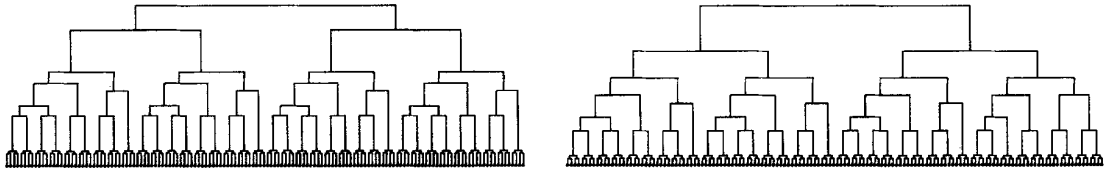
如果采用传统的分层合并聚类算法对 DS 聚类,那么单边连接算法的聚类结果在显示时与图 4(a)相同,虽然它看起来是多叉树,但是在本质上是二叉树. 平均连结算法和完全连接算法的聚类结果分别用图 5(a)、(b)中的二叉树表示. 不难看出,2 棵树的拓扑结构也是相同的,只是分支长度略有差别,但是它们的分层过多,与人类的感知不一致.



(a) 单边连接　　(b) 平均连接

图 4　分层子树合并聚类算法对 DS 的聚类结果

Fig. 4　The clustering results of DS by HSACA



(a) 平均连接　　(b) 完全连接

图 5　传统分层合并聚类算法对 DS 的聚类结果

Fig. 5　The clustering results of DS by HACA

因此,分层子树合并聚类算法给出的聚类结果比传统的分层合并聚类算法更为合理.

4　结束语

本文提出的分层子树合并聚类算法通常给出 1 棵多叉树作为聚类结果,同时在不计分支次序时能够保证聚类结果的唯一性. 计算机仿真实验表明,在样本集中存在较多距离彼此相等的点对时,分层子树合并聚类算法给出的聚类结果要明显比传统算法给出的结果更为合理.

参考文献：

[1] RABINER L R, JUANG B H. Fundamentals of Speech Recognition [M]. Upper Saddle River: Prentice-Hall, 1993.

[2] NATALIJA V, Howard C C. Vector quantization of images using modified adaptive resonance algorithm for hierarchical clustering [J]. IEEE Transactions on Neural Networks, 2001, 12(5): 1147-1162.

[3] OH S J, KIM J Y. A hierarchical clustering algorithm for categorical sequence data[J]. Information Processing Letters, 2004, 91: 135-140.

[4] CHEN Ning, CHEN An, ZHOU Long-xiang. An effective clustering algorithm in large transaction databases[J]. Journal of Software, 2001, 12(4): 475-484.

[5] ANDERBERG M R. Cluster Analysis for Application [M]. New York: Academic Press, 1973.

[6] DUDA R O, HART P E. Pattern Classification and Scene Analysis[M]. New York: Wiley, 1973.

[7] JAIN A K, DUBES R C. Algorithms for Clustering Data[M]. Englewood Cliffs: Prentice-Hall, 1988.

[8] HARTUV E, SHAMIR R. A clustering algorithm based on graph connectivity[J]. Information Processing Letters, 2000, 76: 175-181.

[9] BANDYOPADHYAY S. An automatic shape independent clustering technique[J]. Pattern Recognition, 2004, 37: 34-45.

[10] DASH M, LIU H, SCHEUERMANN P, et al. Fast hierarchical clustering and its validation[J]. Data & Knowledge Engineering, 2003, 44: 109-138.

[11] TALAVERA L, Béjar J. Generality-based conceptual clustering with probabilistic concepts [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(2): 196-206.

[12] CASTRO R M, COATES M J, ROBERT D N. Likelihood based hierarchical clustering[J]. IEEE Transactions on Signal Processing, 2004, 52(8): 2308-2321.

[13] XU Y, OLMAN V, XU D. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees[J]. Bioinformatics, 2002, 18(4): 536-545.

[14] Ronald R Y. Intelligent control of the hierarchical agglomerative clustering process[J]. IEEE Transactions on Systems, Man, Cybernetics—PART B: Cybernetics, 2000, 30(6): 835-845.

[15] AHO A V, HOPRCROFT J E, ULLMAN J D. The Design and Analysis of Computer Algorithms[M]. Reading MA: Addson-Wesley, 1974.

Hierarchical Subtrees Agglomerative Clustering Algorithms

LI Yu-jian

(Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, College of Computer Science and Technology, Beijing University of Technology, Beijing 100022, China)

Abstract: In order to solve the problem that Traditional Hierarchical Agglomerative Clustering Algorithms (HACA) may produce a nonunique binary tree as the clustering result of a same dataset, this paper presents Hierarchical Subtrees Agglomerative Clustering Algorithm (HSACA), the basic idea of which is to find maximal θ-distant subtrees in a minimal spanning tree of the data set and merge its vertex set. HSACA can merge many objects into a cluster in each step, and its clustering result is usually a multiple tree. This paper proves in theory that the multiple tree generated by HSACA is unique for a dataset without considering the branchy orders, and shows in computer simulations that the multiple tree describes a more reasonable clustering result than the binary tree generated by traditional HACA if there are many equidistant pairs of points in the data set.

Key words: hierarchical agglomerative clustering algorithm; hierarchical subtrees agglomerative clustering algorithm; minimal spanning tree; maximal θ-distant subtree; cluster