ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.19, No.3, March 2008, pp.500−510
DOI: 10.3724/SP.J.1001.2008.00500

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 两元指纹向量聚类问题的复杂性与改进启发式算法*

刘培强[1,2+], 朱大铭[2], 谢青松[1], 范 辉[1], 马绍汉[2]

[1](山东工商学院 信息与电子工程学院,山东 烟台 264005)

[2](山东大学 计算机科学技术学院,山东 济南 250061)

# Complexity and Improved Heuristic Algorithms for Binary Fingerprints Clustering

LIU Pei-Qiang[1,2+], ZHU Da-Ming[2], XIE Qing-Song[1], FAN Hui[1], MA Shao-Han[2]

[1](School of Information and Electronic Engineering, Shandong Institute of Business and Technology, Yantai 264005, China)

[2](School of Computer Science and Technology, Shandong University, Ji'nan 250061, China)

+ Corresponding author: Phn: +86-535-6905395, E-mail: liupeiqiang@gmail.com

**Liu PQ, Zhu DM, Xie QS, Fan H, Ma SH. Complexity and improved heuristic algorithms for binary fingerprints clustering. *Journal of Software*, 2008,19(3):500−510.** http://www.jos.org.cn/1000-9825/19/500.htm

**Abstract**: This paper proves the binary fingerprints clustering problem for 2 missing values per fingerprint is NP-Hard, and improves the Figueroa's heuristic algorithm. The new algorithm improves the implementation method for the original algorithm. Firstly, the linked list is used to store the sets of compatible vertices. The linked list can be produced by scanning the fingerprint vectors bit by bit. Thus the time complexity for producing the sets of compatible vertices is reduced from $O(m\cdot n\cdot 2^p)$ to $O(m\cdot(n-p+1)\cdot 2^p)$, and the the running time of finding a unique maximal clique or a maximal clique is improved from $O(m\cdot p\cdot 2^p)$ to $O(m\cdot 2^p)$. The real testing displays that the improved algorithm takes 49% or lower space complexity of the original algorithm on the average for the computation of the same instance. It can use 20% time of the original algorithm for solving the same instance. Particularly, the new algorithm can almost always use not more than 11% time of the original algorithm to solve the instance with more than 6 missing values per fingerprint.

**Key words**: algorithm; complexity; fingerprint clustering; gene expression data; clique partition

**摘 要**: 证明丢失值位数不超过 2 的指纹向量聚类问题为 NP-Hard,并给出 Figueroa 等人指纹向量聚类启发式算法的改进算法.主要改进了算法的实现方法.以链表存储相容顶点集合,并以逐位扫描指纹向量的方法产生相容点集链表,可将产生相容点集的时间复杂性由 $O(m\cdot n\cdot 2^p)$减小为 $O(m\cdot(n-p+1)\cdot 2^p)$,可使划分一个唯一极大团或最大团的时间复杂性由 $O(m\cdot p\cdot 2^p)$减小为 $O(m\cdot 2^p)$.实际测试显示,改进算法的空间复杂性平均减少为原算法的 49%以下,平均可用原算法 20%的时间求解与原算法相同的实例.当丢失值位数超过 6 时,改进算法几乎总可用不超过原算法 11%的时间计算与原算法相同的实例.

**关键词**: 算法;复杂性;指纹向量聚类;基因表达谱;团划分

**中图法分类号**: TP301 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60573024, 60673153 (国家自然科学基金); the Natural Science Foundation of Shandong Province of China under Grant No.Z2004G03 (山东省自然科学基金)

Received 2006-05-18; Accepted 2007-01-23

指纹向量聚类源自基因表达谱聚类分析.利用生物探针与一组DNA序列杂交,获得的杂交信息表示为实数阵列,称为基因表达谱.由基因表达谱的聚类计算求得分类结果.目前所见基因表达谱聚类方法可以归纳为:分层法[1−4]、$K$ 中间值法[5]、贪心算法[6,7]、图划分法[8−11]、概率法[12−15]和自组织映射方法[16,17],该领域研究综述可参考文献[18].近年来,Figueroa等人将基因表达谱转换为0-1-$N$向量集,称为指纹向量集,将基因表达谱聚类归为指纹向量中丢失值的计算问题[19,20].该方法解决了以前方法总存在的部分数值不能确定的困难.基因表达谱聚类分析广泛应用于不同目的的分子生物学实验分析,在疾病诊断中具有重要价值.

指纹向量聚类又称为带有丢失值的二进制数聚类问题,简记为 BCMV(binary clustering with missing values).每个指纹向量中所含丢失值个数对BCMV问题的计算复杂性影响显著.设 $p$ 表示一个BCMV实例中每个指纹向量丢失值个数的最大值,Figuroa等人证明 $p\le3$ 的BCMV问题属于NP-Hard,并针对 $p\le1$ 设计出BCMV问题的多项式时间精确算法.其启发式算法用于一般 BCMV 问题求解,在实践中证明是有效的[19].当 $p\le2$ 时,BCMV问题的复杂性仍是未知的.

本文首先证明 $p\le2$ 的 BCMV 问题属于 NP-Hard,然后改进文献[19]的启发算法.原算法依赖指纹向量集的关联图,通过求解关联图的团划分确定指纹向量集分类,从而得到丢失位的取值.改进算法采用链表取代原算法中的散列表存储相容点集,利用逐位扫描法生成链表,使生成链表的时间复杂性由 $O(m\cdot n\cdot2^p)$减小为 $O(m\cdot(n-p+1)\cdot2^p)$,使生成的相容点集数平均减小为原算法的 49%以下.程序实验表明,本文的改进算法平均可用原算法 20%的时间求解与原算法相同的实例.当丢失值位数超过 6 时,改进算法总可用不超过原算法 11%的时间求解与原算法相同的实例.

## 1 问题简介

一个指纹向量实际是生物探针与一个 DNA 序列杂交结果的解释信息,表示为 0-1-$N$ 向量.设 $X=x_1,x_2,\ldots,x_n$ 是一个指纹向量,$x_i\in\{0,1\}$表示被杂交的 DNA 序列第 $i$ 位取值已定,$x_i=N$ 则表示该位尚未确定取值,称为丢失信息位,$1\le i\le n$.设 $X=x_1,x_2,\ldots,x_n$,$Y=y_1,y_2,\ldots,y_n$ 为两个指纹向量,若 $x_i=0$,$y_i=1$ 或 $x_i=1$,$y_i=0$,则称 $x_i$ 与 $y_i$ 不相容,否则称 $x_i$ 与 $y_i$ 相容.若 $X$ 与 $Y$ 的第 1,2,…,$n$ 位均相容,则称 $X$ 与 $Y$ 相容.若一个指纹向量集合中的向量两两相容,则称该集合为相容指纹向量集,简称相容集.给定指纹向量集合 $F=\{X_1,X_2,\ldots,X_m\}$,指纹向量聚类问题欲计算 $F$ 最少可划分为多少个相容集.

给定指纹向量集 $F=\{X_1,X_2,\ldots,X_m\}$.设 $X_i=x_i[1],x_i[2],\ldots,x_i[n]$,$X_i$ 中含有丢失位数,记为 $p(X_i)=|\{x_i[j]=N|1\le j\le n\}|$,并设 $p=\max\{p(X_i)|1\le i\le m\}$,因此,又将确定参数 $p$ 的指纹向量聚类问题记为 BCMV($p$),含义为每个指纹向量所含有丢失位数均不超过 $p$.Figueroa 等人将顶点覆盖问题归约到指纹向量聚类,证明 BCMV(3)为 NP-Hard,并设计出 BCMV(1)的多项式时间精确算法.BCMV(2)的复杂性则一直未能确定.本文关于 BCMV(2)为 NP-Hard 证明,需利用另外两个 NP-Hard 问题:三元素严格覆盖和团划分问题.三元素严格覆盖问题简记为 X3C(exact cover by 3-sets),判定形式为:

实例:有限集合 $S$,$|S|=3t$,$S$ 的三元素子集的集合 $C=\{c_1,c_2,\ldots,c_n\}$.

询问:$C$ 中是否包含 $S$ 的严格覆盖.即是否存在 $C'\subseteq C$,$|C'|=t$,使得 $S$ 中每个元素都出现在 $C'$成员中.

该问题最先由 Karp 证明为 NP-Hard[21],Gary 与 Johnson 进一步证明,每个 $S$ 中的元素在 $C$ 中出现不超过 3 次的 X3C 亦为 NP-Hard[22].下面将这一元素出现不超过 3 次的 X3C 子问题简记为 X3C(3),其实例应满足:任意 $s\in S$,$|\{c_k|c_k\in C,s\in c_k\}|\le3$.

团划分问题简记为 Clique-Partition,其判定形式为:

实例:无向简单图 $G=(V,E)$,正整数 $J$.

询问:是否存在 $G$ 的顶点划分:$V=V_1\cup V_2\cup\ldots\cup V_j$,使 $j\le J$,且 $V_1,\ldots,V_j$ 均导出 $G$ 的完全子图.

$G$ 的一个完全子图称为 $G$ 的一个团.团划分问题最先由 Karp 证明为 NP-Hard[21].

## 2 BCMV(2)的复杂性

丢失位数不超过 2 的指纹向量聚类问题 BCMV(2)为:

实例:指纹向量集合 $F=\{X_1,X_2,\ldots,X_m\}$,$X_i=x_i[1],x_i[2],\ldots,x_i[n]$,$1\le i\le m$,$|\{x_i[k]=N|1\le k\le n\}|\le 2$,正整数 $K\in \mathbf{Z}^+$.

询问:是否存在 $F$ 的划分 $F=F_1\cup\ldots\cup F_k$,使 $F_1,\ldots,F_k$ 均为相容集,且 $k\le K$.

设 $F=\{X_1,X_2,\ldots,X_n\}$是 BCMV($p$)实例的指纹向量集合,$F$ 的关联图定义为以 $X_i$ 为顶点的无向简单图:$G(F)=(V,E)$,其中,$V=\{X_1,X_2,\ldots,X_n\}$,$E=\{(X_i,X_j)|X_i,X_j$ 相容$\}$.这里,$F$ 中的指纹向量与 $G(F)$中的顶点采用了相同的标记.指纹向量集合聚类与其关联图团划分存在如下关系:

**引理 1**. 若指纹向量聚类实例 $F$ 可划分为 $K>0$ 个相容集,当且仅当 $G(F)$可划分为 $K$ 个团.

下面关于无向图团划分的结论是我们证明 BCMV(2)属于 NP-Hard 的基础.

**引理 2**. 图 1 的 9 点图最少可划分为 4 个团,且有唯一解.图 2 的 6 点图最少可划分为 3 个团,且有唯一解.删除图 1 顶点{7,8,9}中任意 1 个或两个顶点得到的图最少划分为 4 个团.

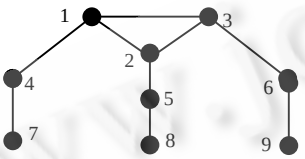


Fig.1 $T_1$-Graph

图 1 $T_1$-图

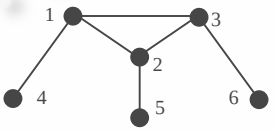


Fig.2 $T_2$-Graph

图 2 $T_2$-图

证明:图 1 的最小团划分为⟨1,2,3⟩,⟨4,7⟩,⟨5,8⟩,⟨6,9⟩,是唯一的.图 2 的最小团划分为⟨1,4⟩,⟨2,5⟩,⟨3,6⟩,是唯一的.删除图 1 顶点 7,得到图 3,最少只能划分为 4 个团.删除图 1 点集{7,8}得到图 4,最少也只能划分为 4 个团.同理,删除图 1 中点{8},{9},{7,9},{8,9}时,得到的图均只能划分为 4 个团. □

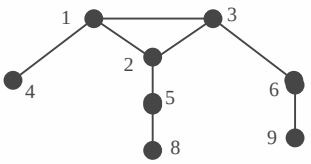


Fig.3 $T_3$-Graph

图 3 $T_3$-图

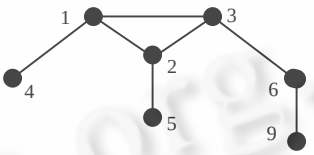


Fig.4 $T_4$-Graph

图 4 $T_4$-图

以下将与图 1、图 2、图 3、图 4 同构的无向图分别称为 $T_1$-图、$T_2$-图、$T_3$-图和 $T_4$-图.

**定理 1**. BCMV(2)是 NP-Hard 问题.

证明:将 X3C(3)归约到 BCMV(2).设 X3C(3)的实例为 $S=\{s_1,s_2,\ldots,s_{3t}\}$,$C=\{c_1,c_2,\ldots,c_m\}$.$c_i=\{s_{x[i]},s_{y[i]},s_{z[i]}\}\subseteq S$.证明步骤是:先构造一个无向简单图 $G=(V,E)$并确定正整数 $J$,使得由图 $G$ 可划分 $J$ 个团,当且仅当 X3C(3)实例存在严格覆盖;然后构造 BCMV(2)实例,恰以 $G$ 为其关联图.从而由引理 1 保证归约正确. □

用一个 $T_1$-子图 $G(c_i)=(V_i,E_i)$代替 $c_i$,其点集记为 $V_i=\{c_{i1},c_{i2},c_{i3},c_{i4},c_{i5},c_{i6},s_{x[i]},s_{y[i]},s_{z[i]}\}$,边集为 $E_i=\{(c_{i1},c_{i2}),(c_{i2},c_{i3}),(c_{i3},c_{i1})\}\cup\{(c_{i1},c_{i4}),(c_{i2},c_{i5}),(c_{i3},c_{i6})\}\cup\{(c_{i4},s_{x[i]}),(c_{i5},s_{y[i]}),(c_{i6},s_{z[i]})\}$,$G(c_i)$如图 5 所示.其中,$s_{x[i]},s_{y[i]},s_{z[i]}$既表示 $c_i$ 的元素,又表示 $G(c_i)$中的顶点.

团划分实例 $G=(V,E)$可看作由 $G(c_1),\ldots,G(c_m)$合并而成,共含有 $3t+6m$ 个点,其中,$3t$ 个点对应 X3C(3)实例 $S$ 中的元素,直接用 $S$ 表示;另外 $6m$ 个点对应 $C$ 中的 3 元素子集.即

$$V(G)=\bigcup_{i=1}^{m}V_i \tag{1}$$

$$V_i=\{c_{i1},c_{i2},c_{i3},c_{i4},c_{i5},c_{i6},s_{x[i]},s_{y[i]},s_{z[i]}\} \tag{2}$$

$$E(G)=\bigcup_{i=1}^{m}E_i \tag{3}$$

$$E_i=\{(c_{i1},c_{i2}),(c_{i2},c_{i3}),(c_{i3},c_{i1})\}\cup\{(c_{i1},c_{i4}),(c_{i2},c_{i5}),(c_{i3},c_{i6})\}\cup\{(c_{i4},s_{x[i]}),(c_{i5},s_{y[i]}),(c_{i6},s_{z[i]})\} \tag{4}$$

由点集$\{c_{i1},c_{i2},c_{i3},c_{i4},c_{i5},c_{i6}\}$支撑的子图是 $T_2$-图.例如,若 $S=\{s_1,s_2,s_3,s_4,s_5,s_6\}$,$C=\{c_1,c_2,c_3,c_4\}$,$c_1=\{s_1,s_2,s_3\}$,$c_2=\{s_4,s_5,s_6\}$,$c_3=\{s_1,s_3,s_5\}$,$c_4=\{s_2,s_4,s_6\}$,则由该实例所构造的团划分实例如图 6 所示.

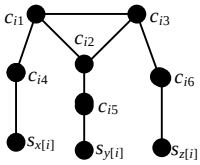


Fig.5 The sub-graph corresponding to $c_i=\{s_{x[i]},s_{y[i]},s_{z[i]}\}$

图 5 对应 $c_i=\{s_{x[i]},s_{y[i]},s_{z[i]}\}$的子图

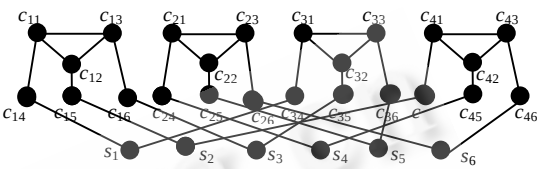


Fig.6 The graph $G$ constructed from X3C(3)

图 6 由 X3C(3)实例构造图 $G$

**性质 1**. X3C(3)实例存在严格覆盖,当且仅当图 $G$ 可划分成 $t+3m$ 个团.

证明:($\rightarrow$)设 X3C(3)实例存在 $C'\subseteq C$,严格覆盖 $S$.显然,$|C'|=t$,则按照如下方法划分 $G$:

(1) 对于每个 $c_i\in C'$,将 $G(c_i)$的点集划分为$\{c_{i1},c_{i2},c_{i3}\},\{c_{i4},s_{x[i]}\},\{c_{i5},s_{y[i]}\},\{c_{i6},s_{z[i]}\}$,由关于 $E_i$ 的表达式(4)可知,每个点集均为团.因若 $c_i\neq c_j,c_i,c_j\in C'$,则 $G(c_i)$与 $G(c_j)$不共享任意图 $G$ 的顶点,因此可得到 $4t$ 个团,其中 $t$ 个三角形、$3t$ 条边;(2) 在 $G$ 中删除由(1)划分的点集,则 $S$ 中的点全部删除,剩余 $G$ 的子图必由 $m-t$ 个互不联通的 $T_2$-子图组成.由引理 2,每个 $T_2$-子图均能划分出 3 个团,还能划分出 $3(m-t)$个团.总团数为 $4t+3(m-t)=t+3m$.

($\leftarrow$)设存在 $G$ 的划分,将 $G$ 划分为 $t+3m$ 个团.任意图 $G$ 的团划分总可看作将图 $G$ 先划分为 $T_1$-子图、$T_2$-子图、$T_3$-子图、$T_4$-子图,再将这些子图分别划分为团.

因$|S|=3t$,$G$ 最多含有 $t$ 个互不共享顶点的 $T_1$-子图.先证明 $G$ 中必含有 $t$ 个两两互不共享顶点的 $T_1$-子图.用反证法,假设 $G$ 的团划分先将 $G$ 划分为 $t_1$ 个 $T_1$-子图、$t_2$ 个 $T_2$-子图、$t_3$ 个 $T_3$-子图、$t_4$ 个 $T_4$-子图,且 $t_1<t,t_1+t_3+t_4>t$.由引理 2,由所有 $T_1$-子图、$T_3$-子图、$T_4$-子图最少可划分为 $4(t_1+t_3+t_4)$个团,由所有 $T_2$-子图最少可划分为 $3t_2$ 个团.故 $G$ 最少可划分的团数为 $4(t_1+t_3+t_4)+3(m-(t_1+t_3+t_4))=t_1+t_3+t_4+3m>t+3m$,这与假设图 $G$ 可划分为 $t+3m$ 个团相矛盾,因此,$G$ 必含有 $t$ 个互不共享顶点的 $T_1$-子图.不失一般性,可设 $G(c_1),\ldots,G(c_t)$为 $G$ 所含有的 $T_1$-子图,则它们对应的 X3C(3)实例中的三元素子集 $c_1,c_2,\ldots,c_t$即为 $S$ 的严格覆盖,$c_1\cup c_2\cup\ldots\cup c_t=S$. □

因为 X3C(3)实例每个元素最多包含在 3 个 $S$ 的三元素子集中,故任意图 $G$ 的顶点 $s\in S$ 最多只邻接 3 条边.下面构造以图 $G$ 为关联图的 BCMV(2)实例.不失一般性,假设图 $G$ 是连通的.

构造 $3t+6m$ 个指纹向量,每个指纹向量均由 $12t$ 位组成.在下面的叙述中,总以 4 位一组表示指纹向量.欲构造的指纹向量分为两类:$F=F(S)\cup F(C)$.

(1) $S$ 类:对应 $S$ 中每个元素 $s_i$,构造一个指纹向量,由大写字母 $S_i$ 表示,即 $F(S)=\{S_1,\ldots,S_{3t}\}$.其中,

$$S_i=x_i[1],x_i[2],x_i[3],x_i[4],\ldots,x_i[12t],1\le i\le 3t \tag{5}$$

将序列 $S_i$ 每 4 位分为一组,并记 $S_i[j]=x_i[4j-3],x_i[4j-2],x_i[4j-1],x_i[4j]$,则,

$$S_i[j]=x_i[4j-3],x_i[4j-2],x_i[4j-1],x_i[4j]=\begin{cases}1,1,N,N, & \text{if } j=i\\ 0,0,0,0, & \text{otherwise}\end{cases},1\le j\le 3t \tag{6}$$

显然,$F(S)$中的任意两个指纹向量 $S_i,S_j$ 必不相容,$i\neq j$.

(2) $C$ 类:若 $C$ 中有$\Delta\in\{1,2,3\}$个三元素子集包含 $s\in S$,则称该元素 $s$ 在 $C$ 中出现$\Delta$次.由 $G$ 的连通性假设,每个集合 $c_i$ 至少包含 1 个在 $C$ 中出现大于 1 次的元素.于是,可将 $C$ 中的三元素子集排序为 $c_1,c_2,\ldots,c_m$,满足 $c_i$ 至少包含 1 个元素 $s$,$s$ 出现在$\{c_1,\ldots,c_{i-1}\}$中,$2\le i\le m$.将 $c_1,\ldots,c_m$ 排序后,若元素 $s\in c_i,s\in c_j,s\in c_k,i<j<k$,则称 $s$ 在 $c_i$ 中第 1 次出现,记 $t(s,c_i)=1$;在 $c_j$ 中第 2 次出现,记 $t(s,c_j)=2$;在 $c_k$ 中第 3 次出现,记 $t(s,c_k)=3$.在 X3C(3)实例中,任意元素最多出现 3 次.

设 X3C(3)实例中的三元素子集 $c_i=\{s_{x[i]},s_{y[i]},s_{z[i]}\}$满足 $t(s_{x[i]},c_i)\ge t(s_{y[i]},c_i)\ge t(s_{z[i]},c_i)$.构造 6 个指纹向量:

$F(c_i)=\{C_{i1},C_{i2},C_{i3},C_{i4},C_{i5},C_{i6}\}$,分别对应图 $G$ 的 6 个顶点$\{c_{i1},c_{i2},c_{i3},c_{i4},c_{i5},c_{i6}\}$,其中,指纹向量采用大写字母表示,为的是与图的顶点相区分.每个指纹向量均以 4 位一组,共 $3t$ 组.设

$$C_{ik}=x_{ik}[1],x_{ik}[2],x_{ik}[3],x_{ik}[4],\ldots,x_{ik}[12t-3],x_{ik}[12t-2],x_{ik}[12t-1],x_{ik}[12t],1\le k\le 6.$$

为方便起见,记 $C_{ik}[j]=x_{ik}[4j-3],x_{ik}[4j-2],x_{ik}[4j-1],x_{ik}[4j]$,即 $C_{ik}=C_{ik}[1],\ldots,C_{ik}[3t].C_{ik}$ 赋值如下.首先,

$$C_{ik}[j]=0,0,0,0,j\notin\{x[i],y[i],z[i]\},1\le j\le 3t,1\le k\le 6 \quad (7)$$

下面确定 $C_{ik}[x[i]],C_{ik}[y[i]],C_{ik}[z[i]],1\le k\le 6.C_{i1}[x[i]],C_{i4}[x[i]],C_{i2}[y[i]],C_{i5}[y[i]],C_{i3}[z[i]],C_{i6}[z[i]]$按照表 1 确定.

**Table 1** Confirmation of partial values of $C_{i1}[x[i]],C_{i4}[x[i]],C_{i2}[y[i]],C_{i5}[y[i]],C_{i3}[z[i]],C_{i6}[z[i]]$

**表 1** 确定 $C_{i1}[x[i]],C_{i4}[x[i]],C_{i2}[y[i]],C_{i5}[y[i]],C_{i3}[z[i]],C_{i6}[z[i]]$部分值

| | $t(s_{x[i]},c_i)=1$ | $t(s_{x[i]},c_i)=2$ | $t(s_{x[i]},c_i)=3$ |
|---|---|---|---|
| $c_{i4}[x[i]]$ | 1,*N*,0,0 | 1,*N*,0,1 | 1,*N*,1,0 |
| $c_{i1}[x[i]]$ | *N*,0,0,0 | *N*,0,0,1 | *N*,0,1,0 |
| | $t(s_{y[i]},c_i)=1$ | $t(s_{y[i]},c_i)=2$ | $t(s_{y[i]},c_i)=3$ |
| $c_{i5}[y[i]]$ | 1,*N*,0,0 | 1,*N*,0,1 | 1,*N*,1,0 |
| $c_{i2}[y[i]]$ | *N*,0,0,0 | *N*,0,0,1 | *N*,0,1,0 |
| | $t(s_{z[i]},c_i)=1$ | $t(s_{z[i]},c_i)=2$ | $t(s_{z[i]},c_i)=3$ |
| $c_{i6}[z[i]]$ | 1,*N*,0,0 | 1,*N*,0,1 | 1,*N*,1,0 |
| $c_{i3}[z[i]]$ | *N*,0,0,0 | *N*,0,0,1 | *N*,0,1,0 |

$F(c_i)$ 中指纹向量的另外两组数据 $C_{i4}[y[i]],C_{i1}[y[i]],C_{i4}[z[i]],C_{i1}[z[i]],C_{i5}[x[i]],C_{i2}[x[i]],C_{i5}[z[i]],C_{i2}[z[i]],C_{i6}[x[i]],C_{i3}[x[i]],C_{i6}[y[i]],C_{i3}[y[i]]$按照表 2 确定.

**Table 2** Confirmation of the last two groups of set or 8-bit data of each vector in $F(c_i)$

**表 2** $F(c_i)$中每个向量最后两组 8 位数据的确定

| | $t(s_{y[i]},c_i)=1$ | $t(s_{y[i]},c_i)=2$ | $t(s_{y[i]},c_i)=3$ | | $t(s_{z[i]},c_i)=1$ | $t(s_{z[i]},c_i)=2$ | $t(s_{z[i]},c_i)=3$ |
|---|---|---|---|---|---|---|---|
| $C_{i4}[y[i]]$ | 0,0,0,0 | 0,0,0,*N* | 0,0,*N*,0 | $C_{i4}[z[i]]$ | 0,0,0,0 | 0,0,0,0 | 0,0,0,0 |
| $C_{i1}[y[i]]$ | 0,0,0,0 | 0,0,0,1 | 0,0,1,0 | $C_{i1}[z[i]]$ | 0,0,0,0 | 0,0,0,*N* | 0,0,*N*,0 |
| | $t(s_{x[i]},c_i)=1$ | $t(s_{x[i]},c_i)=2$ | $t(s_{x[i]},c_i)=3$ | | $t(s_{z[i]},c_i)=1$ | $t(s_{z[i]},c_i)=2$ | $t(s_{z[i]},c_i)=3$ |
| $C_{i5}[x[i]]$ | 0,0,0,0 | 0,0,0,*N* | 0,0,*N*,0 | $C_{i5}[z[i]]$ | 0,0,0,0 | 0,0,0,0 | 0,0,0,0 |
| $C_{i2}[x[i]]$ | 0,0,0,0 | 0,0,0,1 | 0,0,1,0 | $C_{i2}[z[i]]$ | 0,0,0,0 | 0,0,0,*N* | 0,0,*N*,0 |
| | $t(s_{x[i]},c_i)=1$ | $t(s_{x[i]},c_i)=2$ | $t(s_{x[i]},c_i)=3$ | | $t(s_{y[i]},c_i)=1$ | $t(s_{y[i]},c_i)=2$ | $t(s_{y[i]},c_i)=3$ |
| $C_{i6}[x[i]]$ | 0,0,0,0 | 0,0,0,*N* | 0,0,*N*,0 | $C_{i6}[y[i]]$ | 0,0,0,0 | 0,0,0,0 | 0,0,0,0 |
| $C_{i3}[x[i]]$ | 0,0,0,0 | 0,0,0,1 | 0,0,1,0 | $C_{i3}[y[i]]$ | 0,0,0,0 | 0,0,0,*N* | 0,0,*N*,0 |

于是 $C$ 类指纹向量集为

$$F(C)=\bigcup_{i=1}^{m}F(C_i) \quad (8)$$

因为$|S|=3t,|C|=m$,所以将 $C$ 中元素排序显然可在 $t$ 与 $m$ 的多项式时间内完成.所构造指纹向量聚类实例含有 $3t+6m$ 个指纹向量,每个指纹向量最多含有 2 个丢失位.显然,构造所有指纹向量也可在 $t$ 与 $m$ 的多项式时间内完成.下面证明由此构造的 BCMV(2)实例的关联图 $G(F)$与图 $G$ 同构.

**性质 2**. $F(c_i)$的关联图是 $T_2$-图,$T_i=\{S_{x[i]},S_{y[i]},S_{z[i]}\}\cup F(c_i)$的关联图是 $T_1$-图.

证明:易知 $S_{x[i]},S_{y[i]},S_{z[i]}$两两不相容.根据 $s_{x[i]},s_{y[i]},s_{z[i]}$在 $c_i$ 中第 1~3 次出现的 $C_{i4}[x[i]],C_{i4}[y[i]],C_{i4}[z[i]]$赋值,分别比较 $S_{x[i]}[x[i]]$与 $C_{i4}[x[i]]$、$S_{x[i]}[y[i]]$与 $C_{i4}[y[i]]$、$S_{x[i]}[z[i]]$与 $C_{i4}[z[i]]$,可知,$S_{x[i]}$与 $C_{i4}$ 总相容.同理可得 $S_{y[i]}$与 $C_{i5}$ 相容、$S_{z[i]}$与 $C_{i6}$ 相容.同样比较可证:$C_{i1}$ 与 $C_{i4}$ 相容、$C_{i2}$ 与 $C_{i5}$ 相容、$C_{i3}$ 与 $C_{i6}$ 相容,且 $C_{i1},C_{i2},C_{i3}$ 两两相容.因 $C_{i1}[y[i]]=0,x,x,x,C_{i5}[y[i]]=1,x,x,x,x\in\{0,1,N\}$,故 $C_{i1},C_{i5}$ 不相容.同理,$C_{i1}$ 与 $C_{i6}$、$C_{i2}$ 与 $C_{i4}$、$C_{i2}$ 与 $C_{i6}$、$C_{i3}$ 与 $C_{i4}$、$C_{i3}$ 与 $C_{i5}$,均不相容,$C_{i4},C_{i5},C_{i6}$ 两两不相容.又因 $C_{i1}[x[i]]=x,0,x,x,S_{x[i]}[x[i]]=1,1,N,N$,故 $C_{i1}$ 与 $S_{x[i]}$不相容.同理可得,分别取自$\{C_{i1},C_{i2},C_{i3}\}$和$\{S_{x[i]},S_{y[i]},S_{z[i]}\}$的任意两个指纹向量不相容.类似地还有,$C_{i4}$ 与 $S_{y[i]},S_{z[i]}$、$C_{i5}$ 与 $S_{x[i]},S_{z[i]}$、$C_{i6}$ 与 $S_{x[i]},S_{y[i]}$各不相容.性质 2 得证. □

**性质 3**. 若 $i\ne j$,则分别取自 $F(c_i)$与 $F(c_j)$的任意两个指纹向量不相容,$1\le i,j\le m$.

证明:设 $j>i,t(s_{x[i]},c_i)\ge t(s_{y[i]},c_i)\ge t(s_{z[i]},c_i),t(s_{x[j]},c_j)\ge t(s_{y[j]},c_j)\ge t(s_{z[j]},c_j)$.由 $G$ 的联通性假设有:$t(s_{x[j]},c_j)>1$.

(1) 若 $s_{x[j]}\notin\{s_{x[i]},s_{y[i]},s_{z[i]}\}$,则 $C_{ik}[x[j]]=0,0,0,0,1\le k\le 6$,而 $C_{j1}[x[j]],C_{j2}[x[j]],C_{j3}[x[j]],C_{j4}[x[j]]$均取值 $x,x,0,1$ 或 $x,x,1,0$.因此,$C_{j1},C_{j2},C_{j3},C_{j4}$ 均与 $C_{ik}$ 不相容,$1\le k\le 6$.

再证明 $C_{j5},C_{j6}$ 与 $C_{ik}(1\le k\le 6)$不相容.若 $s_{y[j]}\notin\{s_{x[i]},s_{y[i]},s_{z[i]}\}$,则 $C_{ik}[y[j]]=0,0,0,0,1\le k\le 6,C_{j5}[y[j]]=1,N,x,x$;若 $s_{y[j]}\in\{s_{x[i]},s_{y[i]},s_{z[i]}\},t(s_{y[j]},c_j)=2$, 则 $t(s_{y[j]},c_i)=1,C_{ik}[y[j]]=x,x,0,0,C_{j5}[y[j]]=1,N,0,1$; 若 $s_{y[j]}\in\{s_{x[i]},s_{y[i]},s_{z[i]}\},t(s_{y[j]},c_j)=3$, 则 $t(s_{y[j]},c_i)\in\{1,2\},C_{ik}[y[j]]$取值为 $x,x,0,x,C_{j5}[y[j]]=1,N,1,0$,故 $C_{j5}$ 与 $C_{ik}$ 不相容,同理可知,$C_{j6}$ 与 $C_{ik}$ 不相容.

(2) 若 $s_{x[j]}\in\{s_{x[i]},s_{y[i]},s_{z[i]}\},t(s_{x[j]},c_j)=2$,则 $t(s_{x[j]},c_i)=1,C_{ik}[x[j]]=x,x,0,0,1\le k\le 6,C_{jl}[x[j]]=x,x,0,1,1\le l\le 4$.若 $s_{x[j]}\in\{s_{x[i]},s_{y[i]},s_{z[i]}\},t(s_{x[j]},c_j)=3$,则 $t(s_{x[j]},c_i)\in\{1,2\}$,则 $C_{ik}[x[j]]=x,x,0,x,1\le k\le 6,C_{jl}[x[j]]=x,x,1,0,1\le l\le 4$.所以,$C_{jl}\in\{C_{j1},C_{j2},C_{j3},C_{j4}\}$ 与 $C_{ik}\in\{C_{i1},C_{i2},C_{i3},C_{i4},C_{i5},C_{i6}\}$不相容.类似(1)的讨论可证,$C_{jl}\in\{C_{j5},C_{j6}\}$与 $C_{ik}\in\{C_{i1},C_{i2},C_{i3},C_{i4},C_{i5},C_{i6}\}$不相容. □

**性质 4**. 若 $j\notin\{x[i],y[i],z[i]\}$,则 $S_j$ 与 $C_{ik}\in\{C_{i1},C_{i2},C_{i3},C_{i4},C_{i5},C_{i6}\}$不相容.

证明:$S_j[j]=1,1,N,N,$而 $C_{ik}[j]=0,0,0,0$.故有性质 4. □

设 $T_i=\{S_{x[i]},S_{y[i]},S_{z[i]}\}\cup F(c_i)$.由性质 2,$G(T_i)$是图 $G$ 的 $T_1$-子图;由性质 3、性质 4,$G(T_i)$与 $G(T_j)$共享顶点的充要条件是 $c_i\cap c_j\ne\varnothing$,且 $G(T_i)$与 $G(T_j)$只可能共享 $c_i\cap c_j$ 中元素对应的顶点.故 $F=F(S)\cup F(C)$的关联图 $G(F)$恰为 $G$. 再由性质 1、引理 1 可得:若 X3C(3)实例存在 $C'\subseteq C$ 严格覆盖 $S$,当且仅当 $F$ 可划分 $t+3m$ 个相容集. □

## 3 指纹向量聚类启发式算法改进与实现

给定 BCMV($p$)实例 $F=\{X_1,\ldots,X_m\}$,每个指纹向量长度为 $n$.设 $r\in\{0,1\}^n$,若 $r$ 与 $X_i$ 相容,则称 $r$ 为 $X_i$ 的结果向量.设 $R(X_i)$为 $X_i$ 的所有结果向量集合,显然,$|R(X_i)|=2^{p(X_i)}$.记 $R(F)=\bigcap_{X_i\in F}R(X_i)$,若 $F_1\subseteq F$ 是一个相容集,则 $R(F_1)\ne\varnothing$.设 $R\subseteq\{0,1\}^n$,若 $R\subseteq\bigcup_{i=1}^m R(X_i)$,且任意 $X_i\in F$,总有 $R(X_i)\cap R\ne\varnothing$,则称 $R$ 为 $F$ 的一个结果向量集.若 $R=\{r_1,\ldots,r_t\}$为 $F$ 的结果向量集,则依次计算 $F(r_k)=\{X_i|r_k\in R(X_i)\}$,并在 $F$ 中删除 $F(r_k)$,直到 $F=\varnothing,1\le k\le t$,由此可将 $F$ 划分为不超过$|R|$个相容集.$F$ 的最优相容集划分对应于向量个数最少的结果向量集 $R$.任选 $r_i\in R(X_i),1\le i\le m$,则 $R=\bigcup_{i=1}^m\{r_i\}$ 是 $F$ 的结果向量集.由此方法最多可求得 $2^{\sum_{i=1}^m p(X_i)}$ 个结果向量集,其中必含有向量最少的 $F$ 结果向量集.因此,总可在 $O\left(2^{\sum_{i=1}^m p(X_i)}\cdot m\cdot n\right)$时间内给出任意 BCMV($p$)实例最优解,$p=\max\{p(X_i)|1\le i\le m\}$.

### 3.1 贪心团划分算法

一个图 $G$ 的最大团是该图含有顶点个数最多的 $G$ 的完全子图.若 $G$ 含有一个顶点 $v$,使 $v$ 与 $v$ 的所有相邻点形成 $G$ 的团 $C$,则顶点 $v$ 称为 $G$ 的单纯顶点,$C$ 称为 $G$ 的唯一极大团(unique maximal clique).一个唯一极大团可以含有多个单纯顶点.设 $C$ 为图 $G$ 的一个唯一极大团,则必存在 $G$ 的一个最优团划分,使 $V(C)$为该划分的一个顶点子集.

文献[19]最先给出贪心算法求解 BCMV($p$),其算法简称 GCP(greedy clique partition)算法.实践证明,GCP 算法是有效的.其计算步骤为,先在 $G(F)$中寻找唯一极大团,若 $G(F)$不含唯一极大团,则寻找最大团,把已找到的唯一极大团或最大团从 $G(F)$中删除,重复寻找和删除过程,直至 $G(F)$点集为空.一个 $G(F)$的团对应 $F$ 的一个相容集.

高效的 GCP 算法实现方法是 BCMV($p$)快速求解的关键.$G(F)$确定后,利用 $G(F)$寻找一个唯一极大团的时间复杂性是 $O(m^3)$,$m$ 为指纹向量个数.指纹向量 $X_i$ 的结果向量也称为 $G(F)$中顶点 $X_i$ 的结果向量,一个 $F$ 的相容集也称为 $G(F)$的一个相容点集.文献[19]中的 GCP 算法由下述两步实现,称为散列表法:

(1) 建立 $G(F)$的相容点集散列表.逐个计算每个指纹向量 $X_i$ 的结果向量 $r\in R(X_i),1\le i\le m$,建立散列表,记录以 $r$ 为结果向量的指纹向量编号,即 $G(F)$的顶点编号,统计以 $r$ 为结果向量的顶点个数 $c(r)$.

(2) 先由散列表计算唯一极大团,等 $G(F)$不含唯一极大团时再求最大团.重复该过程,直到将 $G(F)$的顶点划分完毕.

因 $p(X_i)\le p,1\le i\le m$,故算法最多需考虑 $O(m\cdot 2^p)$个结果向量,因此散列表中最多含有 $O(m\cdot 2^p)$个相容点集.将一个指纹向量插入散列表需要 $O(n)$时间,故建立散列表的时间复杂性为 $O(m\cdot n\cdot 2^p)$.算法从散列表中选择 $c(r)$最大的点集,可得到 $G(F)$的最大团.求得一个最大团的时间复杂性是 $O(m\cdot 2^p)$.GCP 算法利用散列表计算唯一极大团的时间复杂性是 $O(m\cdot p\cdot 2^p)$,当 $m>>p$ 且 $p$ 较小时,该时间复杂性明显小于直接由 $G(F)$求解唯一极大团的时间复

杂性($O(m^3)$).完成团划分的时间复杂性为 $O(m^2 \cdot p \cdot 2^p)$.

显然,该算法只适合 $p$ 较小的情况.当 $p$ 增大时,散列表所需存储空间增大太快,使得算法所能计算的实例规模受到限制.文献[19]给出了固定散列表大小,这是利用双散列函数将结果向量散列化的计算方法.这种方法所带来的散列值冲突也无法忽视.实验表明,即使对于一般的BCMV($p$)实例,访问一个结果向量平均发生3次冲突.少量的散列冲突对于计算速度影响不大,较多的散列冲突将使该算法的计算速度明显降低.当散列空间不足时,部分结果向量不能存储在散列表中,从而影响团划分的质量.

### 3.2 算法的改进

以固定长度的散列表存储相容点集,实际上浪费了存储空间.另外,散列函数法当遇到散列值冲突时,求解速度会明显降低.当 BCMV($p$)实例 $p$ 很小时,散列冲突不易碰到,当 $p$ 增大时,文献[19]的散列函数值冲突便成为一个不得不考虑的因素.

改进 GCP 算法考虑利用链表存储相容点集.利用链表存储相容点集的原始意图在于节省存储空间,增大 GCP 算法所能计算的实例规模.并在 $G(F)$的团划分计算中,可利用在链表上删除已解出的最大团或唯一极大团顶点,逐渐加快扫描链表的速度,并消除散列冲突.

在 GCP 算法的链表存储法实现中,若采用逐位扫描指纹向量获得结果向量集,可使建立链表的时间复杂性由 $O(m \cdot n \cdot 2^p)$减小为 $O(m \cdot (n-p+1) \cdot 2^p)$.另外,在建立相容点集链表计算中,利用下面的性质 5,对于减少存储相容点集的空间复杂性作用十分明显.

**性质 5**. 给定指纹向量集 $F=\{X_1,...,X_m\}, X_i=x_i[1],\ldots,x_i[n]$.若存在 $k, 1\le k\le n$,使得对任意 $i, 1\le i\le m$,满足 $x_i[k]\in\{0,N\}$,则对 $F$ 的任意相容集划分:$F=F_1\cup\ldots\cup F_t$,总存在 $r\in R(F_j)$,其第 $k$ 位为 0,$r[k]=0$.若 $x_i[k]\in\{1,N\}$,则对 $F$ 的任意相容集划分:$F=F_1\cup\ldots\cup F_t$,总存在 $r\in R(F_j)$,其第 $k$ 位为 1,$r[k]=1$.

我们将该性质称为指纹向量丢失位的不对称性质.改进计算方法如下:

(1) 相容点集链表的建立

逐位扫描指纹向量,计算指纹向量的结果向量,建立链表记录每个结果向量对应的 $G(F)$点子集.扫描指纹向量第 1 位:$x_i[1], 1\le i\le m$,计算 $F_1=\{i|x_i[1]=0, X_i\in F\}, F_2=\{i|x_i[1]=1, X_i\in F\}, F_3=\{i|x_i[1]=N, X_i\in F\}$,若 $F_1\neq\varnothing$,则建立链表记录相容点集 $F(0)=F_1\cup F_3$;若 $F_2\neq\varnothing$,则建立链表记录相容点集 $F(1)=F_2\cup F_3$.设扫描指纹向量前 $k$ 位获得相容点集已确定,算法扫描第 $k+1$ 位时,只需按照前述方法分解每个相容点集.设 $F(r[1],\ldots,r[k])$是一个相容点集,$r[1],\ldots,r[k]\in\{0,1\}^k$,计算 $F_1=\{i\in F(r[1],\ldots,r[k])|x_i[k+1]=0\}, F_2=\{i\in F(r[1],\ldots,r[k])|x_i[k+1]=1\}, F_3=\{i\in F(r[1],\ldots,r[k])|x_i[k+1]=N\}$,若 $F_1\neq\varnothing$,则建立链表记录相容点集 $F(r[1],\ldots,r[k],0)=F_1\cup F_3$;若 $F_2\neq\varnothing$,则建立链表记录相容点集 $F(r[1],\ldots,r[k],1)=F_2\cup F_3$.只需拆分表达相容点集 $F(r[1],\ldots,r[k])$的链表即可得到 $F(r[1],\ldots,r[k],0)$和 $F(r[1],\ldots,r[k],1)$.扫描到指纹向量的最后一位为止.注意相容点集链表只存储顶点编号,无须存储指纹向量.

例:$F=\{NN00, N1N1, 1NN0, 1N1N\}$,$G(F)$的相容点集计算过程见表 3,其最小团划分为$\{1,3\}\cup\{2,4\}$.

**Table 3** Creating of the linked list by scanning $F$

**表 3** 扫描 $F$ 构建链表

| The 1st bit | | The 2nd bit | | The 3rd bit | | The 4th bit | |
|---|---|---|---|---|---|---|---|
| Resolved vector | Compatible set | Resolved vector | Compatible set | Resolved vector | Compatible set | Resolved vector | Compatible set |
| 1*xxx* | {1,2,3,4} | 11*xx* | {1,2,3,4} | 110*x* | {1,2,3} | 1100 | {1,3} |
| | | | | | | 1101 | {2} |
| | | | | 111*x* | {2,3,4} | 1110 | {3,4} |
| | | | | | | 1111 | {2,4} |

(2) 求解 $G(F)$的团划分

一个相容点集用一个单向链表存储.其头结点存储相容点集的结果向量 $r$,所含顶点个数为 $c(r)$,$c(r)$称为关于结果向量 $r$ 的顶点计数器.后续每个结点存储一个顶点编号.链表形成后,按照唯一极大团优先策略寻找并删除唯一极大团或最大团的顶点.$c(r)$最大的链表所存储的点集形成最大团.相容点集链表如图 7 所示.

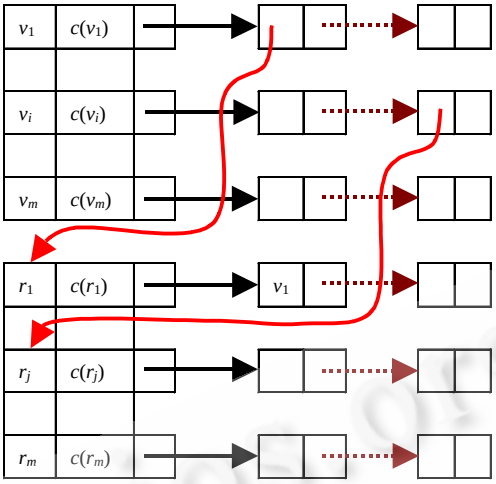


Fig.7 The compatible set linked list and the pointer linked list

图 7 相容点集链表与指针链表

为判断一个相容点集是否为唯一极大团,需记录一个顶点包含在哪些相容集中.我们对每个顶点 $v\in V(G(F))$建立一个指针链表,链表中每个指针指向一个包含 $v$ 的相容集链表的头结点,其中,指针链表的头记录该顶点含在相容点集中的次数,记为 $c(v)$,$c(v)$称为关于顶点 $v$ 的相容集计数器.在团删除时,若顶点 $v$ 被删除,则 $c(v)=0$,下面给出相容集计数器减 1 的条件:

**性质 6**. 设 $C_1,C_2$ 是包含顶点 $v$ 的 $G(F)$的两个团,若 $C$ 是不包含顶点 $v$ 的团,且$|V(C_2)-V(C)|=1,|V(C_1)-V(C)|>1$,则删除 $V(C)$后,$c(v)=c(v)-1$.

证明:若 $v$ 是单纯点,则删除 $v$ 的所有相邻点后剩下孤点团 $v$,此时,$c(v)$不能减 1,否则,删除团 $C$ 的顶点后,$C_2$ 只剩下顶点 $v$,$C_1$ 剩下其他顶点与 $v$ 形成团,$C_2$ 不再需要单独考虑. □

在相容点集链表中删除一个团,只需将表示该团的链表空间释放即可,但要在一个链表中仅删除一个顶点,则有可能扫描链表的所有顶点.为避免每次删除顶点都要扫描整个链表,我们只需在删除操作时将链表头结点中的顶点计数器减 1,$c(r)=c(r)-1$,但当 $c(r)=1$ 时,需判断该链表存储的最后一个顶点是否满足性质 6,因此,只需在最后一次扫描整个链表完成删除即可.另外,算法没有必要输出每个团包含的点,只需给出每个团对应的结果向量.这里给出求解 $G(F)$团划分的贪心算法.

团划分算法:

//设 $r_1,r_2,\ldots,r_m$ 为由指纹向量集 $F$ 得到的所有结果向量,对应相容集链表已建成.

//输出结果向量集 $R$.$R$ 开始为空.

1. While $(\exists x,c(x)>0)$ do
2. 若存在 $G$ 的顶点 $v$,$c(v)=1$,则根据 $v$ 的指针链表找到 $v$ 所在相容点集 $F(r)$;否则,选 $c(r)$.
3. 最大的相容点集 $F(r)$,$r\in\{r_1,r_2,\ldots,r_m\}$,$R=R\cup\{r\}$.
4. 从现有相容点集链表中删除 $F(r)$中的顶点,对每个 $u\in F(r)$,

4.1. 根据 $u$ 的指针链表找到含有 $u$ 的另外相容集 $F(r_t)$,$u\in F(r_t)$.

$C(r_t)=c(r_t)-1$.若 $c(r_t)=1$,扫描 $F(r_t)$删除 $c(v)=0$ 的顶点,找到唯一顶点 $w$,$c(w)>0$.

4.2. 若存在 $F(r_s)$,$w\in F(r_s)$,$c(r_s)>1$,则删除 $F(r_t)$,$c(w)=c(w)-1$.

5. 对每个 $u\in F(r)$,$c(u)=0$.
6. End While
7. 输出 $R$.

**定理 2**. 给定 BCMV($p$)实例 $F=\{X_1,\ldots,X_m\}$,$X_i=x_i[1],\ldots,x_i[n]$,则建立相容点集链表的时间复杂性为

$$O(m\cdot(n-p+1)\cdot 2^p).$$

证明:考察指纹向量 $X_i\in F$ 填充链表的过程.若 $x_i[k]=N,1\le k\le p(X_i),x_i[k']\in\{0,1\},p(X_i)+1\le k'\le n$,则 $X_i$ 在计算中被扫描次数最多.算法扫描前 $p(X_i)$位将 $X_i$ 放置在最多 $2^{p(X_i)}$ 个链表中,最多被扫描 $2^{p(X_i)}$ 次.在后续计算中,含有 $X_i$ 的链表个数不再增多.故 $X_i$ 在链表建立过程中最多被扫描 $2^{p(X_i)}+(n-p(X_i))2^{p(X_i)}$ 次.$X\neg_i$ 每被扫描一次只需常数次链表的指针修正操作,又 $p(X_i)\le p$.故建立相容点集链表的时间复杂性为 $O(m\cdot(n-p+1)\cdot 2^p)$. □

文献[19]建立散列表的时间复杂性是 $O(m\cdot n\cdot 2^p)$,其计算时间主要体现在插入散列表顶点标号的次数上.图 8 给出针对模拟数据的两种计算方法的实际测试结果.产生图 8 的实验中,测得两种实现方式的速度比最小是 5.3,最大是 11.2,平均为 7.1.且由图 8 可知,随着指纹向量个数的增加,建立相容点集散列表与链表时间比呈递增趋势.

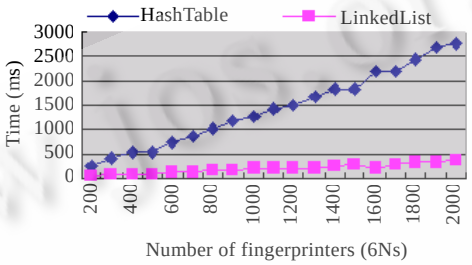


Fig.8 Compraision of speeds between HashTable and LinkedList

图 8 散列表法与链表法速度对比

**定理 3**. 因划分算法第 1 步结束,每个链表中存储的点集必为相容点集.相容点集总数不多于散列表方式得到的相容点集数.若在链表分解中存在一步操作满足 $F_1=\varnothing$或 $F_2=\varnothing$,则算法产生的相容点集数必少于散列表方式得到的相容点集数.

证明:设 $F$ 是前 $k-1$ 位相容的点集,算法扫描第 $k$ 位得到 $F_1\cup F_3$ 和 $F_2\cup F_3$ 必为前 $k$ 位相容.当然,扫描第 $n$ 位得到的每个链表均含有前 $n$ 位相容的点集.设对应于 $F$ 的结果向量为 $r[1],\ldots,r[k-1],x,\ldots,x$,其中,$r[1],\ldots,r[k-1]\in\{0,1\}^{k-1},x$ 不确定.若 $F_1=\varnothing$,则算法不产生对应 $r[1],\ldots,r[k-1],0,x,\ldots,x$ 的相容点集;若 $F_2=\varnothing$,则算法不产生对应 $r[1],\ldots,r[k-1],1,x,\ldots,x$ 的相容点集.散列表方法中,对应于 $r[1],\ldots,r[k-1],0,x,\ldots,x$ 和 $r[1],\ldots,r[k-1],1,x,\ldots,x$ 的相容点集均产生. □

我们目前无法给出链表法所得相容点集数少于散列表法所得相容点集数的确切值.随机数据测试表明,逐位扫描法得到的相容点集数平均不超过散列表法所得相容点集数的 49%.空间复杂性的降低得益于指纹向量丢失位的不对称性.这种空间复杂性的降低,对于算法求解能力的提高是最关键的.

考虑划分团并删除其顶点的时间复杂性.判断一个团是唯一极大团所需时间为 $O(m)$.判断一个最大团所需时间为 $O(m\cdot 2^p)$.一个团最多含有 $m$ 个顶点,每个顶点最多出现在 $2^p$ 个相容集中.删除一个团的所有顶点只需将链表空间依次释放,删除一个团中一个顶点,只是最后一次访问链表时扫描整个链表并释放,因此,只需访问一个顶点两次即可完成顶点删除.因此,在一个链表中删除一个顶点的时间复杂性为 $O(1)$.所以有:

**定理 4**. 求得一个唯一极大团或最大团,并完成删除计算的时间复杂性为 $O(m\cdot 2^p)$.

实际上,删除顶点对后续的最大团划分也具有明显的加速作用.最后给出改进贪心算法与散列表算法计算速度的实际测试结果,见表 4.表 4 的每列数据是根据随机产生的 100 个 BCMV($p$)实例的实际计算时间的平均值得到的.测试中,我们随机产生了一个指纹向量集:$p$=12,$n$=20,$m$=1000,结果向量个数为 20.文献[19]算法求解时间超过 60 小时,改进算法计算时间为 7 小时.对随机产生实例的测试表明,计算同一个 BCMV($p$)实例($p\le 13$),改进算法平均耗时为散列表法的 20%.进一步的测试分析表明,随着丢失值位数的增加,改进算法的计算速度与原算法相比呈增大趋势,改进效果越来越明显.当丢失值位数大于 6 时,改进算法几乎总可用原算法 11%的耗时

计算与原算法相同的实例.

**Table 4** Comparison of two algorithms speeds (Unit: ms)

**表 4** 两种算法计算速度实例对比 (单位:ms)

| Algorithm \ $p$ | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|
| HashTable | 974 | 4 703 | 77 297 | 877 062 | 5 571 313 |
| LinkedList | 688 | 1 938 | 7 984 | 84 406 | 606 297 |

### 3.3 算法实验数据设计

生成指纹向量集需首先确定 4 个参数:指纹向量中丢失值个数($p$)、指纹向量的长度($n$)、指纹向量个数($m$)以及结果向量个数($d$).BCMV($p$)实例的模拟数据是由如下步骤生成的:

指纹向量集生成算法.

//输入参数 $m,n,p,d$.

1. 随机产生正整数集合 $S=\{s_1,s_2,\ldots,s_d\}$,满足 $\sum_{i=1}^{d} s_i=m$.

2. 随机生成结果向量集 $R=\{r_1,r_2,\ldots,r_d\}$.$r_i=r_i[1],\ldots,r_i[n]\in\{0,1\}^n$,$1\le i\le d$.

3. 产生 $s_i$ 个 0-1-$N$ 指纹向量 $X_{i,1},\ldots,X_{i,s_i}$,$1\le i\le d$.产生 $X_{i,j}$,$1\le j\le s_i$:随机选择正整数集合 $P\subseteq\{1,2,\ldots,n\}$,且$|P|=p$.若 $k\in P$,则 $X_{i,j}[k]=N$;否则,$X_{i,j}[k]=r_i[k]$.

4. 输出 $X_{i,j}$,$1\le i\le d$,$1\le j\le s_d$.

本文所有测试均在 P4/1.8G/256M 个人计算机上使用 VC++编程实现.

## 4 结束语

本文证明了 BCMV(2)是 NP-Hard,并改进了 BCMV($p$)的贪心团划分算法.改进方法尝试利用链表实现原始算法.算法利用新数据结构和新方法降低了产生相容点集和划分唯一极大团的时间复杂性,利用指纹向量丢失位的不对称性质降低了空间复杂性.实测表明,改进算法可以在普通个人计算机上计算丢失值位数不超过 14 的 BCMV($p$)实例,而当丢失值位数超过 11 时,原算法在普通计算机上的计算时间就超过了 24 小时.以后的工作是讨论指纹向量聚类的近似计算复杂性,并设计其多项式时间近似算法.

**References**:


[1] Drmanac R, Drmanac S. CDNA screening by array hybridization. Methods in Enzymology, 1999,303:165−178.

[2] Drmanac S, Stavropoulos NA, Labat I, Vonau J, Hauser B, Soares MB, Drmanac R. Gene representing cdna clusters defined by hybridization of 57,419 clones from infant brain libraries with short oligonucleotide probes. Genomics, 1996,37:29−40.

[3] Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. PNAS, 1998, 95:14863−14868. http://www.pnas.org/cgi/content/abstract/95/25/14863?ijkey=IqMr2M.gVTJZY

[4] Sneath PHA, Sokal RR, Numerical Taxonomy. San Francisco: W.K. Freeman and Company, 1973.

[5] Herwig R, Poustka AJ, Muller C, Bull C, Lehrach H, O'Brien J. Large scale clustering of cdna-fingerprinting data. Genome Research, 1999,9(11):1093−1105.

[6] Milosavljevic A, Strezosca Z, Zeremski M, Grujic D, Paunesku T, Crkvenjakov R. Clone clustering by hydbridization. Genomics, 1995,27:83−89.

[7] Meier-Ewert S, Lange J, Gerts H, Herwig R, Schimitt A, Freund J, Elge T, Mott R, Herrmann B, Lehrach H. Comparative gene expression profiling by olignucleotide fingerprinting. Nucleic Acids Research, 1998,26(9):2216−2223.

[8] Ding CHQ. Analysis of gene expression profiles: Class discovery and leaf ordering. In: Pevzner P, eds. Proc. of the RECOMB 2002. New York: ACM, 2002. 127−136. http://portal.acm.org/citation.cfm?id=565196.565212

[9] Hartuv E, Schimitt A, Lange J, Meier-Ewert S, Lehrach H, Shamir R. An algorithm for clustering cdna fingerprints. Genomics, 2000,66(3):249−256.

[10] Sharan R, Shamir R. Click: A clustering algorithm with applications to gene expression analysis. In: Bourne P, ed. Proc. of the ISMB 2000. AAAI Press, 2000. 307−316. http://portal.acm.org/citation.cfm?id=645635.660836

[11] Xing EP, Karp RM. Cliff: Clustering of high dimensional microarray data via iterative feature filtering using normalized cuts. Bioinformatics, 2001,17(Suppl.):306−315.

[12] Barash Y, Friedman N. Context-Specific Baysian clustering for gene expression data. In: Sankoff D, ed. Proc. of the RECOMB 2001. New York: ACM, 2001. 12−21.
[13] Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns. Journal Computational Biology, 1999,6:281−297.
[14] McLachlan GJ, Bean RW, Peel D. A mixture model-based approach to the clustering of microarray expression data. Bioinformatics, 2002,18(3):413−422.
[15] Pan W, Lin J, Le CT. Model based cluster analysis of microarray gene-expression data. Genome Biology, 2002,3(2):1−8.
[16] Tamayo P, Slonim J, Mesirov D, Zhu J, Kitareewan S, Dmitrovsky E, Lander E, Golub T. Interpreting patterns of gene expression with self-organizing maps: Methods and applications to hematopoietic differention. PNAS, 1999,96:2907−2912.
[17] Toronen P, Kolehmainen M, Wong G, Castren E. Analysis of gene expression data using self-organizing maps. FEBS Letters, 1999, 451:142−146.
[18] Shamir R, Sharan R. Algorithmic approaches to clustering gene expression data. Current Topics in Computational Molecular Biology. 2002. 269−300. http://www.math.tau.ac.il/~rshamir/papers/book_rs.ps.gz
[19] Figueroa A, Borneman J, Jiang T. Clustering binary fingerprint vectors with values for DNA array data analysis. In: Blauvelt Pat, eds. Proc. of the Computational Systems Bioinformatics. Washington: IEEE Computer Society, 2003. 38−47. http://portal.acm.org/citation.cfm?id=937976.938130
[20] Valinsky L, Vedova GD, Jiang T, Borneman J. Oligonucleotide fingerprinting of ribosomal rna genes for anaysis of fungal community composition. Applied and Enviromental Microbiology, 2002,68(12):5999−6004.
[21] Karp RM, Reducibility among combinatorial problems. In: Complexity of Computer Computation, Miller RE, Thatcher JW, eds. New York: Plenum Press, 85−103.
[22] Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: Freeman WH and Company, 1979.
[23] Liu PQ, Fan H, Zhu DM. Cluster analysis for DNA array data with missing values.Computer Science,2004,31(Suppl.):136−140 (in Chinese with English abstract).

## 附中文参考文献:

[23] 刘培强,范辉,朱大铭.DNA 阵列数据分析与不确定数据解决.计算机科学,2004,31(增刊):136−140.

刘培强(1970－),男,山东淄博人,博士生,主要研究领域为算法及其复杂性理论.

范辉(1962－),男,教授,主要研究领域为信息可视化,算法及其复杂性理论.

朱大铭(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为算法与复杂性,神经网络.

马绍汉(1938－),男,教授,博士生导师,主要研究领域为算法与复杂性,人工智能.

谢青松(1965－),男,副教授,主要研究领域为算法及复杂性.