基于联合意义度量的 Top-*K* 图模式挖掘

刘 勇 高 宏 李建中
(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 提出了一个新的研究问题:如何挖掘 Top-*K* 图模式,联合起来使某个意义度量最大化.利用信息论的概念,给出了两个具体问题的定义 MES 和 MIGS,并证明它们是 NP-难.提出了两个高效算法 Greedy-Top*K* 和 Cluster-Top*K*. Greedy-Top*K* 先产生频繁子图,然后按增量贪心方式选择 *K* 个图模式. Cluster-Top*K* 先挖掘频繁子图的一个代表模式集合,然后从代表模式中按增量贪心方式选择 *K* 个图模式.当意义度量满足 submodular 性质时,Greedy-Top*K* 能提供近似比保证. Cluster-Top*K* 没有近似比保证,但比 Greedy-Top*K* 更高效. 实验结果显示,在结果可用性方面,文中提出的 Top-*K* 挖掘优于传统的 Top-*K* 挖掘. Cluster-Top*K* 比 Greedy-Top*K* 快至少一个数量级. 而且,在质量和可用性方面,Cluster-Top*K* 的挖掘结果非常类似于 Greedy-Top*K* 的挖掘结果.

关键词 图挖掘;图数据库;频繁子图;代表模式;联合熵;信息增益
中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2010.00215

Top-*K* Graph Patterns Mining Based on Some Joint Significance Measure

LIU Yong GAO Hong Li Jian-Zhong
(*School of Computer Science and Technology, Harbin Institute of Technology, Harbin* 150001)

Abstract This paper proposes a novel problem of mining top-k graph patterns that jointly maximize some significance measure from graph databases. By exploiting the concepts of information theory, it gives two problem formulations, *MES* and *MIGS*, and proves that they are NP-hard. Two efficient algorithms, Greedy-Top*K* and Cluster-Top*K*, are proposed for this new problem. Greedy-Top*K* first generates frequent subgraphs, and then incrementally and greedily selects *K* graph patterns from frequent subgraphs. Cluster-Top*K* first mines a set of representative patterns for frequent subgraphs, and then incrementally and greedily selects *K* graph patterns from representative patterns. When a given significance measure satisfies the submodular property, Greedy-Top*K* can provide tight approximation bound. Cluster-Top*K* has no approximation bound guarantee but is more efficient than Greedy-Top*K*. Extensive experimental results demonstrate that the Top-*K* mining proposed in this paper is superior to the traditional Top-*K* mining in terms of results usefulness. Cluster-Top*K* can achieve at least an order of magnitude speedup than Greedy-Top*K*, while achieving comparable mining results in terms of quality and usefulness.

Keywords graph mining; graph database; frequent subgraph; representative pattern; joint entropy; information gain

1 引 言

作为一种通用的数据结构,图可以用来表示数据对象之间的各种复杂关系.例如:图可以表示化合物的分子结构、蛋白质交互网络、社会网络、Web 结构图等.很多与图有关的应用都需要利用图模式来管理、查询和分析图数据.例如:图查询[1]可以利用

收稿日期:2009-07-15;最终修改稿收到日期:2009-09-16.本课题得到国家"九七三"重点基础研究发展规划项目基金(2006CB303005)和国家自然科学基金(60533110,60773063)资助.**刘 勇**,男,1975 年生,博士研究生,主要研究方向为图挖掘和图数据管理. E-mail: liuyong123456@hit.edu.cn. **高 宏**,女,1966 年生,教授,博士生导师,主要研究领域为数据仓库与数据挖掘、传感器网络等. **李建中**,男,1950 年生,教授,博士生导师,主要研究领域为数据仓库与数据挖掘、传感器网络、数据网格和计算生物学等.

图模式建立有效的索引，图分类[2]可以利用图模式建立有效的分类模型.因此，从图数据库中发现有用的图模式已成为数据挖掘领域一项重要的研究课题.

目前，与图模式挖掘有关的绝大部分研究主要集中在如何高效地挖掘频繁子图[3-8]以及频繁子图的各种简洁表示(频繁闭图模式[9]和频繁最大图模式[10-11]).然而，很少有研究考虑如何根据用户给定的意义度量来挖掘图模式.本质上，频繁子图挖掘所使用的支持度恰恰是一种特殊的意义度量.然而，在不同的应用中，用户很可能需要不同的意义度量.

给定一个意义度量，传统 Top-K 挖掘方法对所有可能的图模式根据度量值的大小排序，输出前 K 个图模式.然而，传统 Top-K 挖掘并不考虑图模式之间的相关性，输出的 Top-K 模式可能非常相似.第 5 节的图 5 显示了信息增益作为意义度量时，传统 Top-K 挖掘方法从一个图数据库中挖掘的 Top-5 图模式.这 5 个图模式在结构上非常相似.如果用户得到其中一个图模式，就会对其它图模式失去了兴趣，因为在实际应用中用户希望得到的是一个多样化的图模式集合.

为了克服传统 Top-K 挖掘方法的缺点，本文研究基于联合意义度量的 Top-K 图模式挖掘方法.联合意义度量的作用域是图模式集合而不是图模式，因此充分考虑了图模式之间的相关性.给定一个联合意义度量，本文要研究的问题就是挖掘 Top-K 图模式，联合起来使该意义度量最大化.本文的目标是设计一个适用于任何意义度量的通用的 Top-K 挖掘算法.

本文首先讨论了适用于图模式集合的意义度量，并利用信息论中的概念(联合熵和信息增益)给出了两个具体问题的定义 MES 和 MIGS，然后证明了它们是 NP-难问题.为了高效地挖掘基于联合意义度量的 Top-K 图模式，本文给出了两个算法 Greedy-Top$K$ 和 Cluster-Top$K$. Greedy-Top$K$ 先产生频繁图模式(或频繁闭图模式)，然后按增量贪心方式选择 K 个图模式.本文证明了如果用户给定的意义度量满足 submodular 性质，Greedy-Top$K$ 能提供近似比保证.然而，当频繁图模式(或频繁闭图模式)数量很大时，Greedy-Top$K$ 效率低，可扩展性差.为此，本文又提出了另一个更高效的算法 Cluster-Top$K$. Cluster-Top$K$ 先从图数据库中挖掘所有频繁图模式的一个代表模式集合，然后从代表模式中按增量贪心方式选择 K 个图模式. Cluster-Top$K$ 最大的优点是无需产生频繁图模式(或闭图模式)就能快速地挖掘一个代表模式集合.而且，因为代表模式的数量比闭图模式的数量少很多，所以 Cluster-Top$K$ 的贪心选择也比 Greedy-Top$K$ 的贪心选择快很多.此外，本文还从理论上严格证明了 Cluster-Top$K$ 产生的解和 Greedy-Top$K$ 产生的解非常接近.

实验结果表明本文提出的 Top-K 挖掘在结果质量和可用性方面要远远优于传统 Top-K 挖掘. Cluster-Top$K$ 和 Greedy-Top$K$ 在结果质量和可用性方面非常接近.然而，Cluster-Top$K$ 能比 Greedy-Top$K$ 快 1～2 个数量级.

本文第 2 节介绍预备知识；第 3 节给出问题定义和 NP-难证明；第 4 节给出完整的算法描述和算法分析；第 5 节通过实验证明算法的有效性以及挖掘结果的可用性；第 6 节介绍相关工作；第 7 节总结全文.

2 预备知识

本节介绍图挖掘和信息论的一些基本概念.

定义 1(标号图). 标号图 G 定义为四元组 $G=(V,E,\Sigma,l)$，其中，V 是顶点集合，$E\subseteq V\times V$ 是边集合，Σ 是标号集合，$l:V\cup E\to\Sigma$ 是一个函数，用来对顶点和边分配标号.

定义 2(子图同构). 给定两个图 $G=(V,E,\Sigma,l)$ 和 $G'=(V',E',\Sigma',l')$，一个从 G 到 G' 的子图同构是一个单射函数 $f:V\to V'$，满足：(1) $\forall u\in V$，$l(u)=l'(f(u))$；(2) $\forall(u,v)\in E$，$(f(u),f(v))\in E'$ 并且 $l((u,v))=l'((f(u),f(v)))$.单射函数 f 也称为 G 在 G' 中的一个嵌入.

如果存在一个从 G 到 G' 的子图同构，则 G 称为 G' 的子图，G' 称为 G 的超图，记为 $G\subseteq G'$.如果 $G\subseteq G'$ 且 $G\neq G'$，则 G 称为 G' 的真子图，G' 称为 G 的真超图，记为 $G\subset G'$.子图同构测试已被证明是一个 NP-完全问题[12].如果 $G\subseteq G'$，也称 G' 包含 G.

给定一个图数据库 $D=\{G_1,G_2,\cdots,G_n\}$ 和一个图模式 p，p 在 D 中的支持集定义为 D 中包含 p 的图集合，记为 $D_{supp}(p)=\{G_i\mid p\subseteq G_i,G_i\in D\}$. $|D_{supp}(p)|$ 称为 p 在 D 中的支持度，记为 $supp(p;D)$. $|D_{supp}(p)|/|D|$ 称为 p 在 D 中的相对支持度.支持度度量具有反单调性质：如果 $p_1\subseteq p_2$，则 $supp(p_1;D)\geq supp(p_2;D)$.对于用户给定的一个最小支持度阈值 min_sup，如果 $supp(p;D)\geq min\_sup$，称 p 在 D 中是频繁的.D 中所有频繁图模式集合记为 $FS=\{p\mid supp(p;$

$D)\geq min\_sup\}$. 本文中，min_sup 既可以表示绝对最小支持度阈值，又可以表示相对最小支持度阈值.

如果在 D 中 p 的任何真超图与 p 支持度都不相同，则称 p 是 D 中的闭图模式. D 中所有频繁闭图模式集合记为 $CS=\{p\mid p\in FS, \neg\exists p'\in FS$ 使得 $p\subset p'$ 并且 $supp(p;D)=supp(p';D)\}$. 上下文明确时，可用 $supp(p)$ 代替 $supp(p;D)$.

因为本文后面介绍的意义度量将使用信息论的一些概念[13]，这里先回顾一下有关的基本定义.

定义 3(熵). 随机变量 x 的熵定义为

$$H(x)=-\sum_{v_x\in dom(x)} p(v_x)\log(p(v_x)),$$

其中 $dom(x)$ 是 x 的定义域，$p(v_x)$ 是 x 等于 v_x 时的概率.

定义 4(条件熵). 在给定随机变量 x 的条件下，随机变量 y 的条件熵定义为

$$H(y\mid x)=-\sum_{v_x\in dom(x)}\sum_{v_y\in dom(y)} p(v_x,v_y)\log(p(v_y\mid v_x)).$$

定义 5(联合熵). 随机变量 x 和 y 的联合熵定义为

$$H(x,y)=-\sum_{v_x\in dom(x)}\sum_{v_y\in dom(y)} p(v_x,v_y)\log(p(v_x,v_y)).$$

3　问题定义

本节首先讨论用于图模式集合的意义度量，然后给出两个具体问题的定义 MES 和 MIGS，最后证明它们是 NP-难问题.

文献中已有大量的用于单一模式(包括项集模式，序列模式和图模式)的意义度量. 例如：在统计学领域，x^2 检验和 Pearson 相关可以用来度量模式的统计意义. 在数据挖掘和机器学习领域，信息增益和交叉熵可以用来度量模式是否适合作为分类特征. 文献[14]总结了 21 个常用的意义度量.

尽管文献中给出的大部分意义度量都能直接用来度量单一图模式的意义，然而它们中的绝大部分并不能用来量化一个图模式集合的意义. 本文算法需要用户提供一个意义度量 M 使得 M 能量化一个图模式集合的意义. 信息论中的联合熵和信息增益可以用来量化一个图模式集合的意义. 在给出具体的问题定义之前，我们先给出本文要研究的一般问题.

假设 S 是给定的图模式集合(例如：S 可以表示某个图数据库中所有频繁图模式集合)，T 是 S 的子集合，$M(T)$ 是 T 的意义度量. 本文要解决的问题就是发现 S 的一个大小为 K 的子集合 T^* 使得 $M(T^*)$ 被最大化，即

$$T^*=\mathrm{argmax}_{T\subset S,|T|=k}M(T) \tag{1}$$

如果所有候选的图模式给定的话，上面定义的问题显然是一个组合优化问题. 本文的目标是设计求解上述问题的通用算法，不受任何具体意义度量的限制. 为了方便描述算法，我们利用信息论中的联合熵和信息增益给出下面两个具体的问题定义.

定义 6(基于熵的意义度量最大化). 给定图数据库 D，D 的一个图模式集合 S(或者最小支持度 min_sup)和一个整数 K，最大化基于熵的意义度量问题(MES)就是从 S(或者 D 的所有频繁图模式集合)中发现一个大小为 K 的子集合 T 使得 $H(T)$ 被最大化，其中 $H(T)$ 是 T 中随机变量(图模式)的联合熵.

定义 7(基于信息增益的意义度量最大化). 给定图数据库 D，D 的一个图模式集合 S(或者最小支持度 min_sup)和一个整数 K. 假设 D 中每个图都有一个类标记，设 C 是表示类标记的随机变量. 最大化基于信息增益的意义度量问题(MIGS)就是从 S(或者 D 的所有频繁图模式集合)中发现一个大小为 K 的子集合 T 使得 $IG(T)=H(C)-H(C\mid T)$ 被最大化，其中 $H(C)$ 是 C 的无条件熵，$H(C\mid T)$ 是 T 中随机变量(图模式)给定的条件下 C 的条件熵.

基于熵的意义度量经常用来在无监督的环境下度量不确定性，而基于信息增益的意义度量经常用来在有监督的环境下选择强分类特征. 在上面的形式化定义中，我们是把每个图模式 p 看成了一个随机变量 v_p，如果数据库中的某个图 G 含有 p，则在 G 上 v_p 等于 1，否则 v_p 等于 0.

通常，当某个具体的意义度量给定时，式(1)定义的问题是 NP-难问题. 下面，我们证明 MIGS 问题(定义 7)是 NP-难的. 类似地，也可以证明 MES 问题(定义 6)是 NP-难的.

定理 1. 最大化基于信息增益的意义度量问题(MIGS)是 NP-难的.

证明. 通过把最大覆盖问题规约到 MIGS 问题，我们来证明该定理.

设 $MC=(E,S,K)$ 是最大覆盖问题的任一实例，其中 $E=\{e_1,e_2,\cdots,e_n\}$ 是元素集合，$S=\{S_1,S_2,\cdots,S_L\}$ 是集合族. 求解 MC 要求从 S 中选择 K 个集合，使得这 K 个集合覆盖的元素数量最大化. 我们可以在多项式时间内把 MC 转化成 MIGS 问题的一

个实例.

(1)把 E 中每个元素看成一个边的标号,那么 S 中的某个集合 $S_i=\{e_{i1},e_{i2},\cdots,e_{it}\}$ 可以转换成图 1 所示的图模式.该图模式用 $P(S_i)$ 表示.$P(S_i)$ 中所有结点的标号都相同,但是边的标号各不相同.以相同的方式,我们可以把 S 中的所有集合都转换成对应的图模式.所有这些图模式构成的集合用 PS 表示.

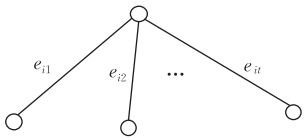


图 1 对应 $S_i=\{e_{i1},e_{i2},\cdots,e_{it}\}$ 的图模式 $P(S_i)$

(2)我们首先为 E 中的每个元素构造一个图.如果一个元素 e 能被 $\{S_{i1},S_{i2},\cdots,S_{im}\}$ 中的集合覆盖,我们为 e 构造一个图 2 所示的图,其中 $P(S_i)$ 是对应集合 S_i 的图模式,连接各个图模式 $P(S_i)$ 的边标号都相同.显然,每个元素都将对应一个图,我们把所有这样的图标记为正类,放入数据库 DB.此时,$|DB|=n$.

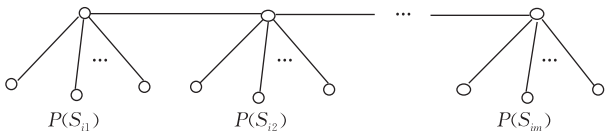


图 2 对应 $\{S_{i1},S_{i2},\cdots,S_{im}\}$ 的数据库图

(3)我们然后随意构造 n 个图使得这 n 个图中不含有 PS 中的任何图模式.我们把这 n 个图标记为负类,放入数据库 DB.此时,$|DB|=2n$.

给定上面构造的图数据库 DB,DB 中的一个图模式集合 PS 以及正整数 K,我们得到了 MIGS 问题的一个实例 IMIG.

DB 中正类图个数等于负类图个数,因此 $H(C)=1$,其中 C 是表示类标记的随机变量.设 T 是 PS 中任意一个大小为 K 的子集合.因为 T 中每个图模式对应 S 中的一个集合,因此从 T 可得到 S 的一个大小为 K 的子集合 S'.设被 S' 覆盖的元素数是 x.对 T 中的任何图模式 p,DB 中的任何负类图都不含有 p.根据条件熵定义,

$$H(C|T)=-\frac{(n-x)+n}{2n}\times\left(\frac{n-x}{(n-x)+n}\times\log\frac{n-x}{(n-x)+n}+\frac{n}{(n-x)+n}\times\log\frac{n}{(n-x)+n}\right)=-\left(\frac{n-x}{2n}\log\frac{n-x}{2n-x}+\frac{1}{2}\log\frac{n}{2n-x}\right).$$

因此,$IG(T)=H(C)-H(C|T)=1+\frac{n-x}{2n}\times\log\frac{n-x}{2n-x}+\frac{1}{2}\log\frac{n}{2n-x}$.令 $f(x)=\frac{n-x}{2n}\log\frac{n-x}{2n-x}+\frac{1}{2}\log\frac{n}{2n-x}$.可以证明 $f(x)$ 的一阶导数 $f'(x)$ 大于 0,$f(x)$ 是一个单调递增函数.当 x 的值增大时,$IG(T)$ 也增大.因此,IMIG 的最优解能被用来导出 MC 的一个最优解. 证毕.

4 挖掘 Top-K 图模式

MES 和 MIGS 都是 NP-难问题,因此需要设计近似算法或启发式算法求解它们.本文提出了两个高效算法 Greedy-Top$K$ 和 Cluster-Top$K$ 来求解它们.Greedy-Top$K$ 从频繁图模式集合中贪心选择 Top-K 模式,具有近似比保证.Cluster-Top$K$ 从图数据库中挖掘代表图模式集合,然后再从中贪心选择 Top-K 模式.尽管 Cluster-Top$K$ 没有近似比保证,Cluster-Top$K$ 却具有极高的挖掘效率.

4.1 Greedy-Top$K$ 算法

4.1.1 算法设计

Greedy-Top$K$ 算法采用著名的贪心策略,从所有频繁图模式集合中增量地选择 K 个图模式.为了应用贪心策略,我们定义了一个收益函数 b.Greedy-Top$K$ 每次选择使收益函数 b 最大的图模式.假定 T 是已经选择的图模式集合.对 MES 和 MIGS 问题,图模式 p 的收益函数 b 定义如下

$$b(p)=\begin{cases}H(T,p)-H(T), & \text{对 MES}\\ H(C|T)-H(C|T,p), & \text{对 MIGS}\end{cases}\quad(2)$$

根据式(2)给出的贪心规则,我们设计了一个贪心算法 Greedy-Top$K$,如算法 1 所示.初始化时,使用某个图挖掘算法(例如:gSpan[6])挖掘所有频繁图模式集合 F,同时置结果集 T 为空.然后算法选择最有意义的图模式 p^* 放入 T 中.此后,算法进入一个循环过程,每次从剩余的图模式集合 $F-T$ 中选择一个使收益函数 b 最大的图模式 p,放入 T 中.当结果集 T 中图模式个数等于 K 时,退出循环,算法结束.

算法 1. Greedy-Top$K$.

输入:图数据库 D;最小支持度阈值 min_sup;意义度量 M;输出模式个数 K.

输出:使 M 最大化的 K 个图模式

1. 根据 min_sup,挖掘 D 中所有频繁图模式 F;

2. 从 F 中选择一个图模式 p^* 使 $M(p)$最大化；

3. $T=\{p^*\}$；

4. While ($|T|<K$) do

5. 从 $F-T$ 中选择图模式 p 使得 $b(p)$ 最大化；

6. $T=T\cup\{p\}$；

7. 输出 T.

下面，我们分析算法 Greedy-Top$K$ 的近似比. 当意义度量 M 满足 submodular 性质(见下面的定义)时，贪心算法将给出近似解[15].

定义 8(submodular 性质). 设 M 是定义在一个集合 S 上的度量函数. 如果对任何 $T\subset T'\subseteq S$ 和任何 $p\in S$，都有 $M(T\cup\{p\})-M(T)\geq M(T'\cup\{p\})-M(T')$，则称 M 是一个 submodular 的集合函数.

仔细分析本文给出的两个意义度量，我们发现基于熵的意义度量满足 submodular 性质，因此算法 Greedy-Top$K$ 对 MES 问题能给出如下的近似比.

定理 2. 设 T 是算法 Greedy-Top$K$ 选择的 K 个图模式集合，T^* 是使联合熵最大化的 K 个图模式集合. 那么，$\frac{H(T^*)}{H(T)}\leq\frac{e}{e-1}$.

文献[15]给出了完整的证明过程.

我们发现基于信息增益的意义度量并不满足 submodular 性质. 因此，对 MIGS 问题，算法 Greedy-Top$K$ 不能给出像定理 2 那样的近似比. 然而，我们注意到，对 MIGS 问题，无条件熵 $H(C)$是任何模式集合信息增益的上界. 因此，算法 Greedy-Top$K$ 对 MIGS 问题能给出如下的联机近似比.

定理 3. 设 T 是算法 Greedy-Top$K$ 选择的 K 个图模式集合. 那么，算法 Greedy-Top$K$ 对 MIGS 问题给出的联机近似比至多是 $H(C)/IG(T)$.

4.1.2　裁剪技术

算法 Greedy-Top$K$ 的第 1 步需要遍历图模式搜索空间，挖掘所有频繁图模式. 本小节研究如何根据给定的意义度量设计有效的裁剪技术，将其集成到图模式挖掘算法的框架中裁剪图模式搜索空间.

所有频繁子图挖掘算法都利用了支持度的反单调性质来裁剪搜索空间. 即一个图模式 p 的支持度是 p 的所有超图支持度的上界. 然而，通常情况下，给定的意义度量并不具有反单调性质. 因此，我们不能像利用支持度那样简单地利用给定的意义度量.

幸运的是，给定一个图模式 p 的意义度量值 $M(p)$，我们可以导出 p 的所有超图意义度量值的上界，从而根据这个上界来裁剪图模式搜索空间.

我们仍然采用 Greedy-Top$K$ 算法的框架来增量地选择 Top-K 图模式. 因此，我们分两种情况介绍裁剪技术：当结果集为空时，选择第 1 个图模式；当某些图模式已经在结果集中，选择下一个图模式.

(1) 选择第 1 个图模式.

下面讨论结果集为空时如何设计有效的裁剪技术. 请注意，我们在 4.1.1.1 节和 4.1.1.2 节给出的裁剪技术可以应用到任何频繁子图挖掘算法的框架中. 因此，我们未给出具体的挖掘算法. 定理 4 给出了基于熵的裁剪条件. 定理 5 给出了基于信息增益的裁剪条件.

定理 4. 设 D 是给定的图数据库，q 是 p 的任意超图. 如果 $supp(p;D)\leq 1/2$，则 $H(q)\leq H(p)$.

定理 5. 给定图数据库 $D=PD+ND$，其中 PD 和 ND 分别是正类和反类图集合. 设 q 是 p 的任意超图，PD 中有 x 个图含有 p，ND 有 y 个图含有 p. 那么，

$$IG(q)\leq\max\begin{cases}H+\frac{|PD|-x}{|D|}\log\frac{|PD|-x}{|D|-x}+\\ \quad\frac{|ND|}{|D|}\log\frac{|ND|}{|D|-x},\\ H+\frac{|ND|-y}{|D|}\log\frac{|ND|-y}{|D|-y}+\\ \quad\frac{|PD|}{|D|}\log\frac{|PD|}{|D|-y},\\ IG(p).\end{cases}$$

其中，$H=-\left(\frac{|PD|}{|D|}\log\frac{|PD|}{|D|}+\frac{|ND|}{|D|}\log\frac{|ND|}{|D|}\right)$.

(2) 选择下一个图模式.

结果集不空时如何设计有效的裁剪技术. 在给出具体的裁剪技术之前，我们先定义一个新的概念——等价类.

定义 9(等价类). 设 D 是图数据库，G 是 D 中的任意图，T 是已经选择的图模式集合. G 相对于 T 的等价类定义为$\{G'|G'\in D,\forall p\in T,I(p\subseteq G)=I(p\subseteq G')\}$，其中 $I(\cdot)$是指示函数.

根据上面等价类的定义，一个图模式集合 T 可以将数据库 D 划分成若干个等价类(块)集合 $D_T=\{B_i|1\leq i\leq l\}$，并且 $D=\bigcup\limits_{1\leq i\leq l}B_i$. 利用等价类的定义，我们可以在结果集非空的情况下导出有效的裁剪条件. 定理 6 给出了基于熵的裁剪条件. 定理 7 给出了基于信息增益的裁剪条件.

定理 6. 设 D 是给定的图数据库，T 是已经选择的图模式集合，$D_T=\{B_i|1\leq i\leq l\}$是用 T 划分 D 得到的等价类集合，q 是 p 的任意超图($p\subseteq q$)，并且 $p\notin T$，$q\notin T$. 那么，

$$H(T\cup\{q\})\leq \sum_{B_i\in D_T}\begin{cases}-\frac{|B_i|}{|D|}\log\frac{|B_i|}{2|D|}, & supp(p;B_i)\geq 1/2\times B_i\\ H_{B_i}(p), & supp(p;B_i)<1/2\times B_i\end{cases}$$

其中 $H_{B_i}(p)=-\left(\frac{x_i}{|D|}\log\frac{x_i}{|D|}+\frac{|B_i|-x_i}{|D|}\times\log\frac{|B_i|-x_i}{|D|}\right)$，$x_i$是$B_i$中含有$p$的图个数.

定理 7. 给定图数据库 $D=PD+ND$，其中 PD 和 ND 分别是正类和反类图集合. T 是已经选择的图模式集合，$D_T=\{B_i|1\leq i\leq l\}$是用 T 划分 D 得到的等价类集合，q 是 p 的任意超图($p\subseteq q$)，并且 $p\notin T$，$q\notin T$，那么，

$$IG(T\cup\{q\})\leq H+\sum_{B_i\in D_T}\max\begin{cases}\alpha_{B_i}\\ \beta_{B_i}\\ \gamma_{B_i}\end{cases},$$

其中，$H=-\left(\frac{|PD|}{|D|}\log\frac{|PD|}{|D|}+\frac{|ND|}{|D|}\log\frac{|ND|}{|D|}\right)$，

$$\alpha_{B_i}=\frac{m_i-x_i}{|D|}\log\frac{m_i-x_i}{|B_i|-x_i}+\frac{n_i}{|D|}\log\frac{n_i}{|B_i|-x_i},$$

$$\beta_{B_i}=\frac{n_i-y_i}{|D|}\log\frac{n_i-y_i}{|B_i|-y_i}+\frac{m_i}{|D|}\log\frac{m_i}{|B_i|-y_i},$$

$$\gamma_{B_i}=\left(\frac{x_i}{|D|}\log\frac{x_i}{x_i+y_i}+\frac{y_i}{|D|}\log\frac{y_i}{x_i+y_i}\right)+\left(\frac{m_i-x_i}{|D|}\times\log\frac{m_i-x_i}{|B_i|-(x_i+y_i)}+\frac{n_i-y_i}{|D|}\log\frac{n_i-y_i}{|B_i|-(x_i+y_i)}\right),$$

m_i和n_i分别是B_i中正类图个数和负类图个数，x_i和y_i分别是B_i中含有p的正类图个数和含有p的负类图个数.

4.2 Cluster-Top$K$ 算法

Greedy-Top$K$ 算法需要先产生所有的频繁图模式，在频繁图模式数量较少的情况下，Greedy-Top$K$ 算法具有较高的效率. 然而，在实际应用中，当数据库中图较稠密或者用户给定的支持度阈值较低时，频繁子图挖掘算法通常产生大量的甚至指数级的频繁子图. 这使得 Greedy-Top$K$ 算法不能在合理的时间内完成任务，限制了该算法的可用性. 即使使用 4.1.1 节中的裁剪技术，Greedy-Top$K$ 也需要遍历图模式空间中的大量图模式. 为此，本节提出了另一个更高效的算法 Cluster-Top$K$.

Cluster-Top$K$ 算法的基本思想是将所有频繁图模式聚类成若干个簇，选择每个簇的中心作为代表模式构成一个候选集合 T，然后再使用贪心策略从 T 中增量地选择 K 个图模式. Cluster-Top$K$ 算法的优点是不需要先产生所有的频繁图模式，就能从图数据库中快速地挖掘代表模式得到候选集，具体内容见 4.2.2 节.

假设 $C=\{p_1,p_2,\cdots,p_n\}$是由若干个图模式构成的一个簇，p_i是该簇中心. 因为 p_i将作为代表模式被选择，p_i应具有这样的特点：$\forall P_j\in C$，$P_j\subseteq P_i$，并且 P_j和 P_i在数据库图中应经常一起出现，即 P_j和 P_i有类似的支持度. 下面，我们给出这种类型簇的两个形式化定义 δ-簇和 Δ-簇.

定义 10(δ-覆盖). 设 $\delta(0\leq\delta\leq1)$是用户给定的一个参数，p 和 q 是任意两个图模式. 如果满足 $q\subseteq p$，$1-\frac{supp(p)}{supp(q)}\leq\delta$，则称 q 被 p δ-覆盖.

定义 11 (δ-簇). 设 $\delta(0\leq\delta\leq1)$是用户给定的一个参数，$C=\{p_1,p_2,\cdots,p_n\}$是一个图模式集合. 如果 C 中存在图模式 p_i 满足 $\forall p_j\in C$，p_j 被 p_i δ-覆盖，则称 C 是一个 δ-簇，称 p_i 为该 δ-簇的代表模式(中心点).

定义 12(Δ-覆盖). 设 $\Delta(\Delta>0$ 的整数)是用户给定的一个参数，p 和 q 是任意两个图模式. 如果满足 $q\subseteq p$，$supp(q)-supp(p)\leq\Delta$，则称 q 被 p Δ-覆盖.

定义 13(Δ-簇). 设 $\Delta(\Delta>0$ 的整数)是用户给定的一个参数，$C=\{p_1,p_2,\cdots,p_n\}$是一个图模式集合. 如果 C 中存在图模式 p_i 满足 $\forall p_j\in C$，p_j 被 p_i Δ-覆盖，则称 C 是一个 Δ-簇，称 p_i 为该 Δ-簇的代表模式(中心点).

Cluster-Top$K$ 算法就是根据用户给定的参数 δ(或 Δ)，将频繁图模式集合划分成若干个 δ-簇(或 Δ-簇)，选择每个 δ-簇(或 Δ-簇)的代表模式构成候选集，然后再从中贪心选择 K 个图模式. 4.2.1 节分析了 Cluster-Top$K$ 算法采用这种聚类策略的优点. 4.2.2 节研究了如何在不产生所有频繁图模式的情况下快速地挖掘每个 δ-簇(或 Δ-簇)的代表模式. 4.2.3 节给出了具体的算法实现.

4.2.1 Cluster-Top$K$ 聚类策略的优点

本小节讨论 Cluster-Top$K$ 算法为什么采用这种先聚类后贪心的策略. 设 N 是给定图数据库 D 的大小，即$|D|=N$. 定义函数 $f(n)=\frac{n}{N}\log\frac{N}{n}(0\leq n\leq N)$. 下面的引理和定理将利用该函数.

引理 1. 设 D 是给定的图数据库，$|D|=N$，P_1和P_2是两个图模式，d 是P_1和P_2之间的海明距离(即

$d=\sum_{G_i\in D}|I(P_1\subseteq G_i)-I(P_2\subseteq G_i)|$，其中 I 是指示函数). 那么，$0\leq H(P_1,P_2)-H(P_1)\leq d(f(1)+f(N-1))$.

证明见文献[16]中的 Proposition 6.3.

根据引理 1，我们容易得到下面的推论 1.

推论 1.　设 D 是给定的图数据库，$|D|=N$，$F=\{P_1,P_2,\cdots,P_n\}$ 是任意图模式集合，P_{n+1} 是任意一个图模式，如果 P_n 和 P_{n+1} 之间的海明距离是 d，则 $H(P_1,\cdots,P_{n-1},P_n)-H(P_1,\cdots,P_{n-1},P_{n+1})\leq d(f(1)+f(N-1))$.

证明.　$H(P_1,\cdots,P_{n-1},P_n)-H(P_1,\cdots,P_{n-1},P_{n+1})\leq H(P_1,\cdots,P_n,P_{n+1})-H(P_1,\cdots,P_{n-1},P_{n+1})=H(P_n|P_1,\cdots,P_{n-1},P_{n+1})\leq H(P_n|P_{n+1})=H(P_n,P_{n+1})-H(P_{n+1})$. 因为 P_n 和 P_{n+1} 之间的海明距离是 d，再根据引理 1，可得 $H(P_1,P_2,\cdots,P_n)-H(P_1,\cdots,P_{n-1},P_{n+1})\leq H(P_n,P_{n+1})-H(P_{n+1})\leq d(f(1)+f(N-1))$.　证毕.

因为频繁图模式的数量通常很大，直接对频繁图模式进行贪心选择，时间复杂性太高. 对频繁图模式先聚类，只提取每个类的代表模式(中心点)构成候选集合，可以将频繁模式的数量降低 2～3 个数量级(见后面的实验结果). 然后对候选集合进行贪心选择，可使算法具有极高的效率. 而且，我们可以证明代表模式集合中的最优解和频繁图模式集合中的最优解差别很小. 定理 8 针对 MES 问题，分析了最优解之间的差异. 定理 9 针对 MIGS 问题，分析了最优解之间的差异. 定理 8 和 9 给出了利用 Δ-簇的分析结果. 类似地，我们也容易给出利用 δ-簇的分析结果.

以 MES 问题为例，我们分析具体的差异. 假设定理 8 中的 $N=10000$，$K=10$，$\Delta=5$，则有 $H(T^*)-H(S^*)\leq\Delta K(f(1)+f(N-1))\approx 0.073$. 这是理论上的最大可能差异，而实验中的结果显示：从聚类结果中选择的解和从频繁图模式集合中选择的解差别微乎其微.

定理 8.　设 D 是给定的图数据库，$|D|=N$，$F=\{P_1,P_2,\cdots,P_n\}$ 是 D 中频繁图模式集合，F 被划分成 m 个 Δ-簇的集合 $CS=\{C_1,C_2,\cdots,C_m\}$，其中每个 $C_i(1\leq i\leq m)$ 都是一个 Δ-簇，并且 $F=C_1\cup C_2\cup\cdots\cup C_m$. CS 中每个 Δ-簇的代表模式被选择构成了一个集合 $RS=\{R_1,R_2,\cdots,R_m\}$，其中 $R_i(1\leq i\leq m)$ 是 C_i 的代表模式. 再假设 T^* 是 F 中的一个大小为 K 的子集合，并且使 $H(T)$ 最大化. S^* 是 RS 的一个大小为 K 的子集合，并且使 $H(S^*)$ 最大化. 那么，$H(T^*)-H(S^*)\leq\Delta K(f(1)+f(N-1))$.

证明. 设 $T^*=\{Q_1,Q_2,\cdots,Q_K\}$. 因为 F 被划分成 m 个 Δ-簇的集合 CS，$RS=\{R_1,R_2,\cdots,R_m\}$ 是 CS 中 Δ-簇的代表模式集合，并且 $T^*\subset F$，根据 Δ-簇定义，对任意 $Q_i\in T^*$，都存在 RS 中的一个代表模式 R 使得 R 能 Δ-覆盖 Q_i. 不失一般性，假定 Q_1 被 R_1 Δ-覆盖. 根据 Δ-覆盖定义，可知 $Q_1\subseteq R_1$，$supp(Q_1)-supp(R_1)\leq\Delta$. 因此，$Q_1$ 与 R_1 之间的海明距离 $d=supp(Q_1)-supp(R_1)\leq\Delta$. 我们试图用 R_1 替换 Q_1. 令 $T=T^*-\{Q_1\}\cup\{R_1\}$. 根据推论 1，可得 $H(T^*)-H(T)\leq d(f(1)+f(N-1))\leq\Delta(f(1)+f(N-1))$. 以此类推，如果 T^* 中每个图模式都被 RS 中对应的代表模式所替换，可得到一个新的集合 T'，并且 $H(T^*)-H(T')\leq\Delta K(f(1)+f(N-1))$. 显然，$T'$ 是 RS 的一个大小为 K 的子集合. 因为 S^* 是 RS 的一个大小为 K 的子集合，并且使 $H(S^*)$ 最大化. 因此，$H(T^*)-H(S^*)\leq H(T^*)-H(T')\leq\Delta K(f(1)+f(N-1))$.　证毕.

定理 9.　设 D 是给定的图数据库，$|D|=N$，$F=\{P_1,P_2,\cdots,P_n\}$ 是 D 中频繁图模式集合，F 被划分成 m 个 Δ-簇的集合 $CS=\{C_1,C_2,\cdots,C_m\}$，其中每个 $C_i(1\leq i\leq m)$ 都是一个 Δ-簇，并且 $F=C_1\cup C_2\cup\cdots\cup C_m$. CS 中每个 Δ-簇的代表模式被选择构成了一个集合 $RS=\{R_1,R_2,\cdots,R_m\}$，其中 $R_i(1\leq i\leq m)$ 是 C_i 的代表模式. 再假设 T^* 是 F 中的一个大小为 K 的子集合，并且使 $IG(T)$ 最大化. S^* 是 RS 的一个大小为 K 的子集合，并且使 $IG(S^*)$ 最大化. 那么，$IG(T^*)-IG(S^*)\leq 2\Delta K(f(1)+f(N-1))$.

证明.　假设 C 是表示数据库 D 中图类别的随机变量. 类似于定理 8 中的证明过程，可得

$$|H(T^*)-H(S^*)|\leq\Delta K(f(1)+f(N-1)),$$

$$|H(C,T^*)-H(C,S^*)|\leq\Delta K(f(1)+f(N-1)).$$

因为 $IG(T^*)-IG(S^*)=(H(C)-H(C|T^*))-(H(C)-H(C|S^*))=H(C|S^*)-H(C|T^*)=(H(C,S^*)-H(S^*))-(H(C,T^*)-H(T^*))=(H(T^*)-H(S^*))-(H(C,T^*)-H(C,S^*))$，因此 $IG(T^*)-IG(S^*)\leq 2\Delta K(f(1)+f(N-1))$.

证毕.

4.2.2　Cluster-Top$K$ 算法的关键技术

如果先得到完整的频繁图模式集合 F，再使用传统的聚类算法(例如 k-means 算法)对 F 进行聚类，时间复杂性将是 $O(|F|^2)$，这显然是不可行的. 为了使 Cluster-Top$K$ 算法高效可扩展，Cluster-

TopK 算法要实现如下两个目标:(1)只扫描频繁子图挖掘算法输出的图模式一遍而能得到一个代表模式集合;(2)裁剪图模式空间中不产生(或极少产生)代表模式的那些分枝.下面两个小节分别介绍实现这两个目标的关键技术.

4.2.2.1 产生代表模式

图 3 显示了图模式空间中的一个分枝,每个节点代表一个频繁子图.大部分频繁子图挖掘算法(gSpan[6],FFSM[7]等)都采用深度优先方式访问该空间中的每个节点.采用深度优先方式将会访问每个节点两次:(1)第 1 次是从父亲节点到当前节点的访问,例如在图 3 中从节点 A 到节点 C 的访问.(2)第 2 次是在完成了所有后裔的访问后再次回到当前节点,例如在图 3 中访问完节点 E、F 和 G 之后第 2 次访问节点 C.目前的频繁子图挖掘算法是在第 1 次访问某个节点时输出它.因此,对图 3 中的节点输出顺序是…,A,B,C,E,F,G,D,….

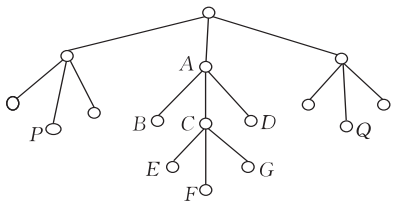


图 3 图模式空间中的一个分枝

为方便描述,我们将 δ-覆盖或 Δ-覆盖都简称为覆盖.我们的目标就是产生一个代表模式集合能覆盖所有的频繁图模式.对一个给定的图模式,例如图 3 中的节点 A,根据定义,能覆盖节点 A 的图模式必然是节点 A 的超图.因此,图 3 中的节点 P,Q 和 A 的后裔都有可能覆盖节点 A.

然而,我们在文献[17]中证明了,如果采用 gSpan[6] 的枚举框架,在一个节点第 2 次访问之后所有能覆盖该节点的图模式都已经被输出.例如:采用 gSpan 的枚举框架,节点 Q 就不能覆盖节点 A.

因此,我们调整图模式空间中节点的输出顺序:只有当一个节点第 2 次被访问时,我们才输出它.这样一来就可以保证当一个图模式 P 输出时,所有能覆盖 P 的图模式都已经被输出,便于我们为 P 选择对应的代表模式.对图 3 中的节点,调整之后的输出顺序将是…,B,E,F,G,C,D,A,….

采用上面调整之后的输出顺序,我们依此处理每个输出的图模式.假定 RS 表示已经产生的部分代表模式集合.如果当前输出的图模式 P 能被 RS 中的某个代表模式覆盖,我们继续处理下一个输出的图模式.如果 P 不能被 RS 中的任何代表模式覆盖,我们创建一个新的能覆盖 P 的代表模式.

为了使最终产生的代表模式数量尽可能少,在创建新的能覆盖 P 的代表模式时,我们采用了贪心策略,选择 P 的后裔中能覆盖 P 的最大图模式.因为这样选择的图模式有更大的可能性来覆盖以后输出的图模式.假设需要为图 3 中的节点 A 创新一个新的代表模式并且已知节点 B,C,F,D 都能覆盖 A,根据贪心策略,我们应创建一个新的代表模式 F,因为节点 F 大于节点 B,C,D.

4.2.2.2 裁剪不产生代表模式的分枝

如果图模式搜索空间中的某个分枝不产生新的代表模式(或者极少产生新的代表模式),完全遍历该分枝会使算法的性能急剧下降.本小节研究如何有效地裁剪这样的分枝.

图 4 显示了整个图模式空间中以图模式 g 为根的一棵子树.通过向图模式 g 中增加一条新边,g 可以被扩展成一系列新的图模式 $g\diamond e_1$,$g\diamond e_2$,…,$g\diamond e_n$.以 $g\diamond e_1$ 为根的分枝含有 $g\diamond e_1$ 的超图.以 $g\diamond e_2$ 为根的分枝含有 $g\diamond e_2$ 的超图,但不含有 $g\diamond e_1$ 的超图.同样地,以 $g\diamond e_i$ 为根的分枝含有 $g\diamond e_i$ 的超图,但不含有任何 $g\diamond e_j(j<i)$ 的超图.

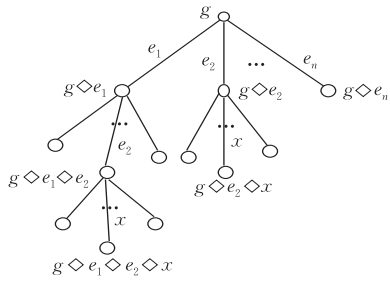


图 4 图模式空间中以 g 为根的一棵子树

对以 $g\diamond e_2$ 为根的分枝中任何图模式 $g\diamond e_2\diamond x$,很可能存在 $g\diamond e_2\diamond x$ 的超图 $g\diamond e_1\diamond e_2\diamond x$,并且 $g\diamond e_1\diamond e_2\diamond x$ 出现在 $g\diamond e_1$ 为根的分枝里.如果在图数据库中,g 和 $g\diamond e_1$ 经常一起出现,$g\diamond e_2\diamond x$ 和 $g\diamond e_1\diamond e_2\diamond x$ 有很大的可能性也经常一起出现.这也就意味着 $g\diamond e_2\diamond x$ 和 $g\diamond e_1\diamond e_2\diamond x$ 在支持度上非常接近.

假设以 $g\diamond e_1$ 为根的分枝已经被遍历.那么,在遍历以 $g\diamond e_2$ 为根的分枝之前,已经产生了一个代表模式 R 能覆盖 $g\diamond e_1\diamond e_2\diamond x$.因为 R 能覆盖 $g\diamond e_1\diamond e_2\diamond x$,根据覆盖定义,$R$ 是 $g\diamond e_1\diamond e_2\diamond x$ 的超图.如果 g 和 $g\diamond e_1$ 经常一起出现,则很可能 $g\diamond e_2\diamond x$ 和 $g\diamond e_1\diamond e_2\diamond x$ 也经常一起出现,即,$supp(g\diamond$

$e_1)\approx supp(g)$，$supp(g\diamond e_1\diamond e_2\diamond x)\approx supp(g\diamond e_2\diamond x)$. 那么，$supp(g\diamond e_2\diamond x)-supp(R)\approx supp(g\diamond e_1\diamond e_2\diamond x)-supp(R)$. 根据覆盖定义($\delta$-覆盖或者$\Delta$-覆盖)，$R$ 具有很大的可能性覆盖 $g\diamond e_2\diamond x$. 因此，如果 g 和 $g\diamond e_1$ 经常一起出现，以 $g\diamond e_2$ 为根的分枝不产生(或者很少产生)新的代表模式，我们可以略过该分枝的遍历，以提高算法的效率. 类似地，如果存在一条边 e_i 使得 g 和 $g\diamond e_i$ 经常一起出现，我们就可以略过那些以 $g\diamond e_j(j<i)$ 为根的分枝，因为这些分枝几乎不产生新的代表模式.

现在需要解决的一个问题是怎样度量 g 和 $g\diamond e$ 是否经常一起出现？为此，我们根据 g 和 $g\diamond e$ 的支持度定义了它们之间的一个距离函数 $D_{supp}(g,g\diamond e)=1-\frac{supp(g\diamond e)}{supp(g)}$. 根据用户说明给出一个小的距离阈值参数 $\varepsilon(0<\varepsilon<1)$. 如果 $D_{supp}(g,g\diamond e)<\varepsilon$，就可以认为 g 和 $g\diamond e$ 经常一起出现. 另一种更精确的方法是考虑 g 和 $g\diamond e$ 在数据库中的嵌入数(所有子图同构次数之和). 设 $embeeding(g)$ 表示 g 在数据库中的嵌入次数. 根据 g 和 $g\diamond e$ 的嵌入数可以定义它们之间的另一个距离函数 $D_{emb}(g,g\diamond e)=1-\frac{embedding(g\diamond e)}{embedding(g)}$. 如果 $D_{emb}(g,g\diamond e)<\varepsilon$，就可以认为 g 和 $g\diamond e$ 经常一起出现.

请注意，尽管使用距离阈值参数 ε 可以略过很多搜索分枝，我们也不可避免地会丢失一些代表模式. ε 值越大，丢失的数量越多. 然而，在实验中，我们发现使用一个很小的 ε 值(例如 0.01)，就可以得到 98%以上的代表模式，同时使算法的性能提高近 2 个数量级.

下面分析丢失少量代表模式对挖掘结果质量的影响. 假设一个代表模式 R 丢失了，R 覆盖的某个频繁图模式 P 能有另一个代表模式 R_1 所覆盖，R_1 没有丢失. 因为 R 能覆盖 P，说明 R 和 P 在结构和支持度上很接近. 同样，因为 R_1 能覆盖 P，说明 R_1 和 P 在结构和支持度上很接近. 因此，R 和 R_1 在结构和支持度上也会很接近. R 对意义度量的贡献能由 R_1 近似代替. 因为我们的目标是选择 K 个图模式联合起来使某一意义度量最大化，丢失的少量代表模式对意义度量的贡献能由其它的代表模式近似地代替. 因此，挖掘结果质量不受什么影响. 实验中，我们发现不同的 ε 值对挖掘结果质量影响非常小，而 ε 值对改进算法效率却起着巨大的作用.

4.2.3　Cluster-Top$K$ 算法描述

本节将 4.2.2 节中的关键技术集成到 gSpan[6] 的 DFS 编码搜索框架中，给出完整算法 Cluster-Top$K$(见算法 2). 在 gSpan 中，每个频繁子图都对应一个最小 DFS 编码(边的序列). DFS 编码搜索框架采用深度优先搜索方法挖掘频繁子图，只在最小 DFS 编码上进行最右扩展. 因为 gSpan 是经典的频繁子图挖掘算法，故省略了它的细节描述. DFS 编码、最右扩展等具体概念请见文献[6].

算法 2.　Cluster-Top$K$.

输入：图数据库 D；最小支持度阈值 min_sup；意义度量 M；输出模式个数 K；聚类质量参数 δ(或 Δ)；距离阈值 ε

输出：使 M 最大化的 K 个图模式

1. 扫描 D 得到所有频繁边；
2. 删除 D 中不频繁的边和结点；
3. $S^1=${所有频繁边的最小 DFS 编码}；
   //S^1 中的 DFS 编码按 DFS 字典顺序排序
4. $GS=\varnothing$；//全局栈
5. $RS=\varnothing$；//存放代表模式的全局数据结构
6. For S^1 中的每个 DFS 编码 s
7. $s.min\_distance=1$；//初始化最大值
8. Call MiningReprePatterns(s, $NULL$, D, min_sup, $\delta(\Delta)$, ε)；
9. 类似于算法 1 中的步 2)～步 7)，从 RS 中贪心增量地选择 K 个图模式使 M 最大化.

算法 Cluster-Top$K$ 如下工作：初始化时，先扫描数据库，得到频繁边集合，然后删除数据库中不频繁的边和结点. 在这之后，对每个 1-边频繁子图，算法调用子过程 MiningReprePatterns 进行深度优先搜索，发现所有代表模式. 在得到所有代表模式的集合 RS 之后，类似于算法 1，从 RS 中贪心增量地选择 K 个图模式.

子过程(MiningReprePatterns).

输入：DFS 编码 s；s 的父亲编码 p；图数据库 D；最小支持度阈值 min_sup；聚类质量参数 δ(或 Δ)；距离阈值 ε

输出：代表模式集合 RS

1. If $p!=$NULL & $p.min\_distance<\varepsilon$, Then
2.　　子过程结束；
3. If $s\neq\min(s)$, Then
4.　　子过程结束；
5. $p.min\_distance=\min(p.min\_distance, D_{supp}(p,s))$；
6. 把 s 的最后一条边放入全局堆栈 GS；
7. 对 GS 中的每个入口(频繁图模式)Q
8.　　根据贪心策略，如果 s 大于 Q 的候选代表模式

$Q.R$,则用 s 替换 $Q.R$;
9. 扫描 D 一次,发现 s 的所有频繁最右扩展孩子;
10. 对 s 的每个频繁最右扩展孩子 $s\diamond_r e$
11. $(s\diamond_r e).min\_distance=1$;//初始化最大值
12. Call MiningReprePatterns($s\diamond_r e, s, D, min\_sup, \delta(\Delta), \varepsilon$);
13. If $GS[top].covered=$False, Then
14. 在 RS 中找一个代表模式 R 能覆盖 s;
15. If RS 中不存在这样的代表模式 R, Then
16. 用 s 的候选代表模式创建一个新的代表 R_{new},放入 RS;
17. 对 GS 中的每个入口(频繁图模式)Q
18. If Q 能被 R_{new} 覆盖, Then
19. $GS[Q].covered=$True;
Else
20. 对 GS 中的每个入口(频繁图模式)Q
21. If Q 能被 R 覆盖, Then
22. $GS[Q].covered=$True;
23. 弹出 GS 的栈顶 $GS[top]$.

在子过程 MiningReprePatterns 的第 1 行,我们测试以当前模式 s 为根的分枝是否能被裁剪. 我们使用 $p.min\_distance$ 表示 p 和它的已经被遍历的孩子之间的最小距离(D_{supp} 或 D_{emb}). 如果 $p.min\_distance<\varepsilon$,说明存在 p 的一个已经被遍历的孩子 c,c 和 p 经常一起出现. 根据 4.2.2.2 节描述的思想,因为当前代表模式集合 RS 能覆盖以 c 为根分枝中的所有频繁图模式,RS 也就具有很大的可能性可以覆盖以 s 为根分枝中的所有频繁图模式. 因此,我们可以跳过以 s 为根的分枝(裁剪该子树). 在第 3 行,$s\neq\min(s)$ 判断 s 是否是当前模式的最小 DFS 编码. 如果不是,可以跳过以 s 为根的分枝[6]. 在第 5 行,我们根据 p 和 s 的之间的距离 $D_{supp}(p,s)$(或 $D_{emb}(p,s)$)更新 $p.min\_distance$. 在第 6 行,我们将当前模式 s 的 DFS 编码中的最后一条边放入全局栈 GS. GS 跟踪图模式空间中从根节点到当前节点(当前模式 s)的所有图模式. GS 中的每个入口对应一个图模式 Q,含有如下信息:(1) Q 的 DFS 编码中的最后一条边;(2) Q 的支持度;(3) 覆盖标记 $covered$;(4) Q 的候选代表模式 $Q.R$. 在第 7 行,算法扫描 GS 中的每个图模式 Q,测试当前模式 s 是否能覆盖 Q. 如果 s 能覆盖 Q 并且 s 大于 Q 的候选代表模式 $Q.R$,根据 4.2.2.1 节中的贪心策略,用 s 替换 $Q.R$. 第 9 行,算法扫描数据库一次,发现当前模式 s 的所有频繁最右扩展孩子. 第 10 行,对 s 的每个频繁最右扩展孩子 $s\diamond_r e$,$(s\diamond_r e).min\_distance$ 被初始化为最大值 1,算法递归调用子过程 MiningReprePatterns 继续深度优先搜索. 随着对 $s\diamond_r e$ 的孩子的遍历,$(s\diamond_r e).min\_distance$ 的值被逐渐减小. 当 $(s\diamond_r e).min\_distance$ 小于距离阈值 ε 时,根据 4.2.2.2 节中的思想,$s\diamond_r e$ 的没被遍历的孩子分枝就可以被裁剪掉. 第 13 行,在遍历当前模式 s 的所有后裔之后,算法测试 s 是否已经被覆盖. 如果 s 没被覆盖,算法在第 14 行扫描代表模式集合 RS,试图发现一个代表模式 R 能覆盖 s. 如果这样的代表 R 不存在,算法用 s 的候选代表模式创建一个新的代表模式 R_{new},把 R_{new} 放入 RS 中(第 16 行),并且算法扫描 GS 中的每个图模式 Q(第 17 行),判断 R_{new} 是否能覆盖 Q(第 18 行). 如果 R_{new} 能覆盖 Q,则标记 Q 被覆盖(第 19 行). 如果在 RS 中发现了某个代表模式 R 能覆盖 s,也扫描 GS 中的每个图模式 Q(第 20 行),判断 R 是否能覆盖 Q(第 21 行). 如果 R 能覆盖 Q,则标记 Q 被覆盖(第 22 行).

5 实验结果及分析

我们进行了大量的实验来考察算法的挖掘结果质量、执行效率、可扩展性以及不同参数对算法性能和结果质量的影响.

5.1 实验环境

我们从一个真实的化合物集合中导出了若干个图集合. 该化合物集合可从下面的网址获得:http://dtp.nci.nih.gov/docs/3d_database/structural_information/structural_data.html. 该化合物集合是用来测试化合物对艾滋病病毒(AIDS)的抑制作用,含有大约44000 个化合物. 根据实验结果每个化合物都被分为下面三类中的一类:CA(confirmed active)、CM(confirmed moderately)和 CI(confirmed inactive). 其中,CA 含有 422 个化合物,CM 含有 1081 个化合物,CI 含有剩余的化合物. 我们根据类别,分别导出了 3 个图集合 CA、CM 和 CI. 其中,CI 图集合是从所有 CI 化合物中随机选择 5000 个化合物.

本文算法使用 C++语言实现,用带有-O3 优化选项的 g++编译. 用于实验的计算机具有 PIV 3.0GHz CPU 和 1GB 内存,运行 RedHat Linux 8.0 操作系统.

Cluster-Top$K$ 算法需要两个额外的参数:聚类质量参数 δ 和距离阈值 ε. 在下面的实验中,若无特别说明,δ 取 0.1,ε 取 0.01. 参数 δ 和 ε 对算法的影响在 5.4 节描述.

5.2　比较挖掘结果的质量

本节比较算法 Greedy-Top$K$ 和 Cluster-Top$K$ 之间的挖掘结果质量，下节比较算法 Greedy-Top$K$ 和 Cluster-Top$K$ 之间的效率. 表 1 显示了在不同的数据集合上，固定最小支持度 min_sup 等于 10%，变化不同的 K 值，算法 Greedy-Top$K$ 和 Cluster-Top$K$ 选择的 K 个图模式而得到的联合熵大小(MES 问题). 表 2 显示了在相同条件下，算法选择的 K 个图模式而得到的信息增益大小(MIGS 问题).

可以看出，算法 Greedy-Top$K$ 和 Cluster-Top$K$ 挖掘出的 Top-K 模式在质量上非常接近. 尽管 Cluster-Top$K$ 算法没有理论上的近似比保证，而它输出的意义度量值(联合熵和信息增益)与 Greedy-Top$K$ 算法相比最多相差 1%. 有时侯，Cluster-Top$K$ 算法得到的结果还稍微优于 Greedy-Top$K$ 算法. 我们也将最小支持度的取值在 5%～30%之间变化，用以比较结果差异. 结果显示，Greedy-Top$K$ 和 Cluster-Top$K$ 给出的 Top-K 模式在质量上仍然非常接近.

表 1　比较 Greedy-Top$K$ 和 Cluster-Top$K$ 计算的联合熵(min_sup=10%)

K	对 CA 类的联合熵		对 CM 类的联合熵		对 CI 类的联合熵	
	Greedy-Top$K$	Cluster-Top$K$	Greedy-Top$K$	Cluster-Top$K$	Greedy-Top$K$	Cluster-Top$K$
5	4.53327	4.56331	4.63226	4.60691	4.63527	4.73849
10	6.56251	6.59219	7.71009	7.69338	7.93517	7.90342
15	7.31281	7.24208	8.84297	8.80495	9.73596	9.67808
20	7.67565	7.60308	9.23750	9.21460	10.58520	10.49630

表 2　比较 Greedy-Top$K$ 和 Cluster-Top$K$ 计算的信息增益(min_sup=10%)

K	对 CA(参照 CM)的信息增益		对 CM(参照 CA)的信息增益		对 CI(参照 CA)的信息增益	
	Greedy-Top$K$	Cluster-Top$K$	Greedy-Top$K$	Cluster-Top$K$	Greedy-Top$K$	Cluster-Top$K$
5	0.193026	0.187017	0.167959	0.165682	0.128673	0.128654
10	0.316993	0.304763	0.300546	0.315074	0.231147	0.230494
15	0.471530	0.453959	0.462120	0.476579	0.303742	0.296234
20	0.584803	0.557849	0.565636	0.573212	0.341270	0.336603

为了进一步验证不同算法的结果质量，我们使用算法 Greedy-Top$K$ 和 Cluster-Top$K$ 最大化信息增益产生的 Top-K 模式进行分类实验，比较分类性能. 我们构造两个分类任务：(1) 对 CA 类和 CM 类中的化合物进行分类；(2) 对 CA 类和 CI 类中的化合物进行分类. 带有缺省参数的 LIBSVM① 被用作分类模型. 分类准确率使用 5 次交叉验证进行评价. ROC 曲线下的面积(AUC)被用来度量分类性能. AUC 越大，表示分类性能越好.

此外，我们也抽取了传统的 Top-K 模式(根据信息增益对图所有模式进行排序，从中选择前 K 个信息增益最大的图模式)进行比较. Trad-Top$K$ 表示抽取传统 Top-K 模式的算法.

我们使用算法 Trad-Top$K$、Greedy-Top$K$ 和 Cluster-Top$K$ 分别从正类和反类中抽取 K 个图模式，建立分类模型. 表 3 显示了 K 变化时，不同的 Top-K 模式构造的分类器的分类性能(AUC). 可以看出，Greedy-Top$K$ 和 Cluster-Top$K$ 产生的 Top-K 模式在分类性能上远远优于传统的 Top-K 模式. 因为传统的 Top-K 模式只考虑优化单一模式的意义度量，而没有考虑一个模式集合中所有模式的联合意义. 此外，Greedy-Top$K$ 和 Cluster-Top$K$ 产生的 Top-K 模式在分类性能方面也非常接近，再次说明 Greedy-Top$K$ 和 Cluster-Top$K$ 的挖掘结果质量差异很小.

表 3　不同 Top-K 模式构造的分类器的分类性能(AUC)(min_sup=10%)

K	对 CA(参照 CM)的分类性能($A \cup C$)			对 CA(参照 CI)的分类性能($A \cup C$)		
	Trad-Top$K$	Greedy-Top$K$	Cluster-Top$K$	Trad-Top$K$	Greedy-Top$K$	Cluster-Top$K$
5	0.6032	0.7530	0.7388	0.7222	0.8284	0.8264
10	0.6608	0.7966	0.8013	0.5699	0.9074	0.9020
15	0.6847	0.8030	0.7997	0.6161	0.9216	0.9255
20	0.7179	0.8023	0.7958	0.7635	0.9244	0.9232

最后，我们从结构上比较一下不同算法产生的 Top-K 模式. 图 5、6 和 7 分别显示了算法 Trad-Top$K$、Greedy-Top$K$ 和 Cluster-Top$K$ 从 CA(参照 CM)数据集上挖掘的 Top-5 模式. 可以看出，Trad-Top$K$ 挖掘的 Top-5 图模式在结构上有很大重叠，因为它只考虑单一模式的意义，使得挖掘出来的图

① Chang C, Lin C. LIBSVM: A library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm

模式存在很大的相关性. Greedy-Top$K$ 和 Cluster-Top$K$ 考虑了模式的联合意义,挖掘的 Top-5 图模式在结构上彼此之间有很大的差异. 在实际应用中,这恰恰是用户想要挖掘的模式类型.

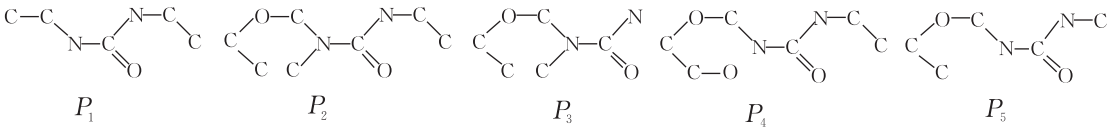


图 5 Trad-Top$K$ 算法产生的 Top-5 模式(MIGS,min_sup=10%)

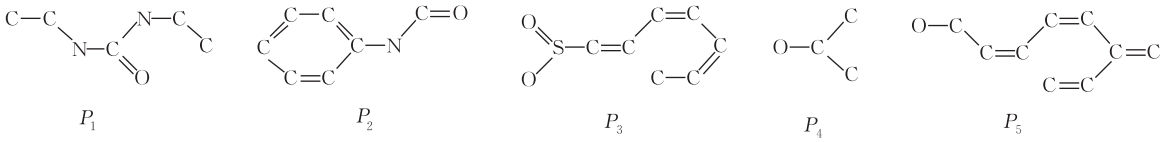


图 6 Greedy-Top$K$ 算法产生的 Top-5 模式(MIGS,min_sup=10%)

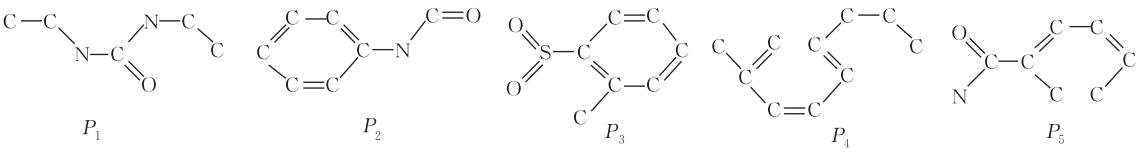


图 7 Cluster-Top$K$ 算法产生的 Top-5 模式(MIGS,min_sup=10%)

5.3 比较算法的效率

本节比较算法 Greedy-Top$K$ 和 Cluster-Top$K$ 的执行效率. 在 Greedy-Top$K$ 的第 1 步,既可以使用 gSpan[6] 挖掘频繁图模式,又可以使用 CloseGraph[9] 挖掘频繁闭图模式. 我们形成了 Greedy-Top$K$ 的两个版本,gSpan + Greedy-Top$K$ 和 CloseGraph + Greedy-Top$K$. 注意:在与 Cluster-Top$K$ 比较时,Greedy-Top$K$ 使用了 4.1.2 节中的裁剪技术以获得最高的执行效率.

图 8 显示了当最小支持度变化时,求解 MES 问题不同算法所用的时间. 图 9 显示了当最小支持度变化时,求解MIGS问题不同算法所用的时间. 如果一

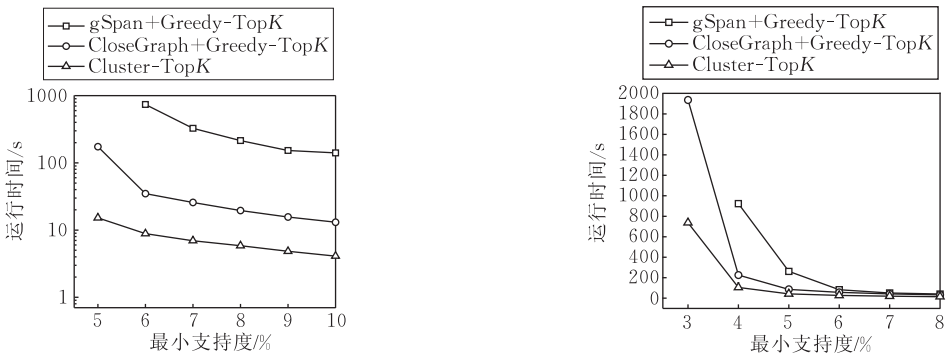


图 8 支持度的变化对算法执行时间的影响(MES,K=10)

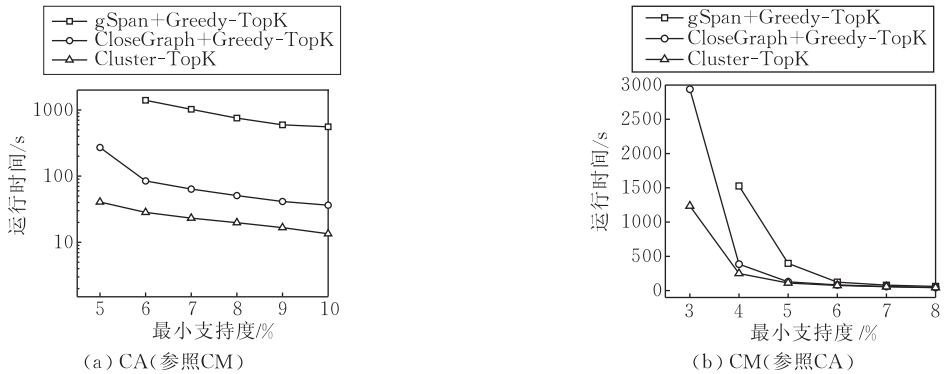
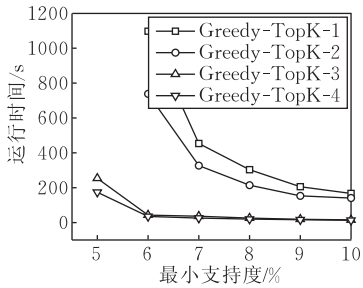


图 9 支持度的变化对算法执行时间的影响(MIGS,K=10)

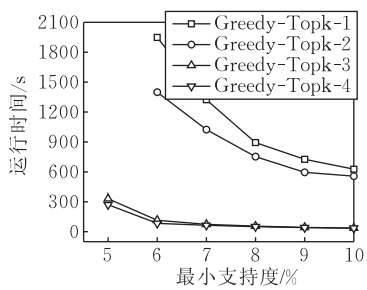
个算法不能在一个小时内完成，我们就终止算法.因此，图中 gSpan＋Greedy-Top$K$ 的结果是不完整的.

从图 8 和图 9 可以看出，在执行效率方面，算法 Cluster-Top$K$ 极大地优于 Greedy-Top$K$.例如：在 CA 数据集合上，Cluster-Top$K$ 比 gSpan＋Greedy-Top$K$ 快 2 个数量级，比 CloseGrpah＋Greedy-Top$K$ 快 1 个数量级.而且，支持度越低，Cluster-Top$K$ 和 Greedy-Top$K$ 的执行效率差异越大.Cluster-Top$K$ 的高效率来源于两个方面：(1) 在挖掘代表模式时，因为 Cluster-Top$K$ 裁剪掉了很多不产生(或少产生)代表模式的分枝，与闭图模式挖掘算法 CloseGraph 相比，Cluster-Top$K$ 遍历了图模式空间中更少的结点，获得了更高的执行效率.(2) 在贪心选择时，因为 Cluster-Top$K$ 产生的代表图模式数量比 CloseGraph 产生的闭图模式少很多，因此 Cluster-Top$K$ 贪心选择的执行效率也比 Greedy-Top$K$ 贪心选择的执行效率快很多.

下面评价算法 Greedy-Top$K$ 中使用的裁剪技术(见 4.1.2 节)的有效性.为此，我们形成了 Greedy-Top$K$ 算法的 4 个变体.Greedy-Top$K$-1 表示不使用裁剪技术的 gSpan ＋ Greedy-Top$K$，Greedy-Top$K$-2 表示使用裁剪技术的 gSpan＋Greedy-Top$K$，Greedy-Top$K$-3 表示不使用裁剪技术的 CloseGraph＋Greedy-Top$K$，Greedy-Top$K$-4 表示使用裁剪技术的 CloseGraph＋Greedy-Top$K$.图 10 显示了当最小支持度变化时，算法 Greedy-Top$K$ 的不同变体所需的执行时间.可以看出，4.1.2 节的裁剪技术能有效地改善算法 Greedy-Top$K$ 的效率.支持度越低，改善越明显.



(a) *MES*



(b) *MIGS*

图 10　支持度变化时，算法 Greedy-Top$K$ 的不同变体所需的执行时间(CA 数据集，K=10)

5.4　不同参数对算法的影响

本节评价算法 Cluster-Top$K$ 中所使用的两个参数：聚类质量参数 δ 和距离阈值 ε 对算法的挖掘结果和运行时间的影响.由于聚类质量参数 δ(相对值)和 Δ(绝对值)具有完全相同的作用，我们只给出对 δ 的评价结果.

我们先评价聚类质量参数 δ 对算法的影响，固定距离阈值 ε=0.01，变化 δ 值.Cluster-Top$K$ 算法的运行时间由两部分构成：聚类时间和贪心选择时间.我们使用 T_{Cluster}，T_{Greedy} 和 T_{Total} 分别表示算法的聚类时间、贪心选择时间和总的时间.*Rep* 表示 Cluster-Top$K$ 聚类之后产生的代表模式数量，M 表示最后输出的度量值.表 4 显示了当 δ 值变化时，算法 Cluster-Top$K$ 的结果质量和运行时间.可以看出，当 δ 增加时，聚类时间 T_{Cluster} 不受影响.然而，随着 δ 值的增加，输出的代表模式数量逐渐减少，因此，贪心选择时间 T_{Greedy} 逐渐减小，总的时间也逐渐减小.一个有趣的现象是 δ 值对最后度量值 M 的影响微乎其微.在 4.2.1.1 节，我们分析了 Cluster-Top$K$ 的结果与 Greedy-Top$K$ 的结果在理论上的最大可能差异.而实际中的差异比这种理论上的最大差异少的多得多.这再次说明了算法 Cluster-Top$K$ 采用的聚类策略非常适用于本文的研究问题.

表 4　聚类质量参数 δ 对算法 Cluster-Top$K$ 的挖掘结果和运行时间的影响

δ	*Rep*	CA, *min_sup*=5%							
		Joint Entropy (MES)				Information gain (MIGS)(res. CM)			
		M	$T_{Cluster}$/s	T_{Greedy}/s	T_{Total}/s	M	$T_{Cluster}$/s	T_{Greedy}/s	T_{Total}/s
0.05	976	6.57483	5.91	11.66	17.57	0.297455	5.92	45.51	51.43
0.1	760	6.59219	5.90	9.14	15.04	0.303155	5.91	34.85	40.76
0.15	631	6.49119	5.89	7.78	13.7	0.301489	5.90	29.54	35.44
0.2	551	6.4912	5.91	7.04	12.95	0.301288	5.89	26.55	32.44
0.25	489	6.4464	5.90	6.33	12.23	0.284938	5.91	23.83	29.74
0.3	453	6.45905	5.92	5.85	11.77	0.291065	5.90	21.81	27.71

下面评价距离阈值 ε 对算法的影响，固定聚类质量参数 $\delta=0.1$，变化 ε 的值. 表 5 显示了当 ε 值变化时，算法 Cluster-Top$K$ 的结果质量和运行时间. 可以看出，当 ε 增加时，聚类时间 T_{Cluster} 明显减少. 这是因为大的 ε 值使得图模式空间中更多的分枝被裁剪掉，从而使得聚类时间 T_{Cluster} 被大大缩减. 当 ε 增加时，由于输出的代表模式数量也在逐渐减少，因此贪心选择时间 T_{Greedy} 和总的时间也逐渐减小. 与聚类质量参数 δ 相比，距离阈值 ε 对结果质量的影响更小. 例如：当 ε 从 0.01～0.1 变化时，最后的度量值几乎不发生变化. 在前面的实验中我们只使用了小的 ε 值 0.01，就能使算法 Cluster-Top$K$ 比 CloseGraph＋Greedy-Top$K$ 快 1 个数量级别. 如果使用更大的 ε 值（例如：0.1）能使算法 Cluster-Top$K$ 更快，而且对结果质量几乎没有影响. 这恰恰说明了算法 Cluster-Top$K$ 所采用的裁剪策略能在保证结果质量的前提下，极大地改善算法的效率.

表 5 距离阈值 ε 对算法 Cluster-Top$K$ 的挖掘结果和运行时间的影响（min_sup＝5%）

ε	代表模式数	对 *CA* 的结果							
		联合熵（*MES*）				信息增益（*MIGS*）（参照 *CM*）			
		M	$T_{Cluster}/s$	T_{Greedy}/s	T_{Total}/s	M	$T_{Cluster}/s$	T_{Greedy}/s	T_{Total}/s
0.01	760	6.59219	5.9	9.31	15.21	0.303155	5.9	34.71	40.61
0.02	731	6.59219	5.36	9.01	14.37	0.303155	5.36	33.78	39.14
0.03	697	6.59219	4.49	8.14	12.63	0.303155	4.49	31.2	35.69
0.04	675	6.59219	3.93	7.98	11.91	0.303155	3.93	30.13	34.06
0.05	644	6.59219	3.13	7.08	10.21	0.303155	3.12	27.05	30.17
0.06	633	6.59219	2.93	6.95	9.88	0.303155	2.93	26.71	29.64
0.07	626	6.59219	2.9	6.93	9.83	0.303155	2.9	26.63	29.53
0.08	616	6.59219	2.71	6.78	9.49	0.303155	2.71	26.15	28.86
0.09	612	6.59219	2.59	6.72	9.31	0.300521	2.59	25.89	28.48
0.10	574	6.59219	2.44	6.45	8.89	0.300521	2.44	24.53	26.97
0.15	465	6.59896	1.83	5.14	6.97	0.292042	1.83	19.63	21.46
0.20	355	6.52253	1.31	4.23	5.54	0.284786	1.31	16.37	17.69

6 相关工作

很多频繁子图挖掘算法已经被提出，大致可分为两类. 第 1 类算法（例如 AGM[3] 和 FSG[4]）根据 Apriori 性质采用逐级搜索策略来枚举所有频繁子图. 第 2 类算法（例如 Mofa[5]、gSpan[6]、FFSM[7] 和 GASTON[8]）采用深度优先搜索策略来枚举所有频繁子图. 通常第 2 类算法比第 1 类算法有更好的内存利用率，因此具有更高的挖掘效率. 挖掘频繁子图经常会产生指数级数量的图模式. 为解决频繁子图挖掘时图模式数量“爆炸”问题，研究人员已经提出了两类主要方法：(1) 挖掘频繁闭图模式[9]；(2) 挖掘频繁最大图模式[10-11]. 第 1 类方法仍然会输出大量的图模式. 第 2 类方法会丢失一些重要的图模式. 最近，我们提出了一个折衷方法[17]，从图数据库中挖掘代表模式集合. 其它与图模式挖掘有关的研究还包括图模式并行挖掘算法[18]、挖掘频繁树模式[19]以及挖掘图产生器[20]等.

上面这些与图模式有关的挖掘方法除了支持度，都没有考虑其它的意义度量. 最近，文献[21]提出了一个框架，根据用户给定的意义度量，挖掘意义度量值最大的一个图模式. 然而，该框架只能输出一个图模式. 在实际应用中，用户经常需要多个图模式来管理和分析图数据. 传统的 Top-K 方法虽然可以输出多个图模式，但没有考虑模式之间的相关性，会输出结构上类似的图模式，无法满足用户对图模式集合多样化的需求.

与上述工作不同，本文研究如何挖掘 Top-K 图模式，联合起来使用户给定的意义度量最大化. 此外，本文提出的 Cluster-Top$K$ 算法使用了一个新的挖掘代表图模式的算法. 文献[17]中挖掘代表图模式最有效的算法 RP-GD 需要枚举所有频繁闭图模式，因此 RP-GD 挖掘效率明显低于 CloseGraph 算法[9]. 本文中挖掘代表图模式的新算法比 CloseGraph 快一个数量级，因此也能比 RP-GD 快一个数量级. 而且，文献[17]挖掘代表图模式的目的只是用来近似概括频繁图模式，本文挖掘代表图模式的目的则是用来选择使联合意义度量最大的 Top-K 图模式.

7 结 论

本文提出了基于联合意义度量的 Top-K 图模

式挖掘问题，并给出两个高效算法 Greedy-Top$K$ 和 Cluster-Top$K$. 实验结果显示本文提出的 Top-K 挖掘优于传统的 Top-K 挖掘. Cluster-Top$K$ 比 Greedy-Top$K$ 快至少一个数量级. 而且，Cluster-Top$K$ 的挖掘结果非常类似于 Greedy-Top$K$ 的挖掘结果. 本文提出的算法是一种通用算法，也适用于其它的联合意义度量和模式类型（例如：项集模式、序列模式、树模式等）.

参 考 文 献

[1] Yan X, Cheng H, Han J, Yu PS. Graph indexing: A frequent structure-based approach//Proceedings of the ACM SIGMOD Conference on Management of Data. Paris, France, 2004: 335-346

[2] Deshpande M, Kuramochi M, Wale N, Karypis G. Frequent substructure-based approaches for classifying chemical compounds. IEEE Transaction on Knowledge and Data Engineering, 2005, 17(8): 1036-1050

[3] Inokuchi A, Washio T, Motoda H. An apriori-based algorithm for mining frequent substructures from graph data//Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery. Lyon, France, 2000: 13-23

[4] Kuramochi M, Karypis G. Frequent subgraph discovery//Proceedings of the 1st IEEE Internatonal Conference on Data Mining. San Jose, USA, 2001: 313-320

[5] Borgelt C, Berhold M R. Mining molecular fragments: Finding relevant substructures of molecules//Proceedings of the 2nd IEEE Internatonal Conference on Data Mining. Maebashi City, Japan, 2002: 51-58

[6] Yan X, Han J. GSpan: Graph-based substructure pattern mining//Proceedings of the 2nd IEEE Internatonal Conference on Data Mining. Maebashi City, Japan, 2002: 721-724

[7] Huan J, Wang W, Prins J. Efficient mining of frequent subgraphs in the presence of isomorphism//Proceedings of the 3rd IEEE Internatonal Conference on Data Mining. Melbourne, USA, 2003: 549-552

[8] Nijssen S, Kok J N. A quickstart in frequent structure mining can make a difference//Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Seattle, USA, 2004: 647-652

[9] Yan X, Han J. Closegraph: Mining closed frequent graph patterns//Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Washington, USA, 2003: 286-295

[10] Huan J, Wang W, Prins J, Yang J. Spin: Mining maximal frequent subgraphs from graph databases//Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Seattle: USA, 2004: 581-586

[11] Thomas L T, Valluri S R, Karlapalem K. Margin: Maximal frequent subgraph mining//Proceedings of the 6th IEEE International Conference Data Mining. Hong Kong, China, 2006: 1097-1101

[12] Cook S A. The complexity of theorem-proving procedures//Proceedings of the 3rd ACM Symposium on Theory of Computing. Shaker Heights, USA, 1971: 151-158

[13] Cover T M, Thomas J A. Elements of Information Theory. New York: Wiley Interscience, 1991

[14] Tan P, Kumar V, Srivastava J. Selecting the right interestingness measure for association patterns//Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Edmonton, Canada, 2002: 32-41

[15] Nemhauser G, Wolsey L, Fisher M. An analysis of the approximations for maximizing submodular set functions. Mathematical Programming, 1978, 14(1): 265-294

[16] Zhang X, Pan F, Wang W, Nobel A. Mining non-redundant high order correlations in binary data//Proceeding of the 34th International Conference on Very Large Data Bases. Auckland, New Zealand, 2008: 1178-1188

[17] Liu Y, Li J, Gao H. Summarizing graph patterns//Proceedings of the 24th IEEE International Conference on Data Engineering. Cancun. Mexico, 2008: 903-912

[18] Chen Y K. Adaptive parallel graph mining for CMP architectures//Proceedings of the 6th IEEE International Conference on Data Mining. Hong Kong, China, 2006: 97-106

[19] Zaki M J. Efficiently mining frequent trees in a forest//Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Edmonton, Canada, 2002: 71-80

[20] Zeng Z, Wang J. Zhang J, Zhou L. FOGGER: An algorithm for graph generator discovery//Proceedings of the 12th International Conference on Extending Database Technology. Saint-Petersburg, Russia, 2009: 517-528

[21] Yan X, Cheng H, Han J, Yu PS. Mining significant graph patterns by scalable leap search//Proceedings of the ACM SIGMOD Conference on Management of Data. Vancouver, Canada, 2008: 335-346

LIU Yong, born in 1975, Ph. D. candidate. His research interests include graph mining and graph data management.

GAO Hong, born in 1966, professor, Ph. D. supervisor. Her research interests include data mining, data warehouse and sensor network.

LI Jian-Zhong, born in 1950, professor, Ph. D. supervisor. His research interests include data mining, data warehouse, sensor network, grid and bioinformatics.

Background

This work was supported by the National Grand Fundamental Research 973 Program of China (grand No. 2006CB303000), the Key Program of National Natural Science Foundation of China (grant No. 60533110), and the National Natural Science Foundation of China (grant No. 60773063). The projects are involved in the key technology of sensor network and graph data management. The research group has been working on many aspects of the Projects since 2006, and published many papers.

Mining significant graph patterns is an active research branch in the area of graph data management with broad applications in chemistry, bioinformatics, wireless sensor network, etc. Different from the traditional Top$K$ pattern mining, in this paper, the authors investigated a novel problem of mining Top$K$ graph patterns that jointly maximize some significance measure from graph databases. Two efficient algorithms, Greedy-Top$K$ and Cluster-Top$K$, are presented for mining Top$K$ graph patterns based on some joint significance measure. Extensive experimental study confirms that the mining results obtained by Greedy-Top$K$ and Cluster-Top$K$ are superior to those obtained by the traditional Top$K$ pattern mining.