反频繁集挖掘可计算复杂性问题研究

吕 品[1],陈年生[2],董武世[2]
(1.武汉工程大学 计算机科学与工程学院,湖北 武汉 430073;
2.湖北师范学院 计算机系,湖北 黄石 435002)

摘 要:频繁集挖掘是总结二进制数据的重要技术,但如何找到一个二进制数据集与频繁集挖掘结果相一致却十分困难。文中从可计算复杂度的观点研究了频繁集的隐私保持。特别分析了反频繁挖掘问题的可计算复杂度。给出了决定是否存在与一个已知频繁集兼容的数据集是一个NP难度问题;当原始数据集 d 由6个集合组成时计算与已知频繁集兼容的数据集的数量是一个P类完全问题。

关键词:反频繁集挖掘;隐私保持;投影
中图分类号:TP301.5 **文献标识码**:A **文章编号**:1005-3751(2006)04-0025-03

Study of Computational Complexity on Inverse Frequent Set Mining

LÜ Pin[1],CHEN Nian-sheng[2],DONG Wu-shi[2]
(1. School of Computer Science, Wuhan University of Technology, Wuhan 430073,China;
2. Department of Computer Science, Hubei Normal University, Huangshi 435002,China)

Abstract: Frequent set mining is a well-known technique to summarize binary data. However, it is difficult to find a binary data set that is compatible with frequent set mining results. The paper studies the frequent sets preserve privacy from the viewpoint of computational complexity, and specially analyzes the computational complexity of inverse frequent set mining. The paper forwards that deciding whether there is a data set compatible with the given frequent sets is NP-hard and computing the number of data sets compatible with the given frequent sets is P-hard even in the case when the original data set d consists of six sets.

Key words: inverse frequent set mining; preserve privacy; projection

0 引 言

数据挖掘最重要的技术就是分析数据并从已知的数据集中找到有趣的模式[1]。频繁集挖掘的主要目标就是从已知的一个序列集合中找到频繁性至少和预定义的最小支持度计数一样的项集。反频繁集挖掘的任务就是寻找是否存在一个数据集与已知项集支持度一致,而且其它项集的支持度小于已知项集的最小支持度。反频繁集挖掘对隐私可能有严重的威胁。在频繁集挖掘中存在一些隐私问题[2,3]。频繁集作为数据挖掘的结果可能是一个诚实者的所为,也可能被恶意者所利用。因此,数据挖掘最基本的关心就是保护用户的隐私不被泄露。文中的前提条件是不能从挖掘结果中找到原始的数据集。

1 反频繁集挖掘的定义

频繁集挖掘问题如下面所阐述:已知一个有限集 R,一个序列 $d = d_1 \cdots d_n$ 是 R 的子集,并且已知值 $\sigma \in [0,1]$,找出所有的被包含在区间 σ 上的子集 $d_i, 1 \leqslant i \leqslant n$,也就是在 $\mathrm{supp}(X,d) = |\{i : X \subseteq d_i, 1 \leqslant i \leqslant n\}|$ 中找频繁集的集合 $F(\sigma,d) = \{X \subseteq R : \mathrm{supp}(X,d) \geqslant \sigma_n\}$。$\mathrm{supp}(X,d)$ 定义为 X 在 d 中的支持度。集合 $\{X \subseteq R : X \notin F(\sigma,d)\} = 2^R \setminus F(\sigma,d)$ 称作频繁集。

文中重点讨论反频繁集挖掘问题。反频繁集挖掘问题可描述如下:已知有限集合 F,并且有限集合 $X \in F$,它们的支持度为 $\mathrm{supp}(X,F)$,反频繁集挖掘问题就是要在这样的条件下找一个数据集 d,它与集合 F 和 F 的支持度保持一致。例如,找一个 X 的子集序列 $d = d_1 \cdots d_n$,对于所有的 $X \in F$ 都有 $\mathrm{supp}(X,F) = \mathrm{supp}(X,d)$。

2 频繁集与投影

数据集 d 在集合 X 上的投影是一个序列 $\mathrm{pr}(X,d) = (d_1 \cap X) \cdots (d_n \cap X)$。在集合 $X \in F$ 之上投影 $\mathrm{pr}(X,d)$ 的集合可定义为:$\mathrm{pr}(F,d) = \{\mathrm{pr}(X,d) : X \in F\}$。最大频繁集 $M(\sigma,d)$ 由频繁集组成,但这些频繁集并不是频繁超集,也就是 $M(\sigma,d) = \{X \in F(\sigma,d) : X \subset Y \Rightarrow Y \notin F(\sigma,d)\}$。最大频繁集挖掘比频繁集挖掘效率更高[4,5]。而且,具有支持度 $X \in F(\sigma,d)$ 的最大频繁集 $F(\sigma,d)$ 包

收稿日期:2005-08-08
基金项目:湖北省自然科学基金资助项目(2004ADA023)
作者简介:吕 品(1973-),女,湖北鄂州人,硕士,讲师,研究方向为数据挖掘、算法分析与设计、软件工程;董武世,副教授,研究方向为数据库、高性能计算机网络。

含了与投影 $\mathrm{pr}(M(\sigma,d),d)$ 相同的信息。

频繁集 $F(\sigma,d)$ 和它的支持度可从投影 $\mathrm{pr}(M(\sigma,d),d)$ 计算得出，且反之也成立。因为，对于每个 $X\in F(\sigma,d)$ 和每个 $Y\supseteq X$，都有 $\mathrm{supp}(X,d)=|\{i:X\subseteq d_i\}|=|\{i:X\subseteq d_i\cap Y\}|=\mathrm{supp}(X,\mathrm{pr}(Y,d))$。同时由定义可知，每个集合 $X\in F(\sigma,d)$ 被包含在一些集合 $Y\in M(\sigma,d)$ 上的每一个集合中，因此，上面描述的第一个条件成立。反之，在最大集合 $X\in M(\sigma,d)$ 上的每个投影 $\mathrm{pr}(X,d)$ 也能从集合 $F=\{Y\in F:(\sigma,d):Y\subseteq X\}$ 中计算得出，且支持度 $\mathrm{supp}(X,F)=\mathrm{supp}(X,d)$，证明如下：

1) 如果集合 M 非空，找集合 $M=\{Y\in F:Y\subset Z\Rightarrow Z\notin F\}$；否则，停止。

2) 对于每个 $Y\in M$，把 Y 的 $\mathrm{supp}(Y,F)$ 的副本加入 $\mathrm{pr}(X,d)$，而且通过 $\mathrm{supp}(Y,F)$ 减少 Y 的每个子集的支持度 $\mathrm{supp}(Z,F)$。

3) 用 $\mathrm{supp}(Y,F)=0$ 删除 $Y\in F$；返回步骤 1。

$\mathrm{pr}(M,F)$ 的频繁集 F 决定了投影。在 $\mathrm{pr}(M,F)$ 中不同集合的数量可能是远远小于在集合 F 中的数量。因此，$\mathrm{pr}(M(\sigma,d),d)$ 可表示为频繁集 $F(\sigma,d)$ 的简化表示[6]。

由于投影比相应的频繁集更接近真实数据集，因此，可用同样的公式来表示反频繁集挖掘问题：假定投影 $\mathrm{pr}(M,F)$ 找到一个数据集 d，则 $\mathrm{pr}(M,F)=\mathrm{pr}(M,d)$。称这个过程为数据集重构问题。但是，有些投影的集合不能实现频繁集。所以，必须要确保投影能实现频繁集。对于一个已知的投影有简单的充要条件可以保证它能实现频繁集。这个充要条件是对于所有的 $1\leqslant i,j\leqslant m$，如果 $\mathrm{pr}(X_i\cap X_j,F_i)=\mathrm{pr}(X_i\cap X_j,F_j)$ 成立，则对于所有的 $1\leqslant i\leqslant m$ 和 $Y\subset X_i$，$\mathrm{supp}(Y,\mathrm{pr}(X_i,F_i))=\mathrm{supp}(Y,F)$ 成立。

3 反频繁集挖掘计算复杂度分析

定理 1 即使一个数据集由 6 个集合组成，寻找是否存在着与数据集 d 相一致的投影 $\mathrm{pr}(M,F)$ 是一个 NP 难度问题。

证明：通过图形 3 色问题的推理，可以得知上述问题是一个 NP 难度问题。图形 3 色问题的描述为：假设一个图 $G=(V,E)$，则能否找到一个质量好的 3 色 $c:\to V\{r,g,b\}$，即对于所有 $\{i,j\}\in E$ 有 $c(i)\neq c(j)$[7]。

假定属性的集合 R 为 $\{r_i,g_i,b_i:i\in V\}$，则可以按如下方法建立投影：对于每一条边 $\{i,j\}\in E$，定义一个投影 $\mathrm{pr}(\{r_i,g_i,b_i,r_j,g_j\},F)$ 为以下顺序：$\{r_i,g_j\}\{r_i,b_j\}\{g_i,r_j\}\{g_i,b_j\}\{b_i,r_j\}\{b_i,g_j\}$。如果不是 3 色图，则不存在数据集 d 与投影一致。对于图中的 3 种颜色，每一条边 $\{i,j\}$ 都具有相同颜色的两个顶点，但 $\{r_i,r_j\}$，$\{g_i,g_j\}$ 和 $\{b_i,b_j\}$ 中的任何一个并不是成对出现在投影 $\mathrm{pr}(\{r_i,g_i,b_i,r_j,g_j\},F)$。因此，寻找与投影兼容的数据集的解决方法应当全盘考虑。如果一个图是 3 色的，就一定存在着与投影兼容的数据集 d。在数据集 d 中的 6 个集合可以考虑成是 3 色 c 的 6 种置换，如一个 3 色 c 对于所有 $\{i,j\}\in E$ 有 $c(i)\neq c(j)$。

定理 2 寻找与已知的投影 $\mathrm{pr}(X_1,F_1)$ 和 $\mathrm{pr}(X_2,F_2)$ 兼容数据集 d 能在多项式时间内确定。

证明：依定义，投影 $\mathrm{pr}(X_1,F_1)$ 与一个数据集 d 相兼容，只有 $\mathrm{pr}(X_1,F_1)=\mathrm{pr}(X_1,d)$。投影 $\mathrm{pr}(X_2,F_2)$ 与数据集 d 兼容，只有 $\mathrm{pr}(X_2,F_2)=\mathrm{pr}(X_2,d)$。一个数据集 d 与两个投影都相兼容，只有当 $\mathrm{pr}(X_1\cap X_2,F_1)=\mathrm{pr}(X_1\cap X_2,d)=\mathrm{pr}(X_1\cap X_2,F_2)$，$\mathrm{pr}(X_1\setminus X_2,F_1)=\mathrm{pr}(X_1\setminus X,d)$ 和 $\mathrm{pr}(X_2\setminus X_1,F_1)=\mathrm{pr}(X_2\setminus X_1,d)$。通过简单地分类整理投影 $\mathrm{pr}(X_1,F_1)$ 和 $\mathrm{pr}(X_2,F_2)$，利用 $\mathrm{pr}(X_1\cap X_2,F_1)$ 和 $\mathrm{pr}(X_1\cap X_2,F_2)$，可以找出数据集 d 与投影 $\mathrm{pr}(X_1,d)$ 和 $\mathrm{pr}(X_2,d)$ 相兼容。通过投影 $\mathrm{pr}(X_1\cap X_2,F_1)$ 和 $\mathrm{pr}(X_1\cap X_2,F_2)$，这个计算可在时间复杂度 $O(|X_1\cap X_2|n)$ 内完成[8]。与 d 兼容的数据集的数量可以通过在投影 $\mathrm{pr}(X_1\cap X_2,d)$，$\mathrm{pr}(X_1,d)$ 和 $\mathrm{pr}(X_2,d)$ 中的不同集合的计数计算得出[9]。

定理 3 可以在多项式内确定一个数据集 d 和已知的投影 $\mathrm{pr}(X_1,F_1)$，$\mathrm{pr}(X_2,F_2)$ 和 $\mathrm{pr}(X_3,F_3)$ 兼容。

证明：首先，建立一个数据集 d' 与 $\mathrm{pr}(X_1,F_1)$，$\mathrm{pr}(X_2,F_2)$ 和 $\mathrm{pr}(X_3\setminus(X_1\setminus X_2),F_3)$ 相兼容。根据数据集 d'，建立一个二分图 $G=(V_1,V_2,E)$，这可以使 d' 与 $\mathrm{pr}(X_3\cap X_1,F_3)$ 相兼容。在 V_1 和 $V_2(V_1=V_2)$ 中的顶点 i 与集合 $\mathrm{pr}(X_3\cap X_1,d')$ 相对应，其中 $1\leqslant i\leqslant n$。边 $\{i,j\}\in E$，当且仅当 $\mathrm{pr}(X_3\cap X_1,d')=\mathrm{pr}(X_3\cap X_1,d')$ 和 $\mathrm{pr}(X_1\cap X_2,d')=\mathrm{pr}(X_2\cap X_3,d')$，或者 $\mathrm{pr}(X_3\cap X_1,d')=\mathrm{pr}(X_3\cap X_1,d')$ 和 $\mathrm{pr}(X_2\cap X_3,d')=\mathrm{pr}(X_2\cap X_3,d')$。

当且仅当在相对应的二分图 G 中有一个完全匹配的二分图时，投影 $\mathrm{pr}(X_1,F_1)$，$\mathrm{pr}(X_2,F_2)$ 和 $\mathrm{pr}(X_3,F_3)$ 才与数据集 d 相兼容。与图 G 中的二分匹配图相对应的所有数据集与投影 $\mathrm{pr}(X_1\cap X_2,d')$ 和 $\mathrm{pr}(X_2\cap X_3,d')$ 相兼容。与图 G 完全匹配的二分图正好与那些和投影 $\mathrm{pr}(X_1,F_1)=\mathrm{pr}(X_1,d')$，$\mathrm{pr}(X_2,F_2)=\mathrm{pr}(X_2,d')$ 和 $\mathrm{pr}(X_3,F_3)$ 相兼容的数据集对应。在二分图 $G=(V,E)$ 找到完全匹配的数据集的时间复杂度为 $O(\sqrt{|V|}\,|E|)$[10]。

定理 4 寻找与投影 $\mathrm{pr}(X_1,F_1)$，$\mathrm{pr}(X_2,F_2)$ 相兼容的数据集是一个 NP 难度问题。

证明：下面从 3 个部分问题进行归纳。假定一个集合 A 包括 $3l$ 个元素，一个定数 $B\in N$，和一个大小 $s(a)\in N$，$B/4<s(a)<B/2$，对于每一个 $a\in A$ 有 $\sum_{a\in A_i}s(a)=lB$，决定 A 是否被分成 l 个不相交的集合，如 $A_1,A_2,\cdots,A_n$ 等等，对于 $1\leqslant i\leqslant l$，$\sum_{a\in A_i}s(a)=B$[7]，因为这个 3 个部分问题是一个 NP 完全问题，则可以推测大小 $s(a)$，$a\in A$，被多项式 l 限制。

3 个部分问题 (A,B,s) 可以变为如下两个投影，W.l.o.g.，且 $A=[3l]$，$X_1=[\lceil\log l\rceil]$，$X_2=\lceil\log 3l\rceil+$

$[\lceil \log l \rceil]$,投影 $\mathrm{pr}(X_1, F)$ 由 $s(a)$ 的每一个集合 X 的副本组成,集合 X_1 对应于一个自然数 a 的二进制编码,每一个 $X(a) \subseteq X_1$,且 $a \in A = [3l]$。投影 $\mathrm{pr}(X_2, F)$ 由 $s(a)$ 的每一个集合 X 的副本组成,集合 X_2 对应于一个自然数 a 的二进制编码,每一个 $X(a) \subseteq X_1, b \in [l]$。

显然,对于 3 个部分问题(A, B, s)存在的充要条件是在两个投影 $\mathrm{pr}(X_1, F_1)$ 和 $\mathrm{pr}(X_2, F_2)$ 中,每一个都存在着一个具有 $3l$ 个不同集合的兼容数据集。

4 结　论

分析了频繁集挖掘的可计算复杂度问题,给出了在许多情况下问题的计算是困难的。对于已知的频繁集,用户不能计算出与已知频繁集兼容的不同数据集的数量,甚至不能决定是否存在着相容数据集。从隐私保持的观点看,频繁集的传递不会引起严重的隐私威胁,但另一方面反频繁集挖掘的计算是困难的。

参考文献:

[1] Mannila H. Local and global methods in data mining: Basic techniques and open problems[A]. In: Widmayer P, Triguero F, Morales R, et al. Automata, Languages and Programming, volume 2380 of Lecture Notes in Computer Science[C]. [s. l.]: Springer - Verlag, 2002. 57 - 68.

[2] Evfimievski A, Srikant R, Agrawal R, et al. Privacy preserving mining of association rules[A]. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. Edmonton, Alberta, Canada: ACM Press, 2002. 217 - 228.

[3] Oliverira S R M, Zaï O R. Privacy preserving frequent itemset mining[A]. In: Clifton C, Estivill - Castro V. IEEE ICDM Workshop on Privacy, Security, and Data Mining, volume 14 of Conferences in Research and Practice in Information Technology[C]. Maebashi City, Japan: [s. n.], 2002. 43 - 54.

[4] Gouka K, Zaki M J. Efficiently mining maximal frequent itemsets[A]. In: Cercone N, Lin T Y, Wu X. Proceedings of the 2001 IEEE International Conference on Data Mining[C]. Washington, DC: IEEE Computer society, 2001. 163 - 170.

[5] Gunopulos D, Khardon R, Mannila H, et al. Discovering all most specific sentences[J]. ACM Transactions on Database Systems, 2003, 28(2): 140 - 174.

[6] Calders T, Goethals B. Minimal k - free representations of frequent sets[A]. In: Lavrac N, Gamgerger D, Todorovski L, et al. Knowledge Discovery in Databases: PKDD 2003, volume 2838 of Lecture Notes in Artifical Intelligence[C]. [s. l.]: Springer - Verlag, 2003.

[7] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP - Completeness[Z]. New York - San Francisco: W. H. Freeman and Company, 1979.

[8] Knuth D E. Sorting and Searching, volume 3 of The Art of Computer Programming (2nd ed)[M]. Reading, Massachusetts: Addison - Wesley Publishing CO., 1998.

[9] Jukan S. Extremal Combinatorics: With Applications in Computer Science[A]. EATCS Texts in Theoretical Computer Science[C]. Berlin: Springer - Verlag, 2001.

[10] Gali Z. Efficient algorithms for finding maximum matchings in graphs[J]. ACM Computing Surveys, 1986, 18(1): 23 - 38.

(上接第 24 页)

件构架建模语言[6]。

软件构架不仅是系统开发项目的蓝图,而且也是将项目所有阶段结合在一起的概念纽带。对软件构架进行准确详尽的描述,是构建软件构架的最重要的步骤之一[7]。但在开发实践中,开发人员往往更关注于软件构架的设计,忽视了软件构架的描述。对再完美的软件构架设计如果没有进行详细明确有组织的描述,这个软件构架对整个开发过程也是没有太大帮助的。

Module, C&C, Allocation Viewpoint[5]是对描述软件构架的视图的高度抽象和概括。对于大型项目来说,Viewpoint 可以有助于开发人员从更高层次上把握整个系统构架,也更易对架构进行详尽准确的描述。

随着 UML2.0[8]的颁发和 IEEE 1471 的采用,基于 Viewpoint 的软件构架描述将更为广泛地应用于软件开发实践中。

参考文献:

[1] 万建成,卢　雷. 软件体系结构的原理、组成与应用[M]. 北京:科学出版社,2002.

[2] 冯　冲,江　贺. 软件体系结构理论与实践[M]. 北京:人民邮电出版社,2004.

[3] IEEE Computer Society. IEEE recommended practice for architectural description of software - intensive systems[R]. IEEE Std 1471, 2000.

[4] OMG (2003). *Unified Modeling Language specification 1.5* [EB/OL]. http://www.omg.org/uml/, 2001.

[5] Clements P, Bachmann F, Bass L. Documenting Software Architectures: Views and Beyond[M]. [s. l.]: Addison - Wesley publication house, 2002.

[6] Medvidovic N, Rosenblum D S, Redmiles D F. Modeling Software Architectures in the United Modeling Language[J]. ACM Transactions on Software Engineering and Methodology, 2002, 11(1): 52 - 57.

[7] Clements P, Kazman R, Klein M. Software Architecture in Practice(2 edition)[M]. [s. l.]: Addison - Wesley publication house, 2003.

[8] OMG. United Modeling Language[EB/OL]. http://www.omg.org/uml/, 2003.