软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(12):2934−2950 [doi: 10.3724/SP.J.1001.2011.03978]

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

基于贝叶斯网络的频繁模式兴趣度计算及剪枝*

胡春玲[1,2], 吴信东[1+], 胡学钢[1], 姚宏亮[1]

[1](合肥工业大学 计算机与信息学院,安徽 合肥 230009)

[2](合肥学院 网络与智能信息处理重点实验室,安徽 合肥 230601)

Computing and Pruning Method for Frequent Pattern Interestingness Based on Bayesian Networks

HU Chun-Ling[1,2], WU Xin-Dong[1+], HU Xue-Gang[1], YAO Hong-Liang[1]

[1](School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China)

[2](Key Laboratory of Network and Intelligent Information Processing, Hefei University, Hefei 230601, China)

+ Corresponding author: E-mail: xwu@hfut.edu.cn

Hu CL, Wu XD, Hu XG, Yao HL. Computing and pruning method for frequent pattern interestingness based on Bayesian networks. *Journal of Software*, 2011,22(12):2934−2950. http://www.jos.org.cn/1000-9825/3978.htm

Abstract: Based on background knowledge represented as a Bayesian network, this paper presents a BN-EJTR method that computes the interestingness of frequent items and frequent attributes, and prunes. BN-EJTR seeks to find inconsistent knowledge relative to background knowledge and to resolve the problems of un-interestingness and redundancy faced by frequent pattern mining. To deal with the demand of batch reasoning in Bayesian networks during computing interestingness, BN-EJTR provides a reasoning algorithm based on extended junction tree elimination for computing the support of a large number of items in a Bayesian network. In addition, BN-EJTR is equipped with a pruning mechanism based on a threshold for topological interestingness. Experimental results demonstrate that BN-EJTR has a good time performance compared with the same classified methods, and BN-EJTR also has effective pruning results. The analysis indicates that both the pruned frequent attributes and the pruned frequent items are un-interesting in respect to background knowledge.

Key words: frequent pattern; Bayesian network; junction tree; interestingness; pruning

摘 要: 采用贝叶斯网络表示领域知识,提出一种基于领域知识的频繁项集和频繁属性集的兴趣度计算和剪枝方法 BN-EJTR,其目的在于发现与当前领域知识不一致的知识,以解决频繁模式挖掘所面临的有趣性和冗余问题.针对兴趣度计算过程中批量推理的需求,BN-EJTR 提供了一种基于扩展邻接树消元的贝叶斯网络推理算法,用于计算大量项集在贝叶斯网络中的支持度;同时,BN-EJTR 提供了一种基于兴趣度阈值和拓扑有趣性的剪枝算法.实验结果表明,与同类方法相比,方法 BN-EJTR 具有良好的时间性能,而且剪枝效果明显;分析发现,经过剪枝后的频繁属性集和频繁项集相对于领域知识符合有趣性要求.

关键词: 频繁模式;贝叶斯网络;邻接树;兴趣度;剪枝

中图法分类号: TP181 **文献标识码**: A

* 基金项目: 国家自然科学基金(60828005, 60975034, 61070131)
收稿时间: 2010-01-11; 修改时间: 2010-07-09; 定稿时间: 2010-12-15

频繁模式是指数据集中满足一定支持度阈值的项集、子序列和子结构[1].频繁模式发现是数据挖掘领域的重要方法和主要研究方向之一,频繁模式在关联规则发现、分类、聚类等重要的数据挖掘方法中有着广泛的应用[2−4],频繁模式发现的研究和突破将对整个数据挖掘方法产生重要影响.当前,对频繁模式发现方法本身的研究比较充分,而对所发现频繁模式的评价和排序研究存在不足,从而导致挖掘出来大量且冗余的频繁模式无法得到有效利用[5].因此,研究频繁模式的兴趣度度量并发现有效的剪枝方法,对初次挖掘产生的频繁模式集进行二次挖掘是非常必要的.

频繁模式冗余的原因主要有两方面:一是频繁模式不满足层次有趣性,子模式有趣可能直接导致包含该子模式的超模式有趣;二是大量的频繁模式符合领域知识,不具有新颖有趣性.针对频繁模式冗余问题的解决方法主要分为两大类:一是根据超模式与其子模式的置信度差异对频繁模式集进行剪枝,去除冗余的频繁模式[6−8];二是定义频繁模式的兴趣度度量,按照频繁模式的兴趣度对频繁模式集进行排序[9,10].目前,已有大量针对频繁模式的兴趣度度量方法,Carvalho[11]和 Ohsaki[12]等人比较了不同的兴趣度度量方法,其研究表明:按现有兴趣度度量产生的兴趣度高的频繁模式往往是平凡的,为用户所熟知的;用户通常只对领域知识不一致的知识真正感兴趣.因此,基于领域知识度量频繁模式的兴趣度将是解决频繁模式冗余问题的有益尝试.

Padmanabhan 等人将局部知识作为领域知识引入到频繁模式兴趣度的度量[13,14],但其领域知识的表示不具有全局性,在此基础上定义的兴趣度具有不一致性,从而导致挖掘出来的频繁项集和关联规则同样不具有全局性.而贝叶斯网络是一种基于概率理论和图论的具有一致性和全局性的知识表示方式,其网络结构中蕴含容易理解的因果依赖关系、独立关系和条件独立关系;同时,贝叶斯网络具有灵活的推理机制和双向推理能力,能对任意变量子集和任意查询条件下的边缘概率和条件概率进行查询,贝叶斯网络中蕴含的条件概率、联合概率和频繁模式的置信度、支持度之间有着一定的联系.因此,基于贝叶斯网络度量频繁模式的兴趣度将成为解决频繁模式兴趣度不一致性问题的一种途径.

目前,基于贝叶斯网络的频繁模式挖掘方法正在逐渐引起人们的重视.丁贵涛等人提出了一种基于贝叶斯网络的关联规则提取方法[15],该方法基于贝叶斯网络的局部结构,只挖掘具有依赖关系和因果语义的关联规则.相比于只具有统计意义上并发的传统关联规则集,此处所得到的规则集具有因果语义,且在减少冗余的同时提高了精度.但得到的规则通常是用户所熟知的,不符合有趣性要求;Jaroszewicz 等人将贝叶斯网络引入频繁模式的兴趣度度量,提出了一种基于贝叶斯网络的频繁模式挖掘方法 ExactInter[16].该方法采用贝叶斯网络表示领域知识,挖掘与领域知识不一致的频繁模式,得到的频繁模式符合有趣性要求.但方法 ExactInter 采用桶消元推理算法计算大量属性集和项集在贝叶斯网络中的支持度,导致其时间性能较差.

为了解决 ExactInter 方法的时间性能问题,Malhas 等人将 Netica 的 NeticaJ 接口中的函数 getFindingsProbability()用于 ExactInter 方法进行贝叶斯网络推理,提出了一种 PJC 方法(patternminer Java class)[17].函数 getFindingsProbability()实现的是一种基于邻接树的贝叶斯网络推理算法,一次推理过程的时间性能较好,但对批量推理问题不具有针对性.调用函数 getFindingsProbability()一次可以计算一个项集在贝叶斯网络中的支持度,一个属性集所对应的不同项集在贝叶斯网络中的支持度要通过多次调用来完成.因此,方法 PJC 的时间性能对支持度阈值高度敏感,不适合处理支持度低的频繁模式发现问题,而本类方法的基本目的是发现支持度较低而兴趣度较高的新颖频繁模式.

Jaroszewicz 等人采用顺序抽样的思想,提出一种基于贝叶斯网络的频繁模式兴趣度近似计算方法 ApproxInter[18,19],该方法在每一次迭代过程中对贝叶斯网络和原数据集进行抽样,在抽样所得的分别代表贝叶斯网络和原数据集的二次数据集中进行属性集和项集的支持度和兴趣度的近似计算,通过反复迭代,最终能在一定的置信区间和误差范围内按兴趣度发现前 K 个频繁模式.ApproxInter 方法的时间性能取决于其迭代过程的收敛速度,而迭代过程的收敛速度取决于数据规模和计算精度要求.方法 ApproxInter 针对基于中小规模的贝叶斯网络的频繁模式发现问题不具有明显的时间性能上的优势,且存在一定的计算误差.因此,该方法一般用于精确推理无法执行的大规模数据集或数据流.

针对基于贝叶斯网络的频繁模式发现问题,本文提出一种基于贝叶斯网络的频繁属性集和频繁项集的兴

趣度计算和剪枝方法 BN-EJTR(BN-extended-junction-tree-reasoning),其目的在于能够有效发现与领域知识不一致的有趣知识.方法 BN-EJTR 有两个特点:① 该方法基于 Apriori 算法及其扩展算法从数据集和贝叶斯网络中产生频繁项集,针对大量项集在贝叶斯网络中支持度的计算问题,提出了一种基于扩展邻接树的推理算法 EJTR:该算法能够基于扩展的邻接树选择完备或近似完备的消元次序,并通过对邻接树结构的扩展,存储消元推理过程中传播的消息因子,从而能够避免批量推理过程中相同消息因子的重复计算,提高批量推理的效率;② 通过对频繁模式的兴趣度和贝叶斯网络结构的分析,提出了一种基于兴趣度阈值和拓扑有趣性进行剪枝的剪枝算法 Prune-Topo.实验结果表明,方法 BN-EJTR 具有良好的时间性能,剪枝效果明显,剪枝后保留下来的频繁属性集和频繁项集不存在层次和拓扑结构上的冗余,相对于其领域知识具有真正的有趣性.

1 相关定义

本文采用大写字母 $X,Y,Z,\ldots$表示数据集 D 中属性和贝叶斯网络 BN 中的相应节点;采用 $I,J,\ldots$表示属性子集,H 表示属性全集;$|I|$表示属性集的长度,$Dom(I)$表示属性集 I 的值域,(I,i)表示项集,其中,$i\in Dom(I)$;$Supp_D(I,i)$和 $Supp_{BN}(I,i)$分别表示项集(I,i)在数据集 D 和贝叶斯网络 BN 中的支持度,$P_{BN}(I)$为属性集在贝叶斯网络 BN 中的边缘概率分布,$P_D(I)$为属性集 I 的不同取值在数据集 D 中的发生频率,且有

$$Supp_D(I,i)=P_D(I=i),Supp_{BN}(I,i)=P_{BN}(I=i).$$

本文采用 $BN=\{V,E\}$表示定义在属性全集 $H=\{X_1,X_2,\ldots,X_n\}$上的贝叶斯网络,其中,$V=H$ 表示节点集,E 表示边集,$Par(X_i)=\{X_j:X_j\to X_i\in E\}$表示节点 X_i 的父节点集,$Anc(X_i)$表示节点 X_i 中的祖先节点集.基于网络结构中所蕴含的条件独立性,贝叶斯网络的联合概率分布可分解成每个节点相对于其父节点的条件概率的乘积,即

$$P(H)=P(X_1,X_2,...,X_n)=\prod_{i=1}^{n}P(X_i\mid Par(X_i)).$$

属性子集 I 在贝叶斯网络中的边缘概率分布可按公式(1)通过对 $H–I$ 中的变量汇总消元进行计算,即

$$P_{BN}(I)=\sum_{H-I}P_{BN}(H),\text{其中},I\subseteq H \tag{1}$$

在网络节点较多的情况下,直接通过对 $P_{BN}(H)$的汇总消元来计算 $P_{BN}(I)$并不可行,需要根据实际问题,设计有效的贝叶斯网络推理算法来完成边缘概率分布的计算.

定义 1. 数据集 D 对应的贝叶斯网络为 BN,如果项集(I,i)在数据集 D 和贝叶斯网络 BN 中的支持度 $Supp_D(I,i)$和 $Supp_{BN}(I,i)$中至少有一个大于等于预先定义的支持度阈值 Min$Supp$,则称项集(I,i)为频繁项集,属性集 I 为频繁属性集.

以下定义 2 和定义 3 均引自文献[16],本文对定义 2 进行了整理,并通过实例进一步分析了其意义.

定义 2. 数据集 D 对应的贝叶斯网络为 BN,频繁项集(I,i)的兴趣度为该项集在数据集 D 和贝叶斯网络 BN 中的支持度之差的绝对值,记为 $Int_{BN}(I,i)$.即:$Int_{BN}(I,i)=|Supp_D(I,i)–Supp_{BN}(I,i)|$.如果 $Int_{BN}(I,i)$大于或等于预先定义的兴趣度阈值ε,则称频繁项集(I,i)有趣;否则,频繁项集(I,i)无趣.

由定义可知,频繁项集的兴趣度即指其在数据集 D 和贝叶斯网络 BN 中的支持度的差异度,而兴趣度大小则表示该频繁项集与当前领域知识不一致的程度.为了更好地理解兴趣度的定义及意义,现举例:假设支持度阈值 Min$Supp$ 为 0.01,兴趣度阈值ε为 0.05,S 表示病人是否具有某种症状,T 表示是否医生采取某种治疗方案,C 表示是否被治愈.

例 1:假设频繁项集$([S,T,C],[1,1,1])$在数据集 D 中的支持度为 0.07,在贝叶斯网络 BN 中支持度为 0.01,则其兴趣度为 $Int_{BN}([S,T,C],[1,1,1])=|0.07–0.01|=0.06$.由定义 2 知,该频繁项集是有趣的.

因为频繁项集$([S,T,C],[1,1,1])$在贝叶斯网络中的支持度仅为 0.01,即领域知识中不提供这种治疗方案,绝大多数医生在处理 S 症状选择 T 以外的其他治疗方案,所以频繁项集$([S,T,C],[1,1,1])$在数据集 D 中的支持度很低,进而导致该项集通过常规的挖掘方法(通过支持度阈值都会选择在 0.1 以上)不容易被发现.而事实上,该项集可能是一种极为有效的治疗方案,它在当前数据集中的支持度低是因为医生目前还没有发现这种治疗方案,因而该项集对应的治疗方案是有趣的.

例 2:假设频繁项集([*S*,*T*,*C*],[1,1,1])在数据集 *D* 中的支持度为 0.957,在贝叶斯网络 *BN* 中支持度为 0.958,则其兴趣度为 Int_{BN}([*S*,*T*,*C*],[1,1,1])=|0.957−0.958|=0.01.由定义 2 知,该频繁项集是无趣的.因为在现实生活中,绝大多数医生会根据领域知识对具有 *S* 症状的病人采取 *T* 这种常规且有效的治疗方案,以达到治疗的目的.也就是说,频繁项集([*S*,*T*,*C*],[1,1,1])对应的是一种常规治疗方案,尽管其在数据集 *D* 中的支持度很高,但却是无趣的.

定义 3. 数据集 *D* 对应的贝叶斯网络为 *BN*,频繁属性集 *I* 的兴趣度为该属性集对应的所有频繁项集(*I*,*i*)的兴趣度的最大值,记为 $Int_{BN}(I)$.即,$Int_{BN}(I)$=Max($Int_{BN}(I,i)$).如果 $Int_{BN}(I)$大于或等于预先定义的兴趣度阈值ε,则频繁属性集 *I* 有趣;否则,频繁属性集 *I* 无趣.

定义 4. 假设 *FreqA* 为不同频繁属性集构成的频繁属性集合,*FreqA* 中不被 *FreqA* 中其他频繁属性集包含的频繁属性集为长频繁属性集,反之则为短频繁属性集.

2 基于贝叶斯网络的频繁项集和属性集的兴趣度计算和剪枝方法 BN-EJTR

方法 BN-EJTR 共分 4 个步骤:① 基于 Apriori 算法从数据集 *D* 中产生频繁项集集合 *FreqI*;② 基于扩展的邻接树推理算法 EJTR 计算 *FreqI* 中所有频繁项集在贝叶斯网络 *BN* 的支持度 $Supp_{BN}(I,i)=P_{BN}(I=i)$;③ 基于扩展的 Apriori 算法从贝叶斯网络 *BN* 中产生频繁项集结合 *FreqI*′,并对其中未在 *FreqI* 中出现的频繁项集从数据集 *D* 中计算其支持度 $Supp_D(I,i)=P_D(I=i)$;④ 基于 Prune-Topo 算法计算所有频繁项集和属性集的兴趣度,并按兴趣度对频繁项集和属性集进行排序,然后根据兴趣度阈值和拓扑有趣性对频繁属性集和频繁项集进行剪枝.每个步骤将在下面的小节里进行详细说明.

2.1 从数据集中产生频繁项集

方法 BN-EJTR 采用经典的 Apriori 算法从数据集 *D* 中产生频繁项集[20],一是由于本方法不仅针对事务数据集,还因为 Apriori 算法具有普遍的适用性;二是频繁项集挖掘算法的时间性能对方法 BN-EJTR 的时间性能影响不大.

算法 1. Apriori 算法.

输入:数据集 *D*,支持度阈值 *MinSupp*,项集最大属性长度 Max*K*;

输出:频繁项集集合 *FreqI*.

1) 项集长度 *K* 初始化为 1:*K*←1.
2) 产生候选 *K* 项集:$Cand_K$←{(*I*,*i*):|*I*|=*K*,*i*∈*Dom*(*I*)}.
3) 从数据集 *D* 计算候选 *K* 项集的支持度:{$Supp_D(I,i)$:(*I*,*i*)∈$Cand_K$}.
4) 根据支持度阈值产生频繁 *K* 项集:$\begin{cases} Freq_k \leftarrow \{(I,i):(I,i)\in Cand_K \wedge Supp_D(I,i)\geqslant \mathrm{Min}Supp\} \\ FreqI \leftarrow Freq_k \end{cases}$.
5) *K*←*K*+1,若 *K*>*MaxK*,则停止迭代并输出 *FreqI*;否则,转下一步.
6) 根据频繁 *K*–1 项集产生候选 *K* 项集:$Cand_K$←{(*I*,*i*):|*I*|=*K*,*i*∈*Dom*(*I*) From $Freq_{K-1}$}.
7) 转第 3)步.

2.2 计算频繁项集在贝叶斯网络中的支持度

方法 BN-EJTR 需要计算从数据集 *D* 中产生的频繁项集集合 *FreqI* 中的所有频繁项集在相应的贝叶斯网络 *BN* 中的支持度,计算频繁项集在贝叶斯网络中的支持度实际上是基于贝叶斯网络的消元推理过程计算频繁项集在贝叶斯网络中的边缘概率分布.贝叶斯网络推理的计算复杂度依赖于网络结构和消元次序[21],网络结构是无法选择的,可以通过选择消元次序来改善推理效率.本文针对批量推理的需求,提出了一种基于扩展的邻接树进行消元并存储消元过程中传播的消息因子的推理算法 EJTR,用于计算大量频繁变量集在贝叶网络中的边缘概率分布.显然,频繁变量集的边缘概率分布包含了该频繁变量集对应的全部项集的边缘概率分布.

2.2.1 扩展邻接树的建立

算法 EJTR 首先根据贝叶斯网络构建一棵邻接树,如果贝叶斯网络对应的道义图即为有弦图,则该道义图

存在完备的消元次序;如果贝叶斯网络对应的道义图不是有弦图,本文按节点及其邻居节点状态数乘积递增的顺序对道义图进行有弦化,构成有弦图.有弦图一定存在完备的消元次序[22,23].贝叶斯网络经过道义化和有弦化,得到一个存在完备消元次序的无向图.此时,可按定理 1 的证明过程构建其对应邻接树.

定理 1. 设 G 是一个无向图,且存在完备消元次序,则 G 可以构成一棵邻接树.

证明:证明的过程就是构造邻接树的过程.对于存在完备消元次序无向图 G,其中至少存在一个节点 X,使得 X 及其所有邻居节点在图 G 中构成一个子团 C_x.选定一个这样的节点 X,从 C_x 中删除所有邻居节点均在其中的节点,并将 C_x 按消元数目 i 编号,记为 C_i.记消元后 C_i 中剩下的节点组成的集合为 L_i,称为子团 C_i 的割集.图 G 删除节点 X 及所有邻居节点均在其中的节点后,所产生的新图 G'仍然存在完备消元次序,按同样的方法进行消元,产生一系列的子团和及相应的割集,并将这些子团(或割集)按消元的累计数编号.对于每个割集 L_i,按下标递增的顺序找到一个下标大于 i 而且与 L_i 当前所连接子团的交集恰好为 L_i 的子团 $C_j(j>i)$,并形成一条从 C_i 经过 L_i 指向 C_j 的边,由此形成的结构就是对应于图 G 的一种树结构.以下证明 C_j 是存在的,且该树结构是满足邻接树定义的,因而是一种邻接树结构.

现在证明树结构产生过程中的每一个割集 L_i,包含 L_i 的另一子团 $C_j(j>i)$是存在的.因为割集 L_i 是一个完全子图,在消去该割集中的变量 X 时,对应的子团一定包含 L_i,故子团 $C_j(j>i)$是存在的.然后证明针对树结构中的任一连接子团 C_i 和 C_j 的路径,如果节点 X 同时属于 C_i 和 C_j,则 X 属于该路径上的任一子团和割集.因为 X 属于子团 C_i 和 C_j,则 X 一定不是在子团 C_i 中消元的,则 X 属于割集 L_i.根据该树结构的构造过程,则 X 一定属于连接割集 L_i 的另一子团 C_k,如果 C_k 即为 C_j,则得证;如果 C_k 不为 C_j,则同理,X 一定属于树结构中 C_k 的父节点和相应割集 L_k,一直进行下去,直到父节点为 C_j 为止. □

合并邻接树中的相邻子团,所得到的树结构仍然满足邻接树的定义[24];若合并后的子团宽度不超过当前邻接树的宽度,则邻接树推理算法的时间复杂度保持不变[21].为了提高批量频繁属性集在贝叶斯网络中推理的效率,现推广邻接树的概念,得到一种扩展邻接树.设 $P=\{P_1,P_2,\ldots,P_n\}$对应贝叶斯网络 BN 中每个节点的条件概率分布表,由 BN 生成的无向有弦图 G 存在完备消元次序,按定理 1 的证明过程构成贝叶斯网络 BN 对应的邻接树 T,$C=\{C_1,C_2,\ldots,C_t\}$是 T 中所有子团组成的集合,$L=\{L_1,L_2,\ldots,L_{t-1}\}$是相应割集组成的集合.对 T 作如下 3 点扩展:

① 若邻接树 T 中相邻子团 C_i 和 $C_j(j>i)$的宽度之和不超过邻接树邻接宽度 w,则对子团 C_i 和 C_j 进行合并,合并后的子团记为 C_j,C_j 和 C_i 的子节点之间的割集为合并前 C_i 与其子节点之间的割集.这一合并不会增加邻接树推理算法的时间复杂度,但却可以使邻接树中的子团包含更多的频繁属性集.因此,在一次消息传播过程结束、邻接树处于一致状态下,可以直接在邻接树的不同子团中完成更多的频繁属性集在贝叶斯网络中支持度的计算,从而提高批量推理的效率.

② 将 P 中任一条件概率分布 P_j 分配邻接树 T 中一个能包含其所有自变量的子团 C_i 中,并为 T 中的每个子团 C_i 定义一个结构体,该结构体由两个数据项组成:一个数据项用于存放该子团所包含的节点,记为 $C_i.node$;另一个数据项用于存放该子团的势函数,即为分配到该子团的所有条件概率分布函数的乘积,记为 $C_i.\phi$.这样,可以避免子团的势函数在批量推理过程的重复计算.

③ 为 T 中连接子团 C_i 和 C_j 的每条边上割集 L_i 定义一个结构体,该结构体由 3 个数据项组成:第 1 个数据项用于存放该边对应的割集所包含的节点,记为 $L_i.node$;第 2 个数据项用于存储邻接树中自上而下传播的消息 $L_i.\pi$,第 3 个数据项为一个数据表 $L_i.\lambda$,用于存储邻接树中自下而上传播的消息.针对大量频繁属性集边缘概率的计算问题,算法 EJTR 在基于扩展的邻接树通过自下而上的消息传播消元计算频繁属性集 I 的边缘概率过程中,消元后产生的消息不仅传播给其在邻接树中的父节点,而且以消元后对应的变量子集为键值保存在数据表 $L_i.\lambda$ 中.在计算其他频繁属性集的边缘概率过程中,如果涉及到同样的计算因子,将不再需要重复计算.由此形成的树结构称为 T 的扩展邻接树,记为 T^*,T^*中编号最大的子团 C_t 为其根节点.

2.2.2 扩展邻接树的初始化

算法 EJTR 通过自下而上和自上而下的两个消息传播阶段完成对扩展邻接树的初始化:

① 在自下而上的消息传播阶段,从扩展邻接树的叶节点开始,每个节点 C_i 在收到来自其所有子节点 C_k 传

播过来的消息 $L_k.\lambda(L_k.node)$后,按公式 $L_i.\lambda(L_i.node)=\sum_{C_i.node-L_i.node} C_i.\phi\prod_k L_k.\lambda(L_k.node)$ 计算向其父节点 C_j 传播的消息,并以 $L_i.node$ 为键值存储在数据表 $L_i.\lambda$中,一直进行到扩展邻接树的根节点;

② 然后进入自上而下的消息传播阶段,每个节点 C_i 收到其父节点 C_j 传播过来的消息和除第 k 个子节点之外的其他子节点传播过来的消息后,按公式 $L_k.\pi=\sum_{C_i.node-L_k.node} C_i.\phi\times L_j.\pi\prod_{p\neq k} L_p.\lambda(L_p.node)$ 计算向其第 k 个子节点 C_k 传播的消息,一直进行到扩展邻接树的叶节点.

初始化结束后,扩展邻接树进入一致状态,则子团 C_i 对应节点集及其子集 I 的边缘概率分布可按公式(2)进行计算:

$$P_{BN}(C_i.node)=C_i.\phi\times L_j.\pi\prod_k L_k.\lambda(L_k.node),P_{BN}(I)=\sum_{C_i.node-I} P_{BN}(C_i.node) \tag{2}$$

其中,$I\subset C_i.node$.一致状态下,割集 L_i 对应节点集及其任一子集的边缘概率分布可按公式(3)进行计算:

$$P_{BN}(L_i.node)=L_i.\pi\times L_i.\lambda(L_i.node),P_{BN}(I)=\sum_{L_i.node-I} P_{BN}(L_i.node) \tag{3}$$

其中,$I\subset L_i.node$.

2.2.3 频繁属性集边缘概率的计算

算法 EJTR 将频繁属性集分成两大类:一类是属于扩展邻接树中某一子团或割集的频繁属性集,此类频繁属性集在邻接树进入一致状态后,可按公式(2)、公式(3)进行计算;另一类是不属于扩展邻接树中某一子团或割集的频繁属性集,算法 EJTR 将此类频繁属性集分为长频繁属性集和短频繁属性集,通过直接对长频繁属性集的边缘概率按公式(4)进行消元的方法计算短频繁属性集的边缘概率:

$$P_{BN}(J)=\sum_{I-J} P_{BN}(I), \tag{4}$$

其中,$J\subset I$.

算法 EJTR 采用基于扩展的邻接树 T^*进行消元的方法计算长频繁项集 I 的边缘概率,消元过程与扩展邻接树 T^*在初始化过程中自下而上的消息传播过程类似,唯一不同的是,每次传播的消息不是按割集进行消元,而是根据割集和待查询变量集进行消元,消去当前子团中既不属于相应割集也不属于待查询变量集的变量,子团 C_i 向其父节点传播的消息为 $L_i.\lambda(S)=\sum_{C_i.node-L_i.node-I} C_i.\phi\prod_k L_k.\lambda(S_k)$,其中,$S$ 为本次消元后对应的变量子集,S_k 为子团 C_i 的子节点 k 传播上来的消息对应的变量子集.最后,通过在根节点 C_t 中按公式(5)进行汇总消元计算长频繁属性集 I 的边缘概率分布:

$$P_{BN}(I)=\sum_{C_t.node-I} C_t.\phi\prod_k L_k.\lambda(S_k) \tag{5}$$

2.2.4 算法 EJTR 一个实例

以下举例演示 EJTR 算法计算频繁属性集边缘概率分布的过程.假设需要计算频繁属性集 $FreqA$={(X_1,X_2,X_3),(X_2,X_3,X_5),(X_2,X_3,X_6),(X_3,X_5,X_6),(X_1,X_2),(X_1,X_3),(X_1,X_5),(X_2,X_3),(X_2,X_4),(X_2,X_5),(X_2,X_6),(X_3,X_4),(X_3,X_5),(X_3,X_6),(X_5,X_6),(X_1),(X_2),(X_3),(X_4),($X5$),(X_6)}在如图 1 所示的贝叶斯网络中的边缘概率分布,该网络对应的扩展邻接树 T^*如图 2 所示,该扩展邻接树的根节点为子团 C_6.基于扩展邻接树 T^*计算频繁属性集合 $FreqA$ 的边缘概率分布时,将频繁属性集分成两大类:第 1 类 $FreqA_1$={(X_1,X_2,X_3),(X_2,X_3,X_5),(X_1,X_2),(X_1,X_3),(X_2,X_3),(X_2,X_4),(X_2,X_5),(X_3,X_5),(X_3,X_6),(X_1),(X_2),(X_3),(X_4),(X_5),(X_6)}为属于邻接树中的某一子团或割集的频繁属性集合,按长度递减的顺序排列;第 2 类 $FreqA_2$={(X_2,X_3,X_6),(X_3,X_5,X_6),(X_1,X_5),(X_2,X_6),(X_3,X_4),(X_5,X_6)}为不属于邻接树中的某一子团或割集频繁属性集合,同样按长度递减的顺序排列.$FreqA_1$ 和 $FreqA_2$ 按长度递减排列的目的是为了确保当前排在第一的频繁属性集为长频繁属性集.

算法 EJTR 首先通过自下而上和自上而下的消息传播对扩展的邻接树 T^*进行初始化,使扩展邻接树处于一致状态,一致状态下的扩展邻接树如图 3 所示.根据这一处于一致状态的扩展邻接树,即可按长度递减的顺序计算第 1 类频繁属性集合 $FreqA_1$ 的每一个频繁属性集及其子集的的边缘概率分布.表 1 记录了这一计算过程,同

时记录了两次消息传播过程中传播并在相应割集相应域存储的消息因子.

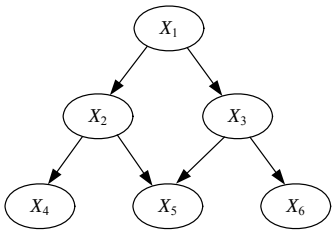

Fig.1 A Bayesian network

图 1 一个贝叶斯网络

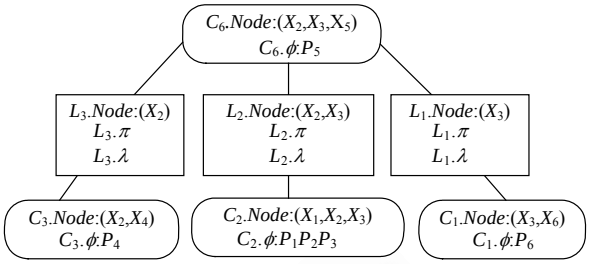

Fig.2 An extended junction tree of Fig.1

图 2 图 1 的扩展邻接树

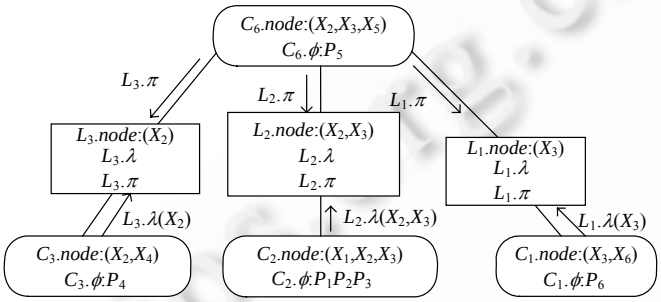

Fig.3 An initialized extended junction tree

图 3 初始化后的扩展邻接树

Table 1 Marginal probability calculations for frequent attribute sets in $FreqA_1$

表 1 $FreqA_1$ 中频繁属性集边缘概率的计算

Inward message propagation phase		
$C_1 \to C_6$	$C_2 \to C_6$	$C_3 \to C_6$
$L_1.\lambda(X_3)=\sum_{X_6} C_1.\phi$	$L_2.\lambda(X_2,X_3)=\sum_{X_1} C_2.\phi$	$L_3.\lambda(X_2)=\sum_{X_4} C_3.\phi$
Outward message propagation phase		
$C_6 \to C_1$	$C_6 \to C_2$	$C_6 \to C_3$
$L_1.\pi=\sum_{X_2,X_5}(C_6.\phi \times L_2.\lambda(X_2,X_3)\times L_3.\lambda(X_2))$	$L_2.\pi=\sum_{X_5}(C_6.\phi \times L_1.\lambda(X_3)\times L_3.\lambda(X_2))$	$L_3.\pi=\sum_{X_3,X_5}(C_6.\phi \times L_1.\lambda(X_3)\times L_2.\lambda(X_2,X_3))$
Extended junction tree entered into accordant state		
Frequent attribute sets in $FreqA_1$	Marginal probability calculations for frequent attribute sets	
$(X_1,X_2,X_3),(X_1,X_2),(X_1,X_3),(X_2,X_3),(X_1),(X_2),(X_3)$	$P_{BN}(X_1,X_2,X_3)=P(C_2.node)=C_2.\phi\times L_2.\pi$ $P_{BN}(X_1,X_2)=\sum_{X_3}P_{BN}(X_1,X_2,X_3), P_{BN}(X_1,X_3)=\sum_{X_2}P_{BN}(X_1,X_2,X_3)$ $P_{BN}(X_2,X_3)=\sum_{X_1}P_{BN}(X_1,X_2,X_3), P_{BN}(X_1)=\sum_{X_2}P_{BN}(X_1,X_2)$ $P_{BN}(X_2)=\sum_{X_1}P_{BN}(X_1,X_2), P_{BN}(X_3)=\sum_{X_1}P_{BN}(X_1,X_3)$	
$(X_2,X_3,X_5),(X_2,X_5),(X_3,X_5),(X_5)$	$P_{BN}(X_2,X_3,X_5)=P(C_6.node)=C_6.\phi\times L_1.\lambda(X_3)\times L_2.\lambda(X_2,X_3)\times L_3.\lambda(X_2)$ $P_{BN}(X_2,X_5)=\sum_{X_3}P_{BN}(X_2,X_3,X_5), P_{BN}(X_3,X_5)=\sum_{X_2}P_{BN}(X_2,X_3,X_5), P_{BN}(X_5)=\sum_{X_3}P_{BN}(X_3,X_5)$	
$(X_2,X_4),(X_4)$	$P_{BN}(X_2,X_4)=P(C_3.node)=C_3.\phi\times L_3.\pi, P_{BN}(X_4)=\sum_{X_2}P_{BN}(X_2,X_4)$	
$(X_3,X_6),(X_6)$	$P_{BN}(X_3,X_6)=P(C_1.node)=C_1.\phi\times L_1.\pi, P_{BN}(X_6)=\sum_{X_3}P_{BN}(X_3,X_6)$	

采用基于扩展的邻接树自下而上的消元推理算法计算频繁属性集合 $FreqA_2$ 中的每一个长频繁属性集的边缘概率分布,并通过公式(4)直接消元的方法计算该长频繁属性集所包含的当前边缘概率分布未计算的短频繁属性集的边缘概率分布.表 2 记录了 $FreqA_2$ 中的频繁属性集的边缘概率的计算过程,该表的第 1 列表示长频繁频繁属性集及其所包含边缘概率未计算的短频繁属性集;最后一列表示在扩展邻接树的根节点 C_6 中根据其势函数和其子节点传播上来的消息因子计算 $FreqA_2$ 中长频繁属性集边缘概率分布,随后通过直接消元的方法计算该长频繁属性集所包含的短频繁属性集的边缘概率分布;中间各列表示不同子团传播给其父节点的消息,并存储在相应割集 $L.\lambda$数据表中.算法 EJTR 能够复用邻接树在建立和消息传播过程中存储的消息,表 2 中传播的所有消息均在建立扩展邻接树和对扩展邻接树初始化的过程中进行计算并存储的,需要进行的计算仅在该表的最后一列进行,即根据公式计算频繁项集合 $FreqA_2$ 中各频繁属性集的边缘概率分布.

Table 2 Marginal probability calculations for frequent attribute sets in $FreqA_2$

表 2 $FreqA_2$ 中频繁属性集边缘概率的计算

Frequent attribute sets	Message propagation, marginal probability calculations			
	$C_1\rightarrow C_6$	$C_2\rightarrow C_6$	$C_3\rightarrow C_6$	C_6
(X_2,X_3,X_6) (X_2,X_6)	$L_1.\lambda(X_3,X_6)=C_1.\phi$	$L_2.\lambda(X_2,X_3)=\sum_{X_1}C_2.\phi$	$L_3.\lambda(X_2)=\sum_{X_4}C_3.\phi$	$P_{BN}(X_2,X_3,X_6)=L_1.\lambda(X_3,X_6)\times L_2.\lambda(X_2,X_3)\times L_3.\lambda(X_2)\times\sum_{X_5}C_6.\phi$
	$P_{BN}(X_2,X_6)=\sum_{X_3}P_{BN}(X_2,X_3,X_6)$			
(X_3,X_5,X_6) (X_5,X_6)	$L_1.\lambda(X_3,X_6)=C_1.\phi$	$L_2.\lambda(X_2,X_3)=\sum_{X_1}C_2.\phi$	$L_3.\lambda(X_2)=\sum_{X_4}C_3.\phi$	$P_{BN}(X_3,X_5,X_6)=L_1.\lambda(X_3,X_6)\times\sum_{X_2}(L_2.\lambda(X_2,X_3)\times L_3.\lambda(X_2)\times C_6.\phi)$
	$P_{BN}(X_5,X_6)=\sum_{X_3}P_{BN}(X_3,X_5,X_6)$			
(X_1,X_5)	$L_1.\lambda(X_3)=\sum_{X_6}C_1.\phi$	$L_2.\lambda(X_1,X_2,X_3)=C_2.\phi$	$L_3.\lambda(X_2)=\sum_{X_4}C_3.\phi$	$P_{BN}(X_1,X_5)=\sum_{X_3}L_1.\lambda(X_3)\times\sum_{X_2}(L_2.\lambda(X_1,X_2,X_3)\times L_3.\lambda(X_2,X_4)\times C_6.\phi)$
(X_3,X_4)	$L_1.\lambda(X_3)=\sum_{X_6}C_1.\phi$	$L_2.\lambda(X_2,X_3)=\sum_{X_1}C_2.\phi$	$L_3.\lambda(X_2,X_4)=C_3.\phi$	$P_{BN}(X_3,X_4)=L_1.\lambda(X_3)\times\sum_{X_2}(L_2.\lambda(X_2,X_3)\times L_3.\lambda(X_2,X_4)\times\sum_{X_5}C_6.\phi)$

2.2.5 EJTR 算法描述

本文采用基于扩展邻接树的消元推理算法 EJTR 计算由 Apriori 算法从数据集 D 中产生的频繁属性集合 $FreqA$ 的频繁属性集 I 的边缘概率 $P_{BN}(I)$,$P_{BN}(I)$包含了频繁属性集对应的所有频繁项集(I,i)的边缘概率分布.$FreqI$ 为频繁项集集合,$FreqA$ 为 $FreqI$ 对应的频繁属性集合;$FreqA_1$ 为 $FreqA$ 中属于扩展邻接树中任一子团的频繁属性集构成的子集,$FreqA_1$ 按频繁属性集长度递减排序.$FreqA_2$ 为 $FreqA$ 中不属于扩展邻接树中任一子团的频繁属性集构成的子集,$FreqA_2$ 同样按频繁属性集长度递减排序.

算法 2. EJTR 算法.

输入:贝叶斯网络 BN,频繁项集集合 $FreqI$;

输出:$\{P_{BN}(I):(I,i)\in FreqI\}$到数据表 MP.

1) 按定理 1 证明过程及相应扩展过程构建贝叶斯网络 BN 对应的扩展邻接树 T^*,并对其进行初始化.
2) $FreqA\leftarrow\{I:(I,i)\in FreqI\}$.
3) $FreqA_1\leftarrow\{I:I\in FreqA\wedge(\exists C_i\in T^*\wedge I\subseteq C_i.node)\}$;$FreqA_2\leftarrow\{I:I\in FreqA\wedge I\notin FreqA_1\}$.
4) 如果 $FreqA_1$ 非空:

I. 取 $FreqA_1$ 中的第 1 个属于扩展邻接树中子团 C_i 的频繁属性集 I;

II. 分别计算 I 的边缘概率: $P_{BN}(I)=\sum_{C_i.node-I} C_i.\phi\times L_i.\pi\prod_K L_k.\lambda(L_k.node)$;

III. 计算 I 的所有子集 J 的边缘概率分布: $P_{BN}(J)=\sum_{I-J} P_{BN}(I),\forall(J\subset I)\wedge(J\in FreqA_1)$;

IV. 从 $FreqA_1$ 中移去 I 及 I 的全部子集,转第 4)步.

5) 如果 $FreqA_2$ 非空:

I. 取 $FreqA_2$ 中的第 1 个频繁属性集 I(一定为长频繁属性集);

II. 基于扩展的邻接树 T^* 消元推理过程计算 $P_{BN}(I)$:

A. 从扩展邻接树 T^* 的叶节点出发,每个子团 C_i 收到其全部子节点 C_k 传播过来的消息后:

i. S_k 为每一个子节点 C_k 传播过来的消息 $L_k.\lambda(S_k)$对应的属性子集, $S=\cup_k S_k$;

ii. 若 $L_i.\lambda(S\cup L_i.node\cup(C_i.node\cap I))$已存在,则直接将其传播给 C_i 的父节点 C_j;

iii. 若 $L_i.\lambda(S\cup L_i.node\cup(C_i.node\cap I))$不存在,则

$$L_i.\lambda(S\cup L_i.node\cup(C_i.node\cap I))=\sum_{C_i.node-L_i.node-I} C_i.\phi\prod_k L_k.\lambda(S_k),$$

将其传播给 C_i 的父节点 C_j,并以 $S\cup L_i.node\cup(C_i.node\cap I)$为键值存储在数据表 $L_i.\lambda$中;

B. 一直进行到扩展邻接树的根节点 C_t,并在根节点中计算: $P_{BN}(I)=\sum_{C_t.node-I} C_t.\phi\prod_k L_k.\lambda(S_k)$;

III. 计算属于 $FreqA_2$ 的 I 的全部频繁属性子集 J 的边缘概率分布(不属于 $FreqA_2$ 的属性子集已在 $FreqA_1$ 中计算): $P_{BN}(J)=\sum_{I-J} P_{BN}(I),\forall(J\subset I)\wedge(J\in FreqA_2)$;

IV. 从 $FreqA_2$ 中移去 I 及属于 I 的全部子集 J,转第 5)步.

算法 EJTR 主要通过以下 4 个方面来提高批量推理效率:① 通过对邻接树结构的扩展,使得更多的频繁属性集属于邻接树中的某个子团,然后通过自下而上和自上而下的两次消息传播,在扩展的邻接树处于一致状态下,完成所有属于邻接树中某一子团的频繁属性集的边缘概率计算;② 基于扩展的邻接树,存储每个子团的势函数和消元过程中传播的消息因子,避免批量推理过程中相同消息因子的重复计算;③ 只计算长频繁属性集的边缘概率,长频繁属性集所包含的短频繁属性集通过直接消元进行计算;④ 基于扩展的邻接树选择完备或近似完备的消元次序.

根据算法 EJTR 计算的所有频繁属性集的边缘概率存储在数据表 MP 中,用于频繁项集和频繁属性集的兴趣度的计算,并用于方法 BN-EJTR 的第③步,即,基于贝叶斯网络产生频繁项集.

2.3 从贝叶斯网络中产生频繁项集

根据定义 1~定义 3,为了能够发现所有频繁项集和频繁属性集,方法 BN-EJTR 不仅从数据集 D 产生频繁项集,还从表示其领域知识的贝叶斯网络 BN 中产生频繁项集.方法 BN-EJTR 同样基于 Apriori 算法思想从贝叶斯网络中产生频繁项集,相应的算法记为 Extended-Apriori.与从数据集 D 中产生频繁项集的 Apriori 算法不同的是,Extended-Apriori 算法从贝叶斯网络中计算项集的支持度,项集在贝叶斯网络中的支持度即为该项集在贝叶斯网络中的边缘概率,并通过比较项集在贝叶斯网络的支持度和支持度阈值来产生频繁项集集合 $FreqI'.FreqI'$中的部分频繁项集在数据集 D 中也是频繁项集,这一部分频繁项集在数据集中支持度 $Supp_D(I,i)$和在贝叶斯网络中支持度 $Supp_{BN}(I,i)$已在方法 BN-EJTR 的第①、第②步完成计算,不再需要重复计算.

算法 3. Extended-Apriori 算法.

输入:贝叶斯网络 BN,$FreqI$ 中频繁项集的边缘概率分布表 MP,支持度阈值 Min$Supp$,项集最大属性长度 Max$K$;

输出:频繁项集集合 $FreqI'$.

1) 项集长度 K 初始化为 1:$K\leftarrow 1$.

2) 产生候选 K 项集:$Cand_K\leftarrow\{(I,i):|I|=K,i\in Dom(I)\}$.

3) 从贝叶斯网络 BN 计算候选 K 项集支持度:$\{Supp_{BN}(I,i):(I,i)\in Cand_k\wedge(I,i)\notin MP\}$.

4) 根据支持度阈值产生频繁 K 项集 $\begin{cases}Freq_k \leftarrow \{(I,i):(I,i)\in Cand_k\wedge Supp_{BN}(I,i)\geqslant \mathrm{Min}Supp\}\\ FreqI'\leftarrow Freq_k\end{cases}$.

5) $K\leftarrow K+1$,若 $K>\mathrm{Max}K$,则停止迭代并转第 8)步,否则转下一步.

6) 根据频繁 $K-1$ 项集产生候选 K 项集:$Cand_K\leftarrow\{(I,i):|I|=K,i\in Dom(I)\ \mathrm{From}\ Freq_{k-1}\}$.

7) 转第 3)步.

8) 计算 $FreqI'$中不属于 $FreqI$ 那一部分频繁项集在数据集 D 的支持度$\{Supp_D(I,i):(I,i)\in FreqI'\wedge(I,i)\notin FreqI\}$,并输出 $FreqI'$.

2.4 频繁属性集和频繁项集的剪枝

方法 BN-EJTR 在其第①步和第③步中产生的所有频繁项集均有两个支持度,其中一个是在数据集 D 的支持度,即 $Supp_D(I,i)=P_D(I=i)$;另一个是在贝叶斯网络中的支持度,即 $Supp_{BN}(I,i)=P_{BN}(I=i)$.根据定义 2、定义 3,可以计算出所有频繁项集和属性集的兴趣度,然后按兴趣度对频繁项集和属性集进行排序.用户可根据需要首先选择兴趣度高的频繁模式,并可进一步根据所选择的兴趣度阈值对频繁属性集和频繁项集进行剪枝.

尽管基于贝叶斯网络的领域知识表示具有一致性和全局性,所产生的兴趣度较高的频繁属性集相对于贝叶斯网络具有有趣性,但仅按兴趣度阈值进行剪枝,剪枝后的频繁属性集仍可能存在冗余,通过对频繁属性集和表示领域知识的贝叶斯网络结构的分析发现,有以下两个方面的原因可能导致方法 BN-EJTR 产生的频繁属性集冗余:① 由于属性子集的兴趣度会传递到其属性超集的兴趣度,从而导致一些属性集有趣仅仅是因为它的属性子集有趣;② 因为贝叶斯网络结构所蕴含的因果依赖关系及因果依赖关系的传递性,网络结构中前驱节点所对应的属性集的兴趣度会传递到其后继节点对应的属性集的兴趣度,所以直接导致其后继节点对应的属性集有趣.为了能够发现真正有趣的频繁属性集,本文按定义 5~定义 7 给出频繁属性集和频繁项集的层次有趣性和拓扑有趣性的定义,并提出一种同时基于兴趣度阈值和拓扑有趣性的剪枝算法.

定义 5. 如果一个频繁属性集的兴趣度大于其所有频繁属性子集的兴趣度,则称该频繁属性集满足层次有趣性;如果一个频繁项集的兴趣度大于该频繁项集对应频繁属性集所有频繁属性子集的兴趣度,则称该频繁项集满足层次有趣性.

例如:假设频繁属性集(X_1,X_2)有趣,且属性 X_3 独立于(X_1,X_2)(若属性 X_3 依赖(X_1,X_2),则属于拓扑有趣的范畴),即使 $P_{BN}(X_3)=P_D(X_3)$,频繁属性集$\{X_1,X_2,X_3\}$也可能是有趣的,只有满足条件 $Int_{BN}((X_1,X_2,X_3))>Int_{BN}((X_1,X_2))$的条件下,才能保证频繁属性集$(X_1,X_2,X_3)$有趣不仅仅是因为其频繁属性子集$(X_1,X_2)$有趣.层次有趣性可以防止一个频繁属性集或频繁项集有趣仅仅是因为其属性子集有趣的情况出现,同时也有助于发现兴趣度高的短频繁属性集和频繁项集.

定义 6. I 为一频繁属性集,$Anc(I)$为 I 所对应的节点集在贝叶斯网络中的祖先节点集,若不存在同时满足以下 3 个条件的频繁属性集 J:① J 是 I 和 I 的祖先节点集 $Anc(I)$的并集的一个子集;② I 不是 J 的子集;③ J 的兴趣度大于等于 I 的兴趣度,则称频繁属性集 I 满足拓扑有趣性.

定义 7. (I,i)为一频繁项集,$Anc(I)$为 I 所对应的节点集在贝叶斯网络中的祖先节点集,若不存在同时满足以下 3 个条件的频繁属性集 J:① J 是 I 和 I 的祖先节点集 $Anc(I)$的并集的一个子集;② I 不是 J 的子集;③ J 的兴趣度大于等于(I,i)的兴趣度,则称频繁项集(I,i)满足拓扑有趣性.

基于贝叶斯网络的有趣性还会出现因为前驱节点集有趣而导致后继节点集有趣的情况,从而导致频繁属性集冗余.例如,对于网络结构 $X_1\rightarrow X_2$,如果属性 X_1 是有趣的,即使 $P_{BN}(X_2|X_1)=P_D(X_2|X_1)$,也可能会导致属性 X_2 和属性集(X_1,X_2)有趣.

若一个频繁项集(I,i)满足拓扑有趣性,其对应的频繁属性集 I 一定不存在满足定义 6 中 3 个条件的频繁属性子集 J,则频繁属性集 I 一定满足拓扑有趣性,因此只需从满足拓扑有趣性的频繁属性集中去寻找满足拓扑有趣性的频繁项集.若一个频繁属性集 I 不满足层次有趣性,则一定存在同时满足拓扑有趣性定义中 3 个条件的属性集 J,频繁属性集 I 一定不满足拓扑有趣性,也即:满足拓扑有趣性的频繁属性集一定满足层次有趣性.同理,满

足拓扑有趣性的频繁项集也一定满足层次有趣性.因此,拓扑有趣性是一种比层次有趣性要求更高的性质,按拓扑有趣性剪枝后保留下来的频繁属性集和频繁项集一定是既满足层次有趣性又满足拓扑有趣性.

方法 BN-EJTR 提出了一种基于兴趣度阈值和拓扑有趣性的剪枝算法 Prune-Topo,该算法首先根据兴趣度阈值对频繁属性集和频繁项集进行剪枝,然后根据频繁属性集拓扑有趣性对频繁属性集进行剪枝,再从满足拓扑有趣性的频繁属性集中寻找所有满足拓扑有趣性的频繁项集.大于兴趣度阈值且满足层次有趣性和拓扑有趣性的频繁属性集和频繁项集为最终保留下来的频繁属性集和频繁项集.

方法 BN-EJTR 在频繁属性集的产生过程中始终保持频繁属性集按属性编号有序,即类似如(X_3,X_2)的频繁属性以(X_2,X_3)形式存储.为了提高在剪枝过程中对频繁属性子集的搜索效率,算法 Prune-Topo 将所有的有趣频繁属性集表示成一棵频繁属性树.例如,有趣频繁属性集合$\{(X_1,X_2,X_3),(X_1,X_2),(X_1,X_3),(X_2,X_3)\}$对应的频繁属性树如图 4 所示,树节点的 Item 项表示对应的频繁属性集,若为 None,则表示从树的根节点到该节点的路径对应的属性集不构成有趣的频繁属性集.算法 Prune-Topo 在剪枝阶段设计频繁属性子集的搜索算法 *Iter_included*(*ST*, *Supset*),用于在频繁属性树 *ST* 中搜索属性超集 *Supset* 的频繁属性子集.

算法 4. Prune-Topo 算法.

输入:贝叶斯网络 *BN*,频繁属性集集合 *FreqA*,兴趣度阈值ε;

输出:频繁属性集集合 *FreqAP*,*FreqATP*;频繁项集集合 *FreqIP*,*FreqITP*.

1) 计算 FreqA 中所有频繁属性集兴趣度:$FreqA' \leftarrow \{(I, Int_{BN}(I)): I \in FreqA\}$.

2) 按兴趣度阈值剪枝:$\begin{cases} FreqAP \leftarrow \{(I, Int_{BN}(I)): (I, Int_{BN}(I)) \in FreqA' \wedge Int_{BN}(I) \geqslant \varepsilon\} \\ FreqIP \leftarrow \{((I,i), Int_{BN}(I,i)): (I, Int_{BN}(I)) \in FreqAP \wedge Int_{BN}(I,i) \geqslant \varepsilon\} \end{cases}$.

3) 拓扑有趣集置空:$FreqATP \leftarrow \varnothing$;$FreqITP \leftarrow \varnothing$.

4) 根据频繁有趣集 *FreqAP* 建立频繁属性树:$ST \leftarrow Setree(FreqAP)$.

5) 如果 *FreqAP* 非空:

I. 从 *FreqAP* 中取出频繁属性集$(I, Int_{BN}(I))$;

II. *J* 为 *I* 和 *I* 在贝叶斯网络中祖先节点的并集:$J \leftarrow I \cup Anc(I)$;

III. *SJ* 为 *J* 所包含的全部频繁属性子集:$SJ \leftarrow Iter\_included(ST, J)$;

IV. *S* 为 *SJ* 中不包含频繁属性 *I* 且兴趣度大于等于 *I* 的元素集合

$$S \leftarrow \{L: (L \in SJ) \wedge (I \not\subset L) \wedge (Int_{BN}(L) \geqslant Int_{BN}(I))\}$$

V. 若 *S* 为空,则 *I* 为拓扑有趣集:

A. $FreqATP \leftarrow (I, Int_{BN}(I))$;

B. 按频繁项集拓扑有趣性定义依次判断 *I* 对应的频繁项集(I,i)是否满足拓扑有趣性,若满足,则

$$FreqITP \leftarrow ((I,i), Int_{BN}(I,i))$$

VI. 转第 5)步.

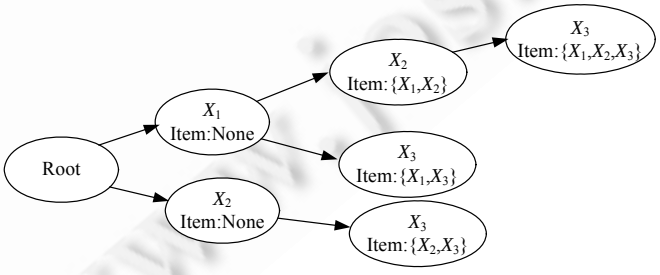

Fig.4 Frequent attribute tree corresponding to frequent attribute sets $\{(X_1,X_2,X_3),(X_1,X_2),(X_1,X_3),(X_2,X_3)\}$

图 4 频繁属性集$\{(X_1,X_2,X_3),(X_1,X_2),(X_1,X_3),(X_2,X_3)\}$对应的频繁属性树

3 实验结果及其分析

为了验证方法 BN-EJTR 具有良好的时间性能并能发现真正有趣的频繁项集,本文针对数据集 KSL, Lymphography,Soybean 和 Splice 进行了一系列的实验.其中:KSL 数据集是一个关于丹麦人身体健康状况和生活方式的数据集[25],该数据集包含 FEV(forced objection volume of person's lungs),Kol(cholesterol),Hyp (hypertension: no/yes),BMI(body mass index),Smok(smoking: no/yes),Alc(alcohol consumption: seldom/frequently),Work(working: yes/no),Sex(male/female),Year(survey year: 1967/1984)等 9 个属性;Soybean, Lymphography 和 Splice 均为来自 UCI 的标准数据集.标准数据集很少提供领域知识,本文同时采用 ISOR 算法[26]和 B-Course 算法[27]从数据集 D 中学习相应的贝叶斯网络结构 BN,并选择其中最优的网络结构作为表示相应数据集领域知识的网络结构.KSL 数据集对应的贝叶斯网络结构如图 5 所示.

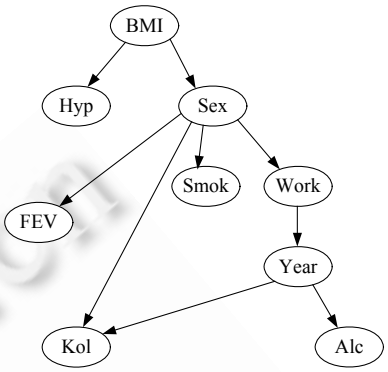

Fig.5 KSL network

图 5 KSL 网络

以下实验均在 Intel(R) Core(TM)2 CPU 1.6GHz,2G 内存的个人笔记本电脑上完成,操作系统为 WindowsVista™.实验中选择的支持度阈值 Min*Supp* 均为 0.01,ApproxInter 方法中的兴趣度阈值 ε=0.01,显著性水平 δ=0.05 和 K=5.以下各表中,N 表示数据集 D 所包含属性个数,Max$K$ 表示单个频繁项集或频繁属性集中属性集的最大长度,*Nmarginal* 表示在贝叶斯网络中计算边缘概率的次数,*Time*[*s*]表示以秒为单位的时间长度,*Interestingness* 表示频繁项集或频繁属性集的兴趣度,Max*Int* 和 *A*Max*Int* 分别表示所有频繁项集兴趣度最大值的精确值和近似值,ε表示兴趣度阈值,*FreqA* 和 *FreqI* 分别表示剪枝前的频繁属性集和频繁项集,*FreqAP* 和 *FreqIP* 分别表示按兴趣度阈值ε剪枝后的频繁属性集和频繁项集, *FreqATP* 和 *FreqITP* 分别表示按拓扑有趣性剪枝后的频繁属性集和频繁项集,*Pruning Rate* 为剪枝后的频繁模式集长度和剪枝前频繁模式集的长度之比.

3.1 时间性能分析

下面对方法 ExactInter,ApproxInter,PJC 和本文提出的方法 BN-EJTR 的时间性能进行分析比较.ApproxInter 方法不直接进行贝叶斯网络的推理,而是通过对贝叶斯网络进行抽样得到的数据集进行计数来近似计算属性集在贝叶斯网络中的支持度.该方法通过反复迭代和顺序抽样,在迭代过程收敛之后能在一定置信区间和误差范围之内得到前 K 个频繁属性集,该方法的时间性能取决于为满足一定误差约束而进行的迭代过程的收敛速度.PJC 方法通过调用 NeticaJ 接口中函数 getFindingsProbability()计算频繁项集在贝叶斯网络中的支持度,一个频繁属性集对应的不同候选项集在贝叶斯网络中的支持度要通过多次计算来完成,其时间性能取决于数据规模和频繁项集的长度.ExactInter 采用桶消元推理算法 BEI(bucket elimination inferening)计算批量频繁属性集中不同的 I 在贝叶斯网络中的边缘概率 $P_{BN}(I)$,其时间性能取决于数据规模和频繁属性集的长度.

在以上 4 种方法中,只有 ExactInter 和 BN-EJTR 采用不同的精确推理算法 BEI 和 EJTR 计算批量频繁属性集在贝叶斯网络中的支持度.图 6 比较了 BEI 算法和 EJTR 算法的时间性能,记录的是在 Lymphography 数据集上,针对与不同 *MaxK* 对应的候选频繁属性集长度 *NMarginal*,算法 EJTR 和 BEI 计算 *NMarginal* 次边缘概率所需要的时间.从图 6 可以看出,随着边缘概率计算次数 *Nmarginal* 的增加,算法 EJTR 相对于算法 BEI 的时间性能优势明显.通过对图 6 的进一步分析还发现:算法 EJTR 的推理时间增加速度明显慢于边缘概率计算次数的增加速度,原因在于算法 EJTR 在消元过程中传播的相同消息因子只计算一次,边缘概率的计算次数越多,消息因子的复用程度越高;同时,算法 EJTR 只计算长频繁属性集的边缘概率分布 $P_{BN}(I)$,而它所包含的频繁属性子集的边缘概率通过按公式(4)直接消元获得,单个频繁属性集越长,被包含的频繁属性子集也就越多.

在同一数据集 Soybean 上,针对不同的频繁属性集最大长度 Max$K$,图 7 比较了以上 4 种不同方法的时间性能.图 7 中的第 1 个图描述的是当 Max$K$ 从 1 变化到 3 时,4 种不同方法的时间性能比较.可以看出,当 Max$K$ 较

小时,方法 ApproxInter 在存在计算误差的情况下时间性能也最差,因为该方法要经过反复迭代才能满足误差约束,处理中小规模的频繁属性集不具有时间性能上的优势;方法 BN-EJTR 的时间性能最优,尽管在 Max$K$ 等于 1 和 2 时,该方法和 PJC 方法的时间性能的相近,但二者随 Max$K$ 的变化趋势是不同的.图 7 中第 2 个图描述的是当 Max$K$ 从 1 变化到 5 时,4 种不同方法的时间性能比较.从该图可以看出:方法 PJC 的时间性能下降最快,当 Max$K$=4 时,该方法在本机上已无法在有效长的时间内完成计算,原因在于方法 PJC 是直接计算频繁项集在贝叶斯网络中支持,一个频繁属性集所对应的不同频繁项集要通过多次函数调用才能完成,而随着 Max$K$ 的增加,频繁项集迅速增多;随着 Max$K$ 的增加,相比于方法 ApproxInter,BN-EJTR 的时间性能优势不再明显,但 BN-EJTR 方法是基于精确推理的频繁模式兴趣度的精确计算方法,而方法 ApproxInter 是基于样本数据的频繁模式兴趣度的一种近似计算.

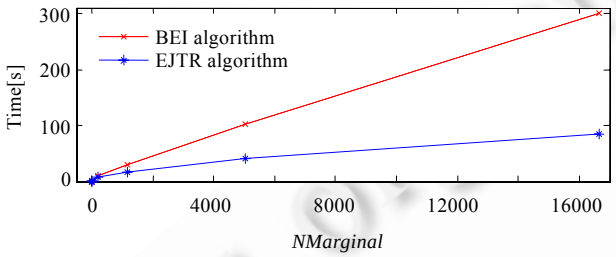

Fig.6 Time performance comparison between BEI and EJTR

图 6 BEI 和 EJTR 的时间性能比较

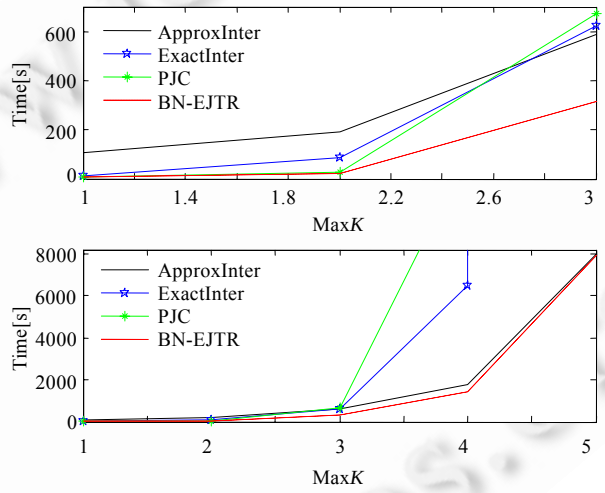

Fig.7 Time performance comparison among ExactInter, ApproxInter, PJC and BN-EJTR

图 7 ExactInter,ApproxInter,PJC 和 BN-EJTR 的时间性能比较

方法 ExactInter,ApproxInter,PJC 和 BN-EJTR 的时间性能均同时受数据集 D 和 Max$K$ 的影响.针对不同的数据集和 Max$K$,表 3 对方法 ExactInter,ApproxInter,PJC 和 BN-EJTR 的时间性能进行了比较,该表 *time*[s]栏中的时间不包括贝叶斯网络的建立时间和剪枝时间.*time*[s]栏中,－表示在本机上无法在有限长的时间内得到运算结果.从表 3 中可以看出:与基于精确推理的方法 ExactInter 和 PJC 相比,BN-EJTR 的具有明显的时间性能优势;且随着数据库规模和 Max$K$ 的增大,需要在贝叶斯网络中计算边缘概率的次数越多,方法 BN-EJTR 的优势越明显.与近似方法 ApproxInter 相比,在一定规模的数据集和 Max$K$ 范围内,方法 BN-EJTR 在保证计算精度的前提下在时间性能上仍具有一定优势.

Table 3 Time performance comparison among ExactInter, ApproxInter, PJC and BN-EJTR

表 3 ExactInter,ApproxInter,PJC 和 BN-EJTR 的时间性能比较

D	Max*K*	*NMarginal*	ApproxInter		ExactInter		PJC		BN-EJTR	
			AMaxInt	*Time*[s]	Max*Int*	*Time*[s]	Max*Int*	*Time*[s]	Max*Int*	*Time*[s]
KSL (*N*=9)	4	256	0.033 36	30	0.031 90	2.10	0.031 90	1.87	0.031 90	1.89
KSL	5	382	0.032 29	33	0.031 90	2.29	0.031 9	2.77	0.031 9	2.03
Lymphography (*N*=19)	3	1 160	0.099 43	58	0.123 08	32	0.123 08	29	0.123 08	15
Lymphography	4	5 036	0.124 3	110	0.126 31	113	0.126 31	150	0.126 31	37
Soybean (*N*=36)	3	7 807	0.071 85	590	0.063 0	625	0.063 0	674	0.063 0	314
Soybean	4	66 707	0.063 88	1 775	0.070 65	6 467	0.070 65	12 757	0.070 65	1426
Soybean	5	443 505	0.078 06	7 985	—	—	—	—	0.076 72	7 945
Splice (*N*=61)	3	37 882	0.036 52	1 997	0.036 43	8 765	—	—	0.036 43	1 754

3.2 剪枝效果分析

方法 BN-EJTR 基于贝叶斯网络计算所有频繁项集的兴趣度,用户可以按兴趣度对所有的频繁项集进行排序并选取感兴趣的频繁项集(按兴趣度阈值或指定数目).取 Max*K*=5,对方法 BN-EJTR 在 KSL 数据集上产生的频繁项集集合(长度为 8 688)按兴趣度降序排序,排在前五位和后五位的频繁项集见表 4 和表 5 所示.表中 *Index-Supp*$_D$(*I*,*i*)栏给出了同一频繁项集在从数据集 KSL 中产生的频繁项集集合(总长度为 8 254)中按支持度降序排列所在的位置,*Index-Supp*$_{BN}$(*I*,*i*)栏给出了同一频繁项集在从贝叶斯网络 *BN* 中产生的频繁项集集合(总长度为 8 242)中按支持度降序排列所在的位置.比较表 4、表 5 中的频繁项集在 KSL 数据集中支持度绝对大小及相对位置、在 KSL 网络中支持度绝对大小及相对位置和其兴趣度大小之间的关系发现,方法 BN-EJTP 挖掘出来的兴趣度大的频繁项集仅从数据集 *D* 或贝叶斯网络 *BN* 都是不太容易被发现的,而且兴趣度小的频繁项集仅从数据集 *D* 或贝叶斯网络 *BN* 都是很容易被发现的.

Table 4 Top 5 frequent item sets from KSL dataset

表 4 KSL 数据集上的前 5 个频繁项集

FreqI	*Interestingness*	*Supp*$_D$(*I*,*i*)	*Index-Supp*$_D$(*I*,*i*)	*Supp*$_{BN}$(*I*,*i*)	*Index-Supp*$_{BN}$(*I*,*i*)
([Hyp,Smok,Year],[0,2,2])	0.031 90	0.257 62	82	0.225 72	116
([FEV,Alc,Year],[0,1,1])	0.030 70	0.097 87	725	0.067 17	1 341
([FEV,Hyp,Alc,Year],[0,1,1,1])	0.030 24	0.067 41	1 365	0.037 17	2 993
([FEV,Smok,Year],[0,1,2])	0.029 89	0.041 55	2 651	0.071 44	1 226
([FEV,Alc,Work,Year],[0,1,2,1])	0.029 69	0.083 10	964	0.053 41	1 925

Table 5 Last 5 frequent item sets from KSL dataset

表 5 KSL 数据集上的后 5 个频繁项集

FreqI	*Interestingness*	*Supp*$_D$(*I*,*i*)	*Index-Supp*$_D$(*I*,*i*)	*Supp*$_{BN}$(*I*,*i*)	*Index-Supp*$_{BN}$(*I*,*i*)
([BMI,Sex],[0,2])	2.77555756156e−017	0.172668513389	232	0.172668513389	224
([Hyp,BMI],[0,0])	5.55111512313e−017	0.182825484765	196	0.182825484765	197
([FEV],[2])	5.55111512313e−017	0.286241920591	61	0.286241920591	60
([Work,Sex],[2,1])	5.55111512313e−017	0.409972299169	23	0.409972299169	23
([Smok,Sex],[2,1])	5.55111512313e−017	0.459833795014	14	0.459833795014	14

在常规频繁项集挖掘方法中,如果选择支持度阈值为 0.1,仅从 KSL 数据集或 KSL 网络进行挖掘,表 4 中除第 1 个频繁项集之外的其他频繁项集都不能被发现;而表 5 中的频繁项集却都能被发现.例如:表 4 中频繁项集([FEV,Alc,Year],[0,1,1])在数据集 KSL 中的支持度为 0.09787,排在第 725 位;在图 5 所示 KSL 网络中的支持度为 0.06717,排在第 1341 位;而表 5 中的频繁项集([Smok,Sex],[2,1])在数据集 KSL 中的支持度为 0.459833795014,排在第 14 位;在 KSL 网络中的支持度为 0.459833795014,也排在第 14 位.如果在常规挖掘方法中支持度阈值为 0.1,([FEV,Alc,Year],[0,1,1])是不能被发现的,而([Smok,Sex],[2,1])却可以被发现.事实上,([Smok,Sex],[2,1])是无趣的,因为抽烟和男性之间的关系是人所共知的常识;而([FEV,Alc,Year],[0,1,1])所对应的肺功能指标、饮酒及某年份之间的关系则是人们感兴趣的,因为在常识中,饮酒和肺功能之间是没有直接关系的.

通过对表 4 中频繁项集和图 5 所示的贝叶斯网络的比较分析发现:表 4 中频繁项集均不对应网络结构中所

蕴含的因果依赖关系,且均不构成马尔可夫毯;表 5 中的频繁项集的兴趣度均近似为 0,进一步的分析发现,表 5 中的频繁项集均对应贝叶斯网络中单节点和因果关系.事实上,贝叶斯网络中的单节点对应的所有频繁 1 项集中,(Kol,1)的兴趣度最大,其兴趣度为 0.000674149413842,在所有频繁项集中排在倒数第 39 位;贝叶斯网络中蕴含的直接因果关系对应的所有频繁 2 项集中,([Alc,Sex],[1,1])的兴趣度最大,其兴趣度为 0.0022260501597,排在倒数第 63 位.从而说明了兴趣度较大的频繁项集表示是与当前领域知识不一致的知识,相对于当前的领域知识是有趣的;兴趣度较小的频繁项集表示与当前领域知识一致的知识,兴趣度表示了频繁项集的有趣程度.

分析表 4 中的频繁项集发现,尽管兴趣度高的频繁项集具有有趣性,但存在层次和拓扑结构上的冗余性.例如:表 4 中频繁项集([FEV,Alc,Year],[0,1,1])的兴趣度大于频繁项集([FEV,Alc,Work,Year],[0,1,2,1])的兴趣度,而在 KSL 网络中,Work 是 Year 的父节点,根据定义 7,频繁项集([FEV,Alc,Work,Year],[0,1,2,1])不满足拓扑有趣性;又因为([FEV,Alc,Year],[0,1,1])是([FEV,Alc,Work,Year],[0,1,2,1])的子集,根据定义 5,频繁项集([FEV,Alc,Work,Year],[0,1,2,1])也不满足层次有趣性.为了能够发现真正有趣的频繁项集,方法 BN-EJTR 不仅按照兴趣度阈值ε进行剪枝,而且按照频繁属性集和频繁项集的拓扑有趣性进行剪枝.为了验证方法 BN-EJTR 的剪枝效果,针对不同兴趣度阈值ε,并按频繁属性集和频繁项集的拓扑有趣性对 KSL 和 Soybean 数据集上的频繁属性集和频繁项集进行剪枝,剪枝后保留下来频繁属性集集合长度、频繁项集集合长度和剪枝率见表 6.KSL(Max$K$=5)和 Soybean(Max$K$=3)数据集在剪枝前的频繁属性集集合和频繁项集集合的长度分别为

$$Length(FreqA)|KSL=382, Length(FreqI)|KSL=8688;$$

$$Length(FreqA)|Soybean=7807, Length(FreqI)|Soybean=91840.$$

显然,方法 BN-EJTR 的剪枝效果明显.

Table 6 Pruning performance of BN-EJTR on different datasets

表 6 BN-EJTR 针对不同数据集的剪枝效果

D	ε	*FreqAP*		*FreqIP*		*FreqATP*		*FreqITP*	
		Length	Pruning rate (%)	*Length*	Pruning rate (%)	*Length*	Pruning rate (%)	*Length*	Pruning rate (%)
KSL (Max$K$=5)	0.01	307	80	886	10	21	5.5	86	0.98
	0.015	141	37	231	2.7	16	4.2	39	0.45
	0.02	47	12	58	0.67	9	2.4	14	0.16
	0.025	14	3.7	16	0.18	6	1.6	8	0.09
Soybean (Max$K$=3)	0.01	4 933	63	15 699	17	23	0.29	172	0.19
	0.02	1 811	23	3 494	3.8	18	0.23	67	0.072
	0.03	709	9.1	1 086	1.2	18	0.23	28	0.030
	0.04	192	2.5	233	0.25	9	0.12	10	0.011
	0.05	15	0.19	15	0.016	2	0.026	2	0.002

为了对剪枝后频繁属性集和频繁项集进行分析,表 7 给出了 Max$K$=5,ε=0.025 时,按兴趣度阈值和拓扑有趣性对 KSL 数据集上的频繁属性集和频繁项集剪枝后保留下来的 6 个频繁属性集和 8 个频繁项集.通过对表 7 中频繁项集、频繁属性集和图 5 所示的 KSL 网络结构的分析发现:剪枝后的频繁属性集和频繁项集均满足层次有趣性和拓扑有趣性;同时,剪枝后的频繁属性集很短,便于进行进一步的分析和使用.分析发现,Year 和 FEV 是 KSL 数据集中应该引起足够重视的属性,剪枝后所有频繁属性集均包含 Year 属性,说明当时的政治经济环境对人们的身体健康指标产生重要而潜在的影响.

Table 7 Frequent attribute sets and frequent item sets from dataset KSL after pruning

表 7 KSL 数据集上剪枝后的频繁属性集和频繁项集

ε	*FreqATP*	*FreqITP*	*Interestingness*	*Index-Supp*$_D$(I,i)	*Index-Supp*$_{BN}$(I,i)
0.025	[Hyp,Smok,Year]	([Hyp,Smok,Year],[0,2,2])	0.031 90	82	116
		([Hyp,Smok,Year],[1,2,2])	0.247 46	93	71
	[FEV,Alc,Year]	([FEV,Alc,Year],[0,1,1])	0.030 70	725	1 341
	[FEV,Smok,Year]	([FEV,Smok,Year],[0,1,2])	0.029 89	2 651	1 226
	[FEV,Hyp,Year]	([FEV,Hyp,Year],[0,1,1])	0.028 72	1 130	2 340
	[FEV,Kol,Year]	([FEV,Kol,Year],[0,1,2])	0.028 18	1 365	742
	[FEV,Year]	([FEV,Year],[0,1])	0.027 42	574	955
		([FEV,Year],[0,2])	0.027 42	221	128

4 结论与展望

基于贝叶斯网络的知识表示方法具有全局性和一致性的特点,本文采用贝叶斯网络表示领域知识,研究了基于贝叶斯网络的频繁项集和属性集兴趣度的计算和剪枝方法.方法 BN-EJTR 的创新之处主要包括以下两个方面:① 针对本问题的批量推理需求,提出了一种基于扩展邻接树选择消元次序的推理算法,用于计算频繁项集和频繁属性集在贝叶斯网络中的支持度.该算法不仅能够基于扩展的邻接树选择完备或近似完备的消元次序,而且通过在扩展邻接树中存储消元过程中传播的消息,避免了批量推理过程中传播的相同消息因子的重复计算;② 通过对频繁模式和贝叶斯网络结构分析,提出了一种基于兴趣度阈值和拓扑有趣性对频繁属性集和频繁项集进行剪枝的剪枝算法.在标准数据集上的实验结果也表明:该方法与同类方法相比具有良好的时间性能;同时,该方法对频繁属性集和频繁项集的剪枝效果明显;而且经过分析发现,剪枝后的频繁项集满足层次有趣性和拓扑有趣性.

今后进一步工作的方向:① 尝试将在理论上具有收敛性保证的贝叶斯网络的近似推理算法 MCMC 引入基于贝叶斯网络的批量频繁项集和属性集的兴趣度计算,进一步提高此类方法的时间性能,以解决针对更大规模数据集的频繁项集和属性集的兴趣度计算问题;② 进行基于有趣属性集,迭代优化贝叶斯网络结构的尝试,探索贝叶斯网络结构学习的新思路;③ 针对频繁模式的挖掘问题,探索更有效的表示领域知识的方法和模型.

References:

[1] Han JW, Cheng H, Xin D, Yan XF. Frequent pattern mining: Current status and future directions. Data Mining Knowledge Discovery, 2007,15(1):55−86. [doi: 10.1007/s10618-006-0059-1]

[2] Zhang H, Padmanabhan B, Tuzhilin A. On the discovery of significant statistical quantitative rules. In: Kim W, Kohavi R, Gehrke J, DuMouchel W, eds. Proc. of the 10th ACM-SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2004.374−383. [doi: 10.1145/1014052.1014094]

[3] Cheng H, Yan XF, Han JW, Hsu CW. Discriminative frequent pattern analysis for effective classification. In: Proc. of the 23rd Int'l Conf. on Data Engineering. Los Alamitos: IEEE Computer Society Press, 2007. 716−725. [doi: 10.1109/ICDE.2007.367917]

[4] Wang HX, Wang W, Yang J, Yu PS. Clustering by pattern similarity in large data sets. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 28th ACM-SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2002. 394−405. [doi: 10.1145/564691.564737]

[5] Mcgarry K. A survey of interestingness measures for knowledge discovery. Knowledge Engineering Review, 2005,20(1):39−61.

[6] Jaroszewicz S, Simovici DA. Pruning redundant association rules using maximum entropy principle. In: Cheng MS, Yu PS, Liu B, eds. Proc. of the 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. LNCS 2336, Heidelberg: Springer-Verlag, 2002. 135−147. [doi: 10.1007/3-540-47887-6_13]

[7] Zaki MJ. Generating non-redundant association rules. In: Proc. of the 6th ACM-SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2000. 34−43. [doi: 10.1145/347090.347101]

[8] Huang MX, Yan XW, Zhang SC. Query expansion of pseudo relevance feedback based on matrix-weighted association rules mining. Journal of Software, 2009,20(7):1854−1865 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3368.htm [doi: 10.3724/SP.J.1001.2009.03368]

[9] Yao H, Hamilton HJ. Mining itemset utilities from transaction databases. Data & Knowledge Engineering, 2006,59(3):603−626. [doi: 10.1016/j.datak.2005.10.004]

[10] Blanchard J, Guillet F, Gras R, Briand H. Using information-theoretic measures to assess association rule interestingness. In: Proc. of the 5th Int'l Conf. on Data Mining. Washington: IEEE Computer Society, 2005. 66−73. [doi: 10.1109/ICDM.2005.149]

[11] Carvalho DR, Freitas AA, Ebecken N. Evaluating the correlation between objective rule interestingness measures and real human interest. In: Jorge A, Torgo L, Brazdil P, Camacho R, Gama J, eds. Proc. of the 9th European Conf. on Principles of Data Mining and Knowledge Discovery. LNCS 3721, Heidelberg: Springer-Verlag, 2005. 453−461. [doi: 10.1007/11564126_45]

[12] Ohsaki M, Kitaguchi S, Okamoto K, Yokoi H, Yamaguchi T. Evaluation of rule interestingness measures with a clinical dataset on hepatitis. In: Boulicaut JF, Esposito F, Giannotti F, Pedreschi D, eds. Proc. of the 8th European Conf. on Principles of Data Mining and Knowledge Discovery. LNCS 3202, Heidelberg: Springer-Verlag, 2004. 362−373. [doi: 10.1007/978-3-540-30116-5_34]

[13] Padmanabhan B, Tuzhilin A. Small is beautiful: discovering the minimal set of unexpected patterns. In: Proc. of the 6th ACM-SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2000. 54−63. [doi: 10.1145/347090.347103]

[14] Padmanabhan B, Tuzhilin A. On characterization and discovery of minimal unexpected patterns in rule discovery. IEEE Trans. on Knowledge Data Engineer, 2006,18(2):202−216. [doi: 10.1109/TKDE.2006.32]

[15] Ding GT. Data mining method based on Bayesian network and its applications to gene expression analysis [MS. Thesis]. Tianjin: Naikai University, 2004 (in Chinese with English abstract).

[16] Jaroszewicz S, Simovici DA. Interestingness of frequent itemsets using Bayesian networks as background knowledge. In: Kim W, Kohavi R, Gehrke J, DuMouchel W, eds. Proc. of the 10th ACM-SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2004. 178−186. [doi: 10.1145/1014052.1014074]

[17] Malhas R, Aghbari ZA. Fast discovery of interesting patterns based on Bayesian network background knowledge. University of Sharjah Journal of Pure & Applied Sciences, 2007,4(3):29−47.

[18] Jaroszewicz S, Scheffer T. Fast discovery of unexpected patterns in data relative to a Bayesian network. In: Grossman R, Bayardo RJ, Bennett KP, eds. Proc. of the 11th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2005. 118−127. [doi: 10.1145/1081870.1081887]

[19] Jaroszewicz S, Scheffer T, Simovici DA. Scalable pattern mining with Bayesian networks as background knowledge. Data Mining and Knowledge Discovery, 2008,18(1):56−100. [doi: 10.1007/s10618-008-0102-5]

[20] Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. In: Buneman P, Jajodia S, eds. Proc. of the 18th ACM-SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 1993. 207−216. [doi: 10.1145/170035.170072]

[21] Cooper GF. The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence, 1990, 42(2-3):393−405. [doi: 10.1016/0004-3702(90)90060-D]

[22] Wang RG. Research on the theory of Bayesian network and its application in object detection [Ph.D. Thesis]. Hefei: Hefei University of Technology, 2004 (in Chinese with English abstract).

[23] Jensen FV, Nielsen TD. Bayesian Networks and Decision Graphs. 2nd ed., New York: Springer-Verleg, 2007. 109−166. [doi: 10.1007/s00362-009-0201-4]

[24] Kask K, Dechter R, Larrosa J, Dechter A. Unifying tree decompositions for reasoning in graphical models. Artificial Intelligence, 2005,166(1-2):165−193. [doi: 10.1016/j.artint.2005.04.004]

[25] Bøttcher SG, Dethlefsen C. Deal: A package for learning Bayesian networks. http://www.jstatsoft.org/v08/i20/paper

[26] Hu XG, Hu CL. A dependency analysis based algorithm for learning Bayesian networks. Pattern Recognition and Artificial Intelligence, 2006,19(4):445−449 (in Chinese with English abstract).

[27] Myllymäki P, Silander T, Tirri H, Uronen P. B-Course: A Web-based tool for Bayesian and causal data analysis. Int'l Journal on Artificial Intelligence Tools, 2002,11(3):369−387. [doi: 10.1142/S0218213002000940]

附中文参考文献:

[8] 黄名选,严小卫,张师超.基于矩阵加权关联规则挖掘的伪相关反馈查询扩展.软件学报,2009,20(7):1854−1865. http://www.jos.org.cn/1000-9825/3368.htm [doi: 10.3724/SP.J.1001.2009.03368]

[15] 丁贵涛.基于贝叶斯网络的数据挖掘方法及其基因表达分析应用[硕士学位论文].天津:南开大学,2004.

[22] 汪荣贵.Bayes 网络理论及其在目标检测中应用研究[博士学位论文].合肥:合肥工业大学,2004.

[26] 胡学钢,胡春玲.一种基于依赖分析的贝叶斯网络结构学习算法.模式识别与人工智能,2006,19(4):445−449.

胡春玲(1970－),女,安徽枞阳人,博士生,讲师,CCF 会员,主要研究领域为数据挖掘,贝叶斯网络.

胡学钢(1961－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为知识工程,数据挖掘.

吴信东(1963－),男,博士,教授,博士生导师,主要研究领域为数据挖掘,专家系统,万维网信息处理.

姚宏亮(1972－),男,博士,副教授,主要研究领域为贝叶斯网络.