



文章编号:1001-9081(2017)11-3107-08

DOI:10.11772/j.issn.1001-9081.2017.11.3107

基于最大团的条件偏好挖掘

谭征, 刘惊雷*, 余航

(烟台大学 计算机与控制工程学院, 山东 烟台 264005)

(*通信作者电子邮箱 jinglei_liu@sina.com)

摘要:针对在数据库的个性化查询中条件约束(或上下文约束)没有被充分考虑的问题,首先提出了条件约束模型 $i^+ > i^- | X$, 它表示在上下文 X 的约束下, 相对于 i^- , 用户更偏好 i^+ 。在此模型的基础上, 采用最大团 (MaxClique) 关联规则算法挖掘获得用户偏好; 随后又提出了条件偏好挖掘 (CPM) 算法, 该算法结合上下文用于挖掘偏好规则, 从而得出用户的偏好。实验结果表明, 基于 CPM 算法的偏好挖掘模型具有较强的偏好表达能力, 将 CPM 算法与基于 Apriori 的算法以及 CONTEMUM 算法进行了实验对比, 实验的主要参数为最小支持度、最小可信度、数据规模等, 实验结果进一步表明所提出的 CPM 算法可明显提高用户偏好规则的产生效率。

关键词:最大团; 关联规则; 偏好数据库; 条件偏好规则; 偏好挖掘

中图分类号:TP311 文献标志码:A

Conditional preference mining based on MaxClique

TAN Zheng, LIU JingLei*, YU Hang

(School of Computer and Control Engineering, Yantai University, Yantai Shandong 264005, China)

Abstract: In order to solve the problem that conditional constraints (context constraints) for personalized queries in database were not fully considered, a constraint model was proposed where the context $i^+ > i^- | X$ means that the user prefers i^+ than i^- based on the constraint of context X . Association rules mining algorithm based on MaxClique was used to obtain user preferences, and Conditional Preference Mining (CPM) algorithm combined with context obtained preference rules was proposed to obtain user preferences. The experimental results show that the context preference mining model has strong preference expression ability. At the same time, under the different parameters of minimum support, minimum confidence and data scale, the experimental results of preferences mining algorithm of CPM compared with Apriori algorithm and CONTEMUM algorithm show that the proposed CPM algorithm can obviously improve the generation efficiency of user preferences.

Key words: MaxClique; association rule; preference database; conditional preference rule; preference mining

0 引言

电子商务的飞速发展使得人们更愿意通过网络快捷地选择商品和服务。但由于 Web 数据库中通常蕴含海量数据, 增加了用户选择商品的成本。大多数推荐系统能通过偏好抽取的手段获取用户的偏好要求, 并以经济的方式获取商品的信息。因此研究用户条件偏好的挖掘具有十分重要的意义。

目前已存在的一些电子商务推荐系统在实际运用中还存在着问题, 推荐效率较低, 有些还不能满足用户的个性化需求。优化具有条件偏好(即本文所指的上下文偏好)的查询功能的数据库系统引起了数据库领域研究者极大的兴趣。无论是电子商务还是个性化推荐系统, 用户偏好在系统模型的设计中都是基本要素^[1]。消费者偏好是一种消费心理, 反映消费者对某种产品或服务的兴趣或爱好程度, 受消费者个性特征以及外界环境等主客观因素的共同影响, 把握消费者偏好的变化规律比较困难, 消消费者的偏好往往通过购买行为体现^[2], 因此偏好抽取是关键技术所在。偏好抽取的结果能使用户明白自身的偏好要求, 并能以最经济的方式获取偏好商品信息。偏好挖掘的使用方法中, 基于关联规则的技术是较

为热门的, 但在实际应用中, 关联规则推荐技术也存在着一些问题, 比如: 发现关联规则比较困难, 关联规则的产生仅与用户消费的结果有关而忽视了产生偏好的上下文约束, 而用户的偏好不稳定往往随着上下文的不同变化。本文主要阐述用偏好挖掘的算法获取用户的潜在偏好, 使用的是系统事先收集的样本, 用成对出现的数据表达用户的偏好。用户偏好由一系列条件偏好规则详细说明, 这些规则满足一定的兴趣标准, 并具有较强的预测性。本文提出基于条件的偏好规则的模型为 $i^+ > i^- | X$, 它表示在上下文 X 的约束下, 相对于 i^- , 用户更偏好 i^+ 。这里的条件约束就是指上下文约束。

例如, 电影偏好数据库中的元组信息表明了用户的偏好, 这些偏好是由用户对点击电影标记表示他们的兴趣而产生的。标记 A, B, C, D, E 分别代表电影的导演、主演、影片类型等。如: A 为张艺谋, B 为章子怡, C 为戏剧片, D 为巩俐, E 为动作片, 每个 t_i ($i = 1, 2, \dots, n$) 代表用户的事物标记 TID , 假设用户点击了 10 次 t_1 , 点击了 6 次 t_3 , 则表明他们对与 t_1 有关的电影更感兴趣, 就得到一条偏好, 用 $\langle t_1, t_3 \rangle$ 表示, 以此形成一个偏好数据库 P , 如图 1(b) 所示。

分析偏好规则 $\langle t_1, t_3 \rangle$, 发现无论在哪个元组中包含了标

收稿日期:2017-05-16;修回日期:2017-06-07。基金项目:国家自然科学基金资助项目(61572419, 61572418, 61403328)。

作者简介:谭征(1968—), 男, 山东乳山人, 副教授, 硕士, 主要研究方向:数据挖掘、自然语言处理; 刘惊雷(1970—), 男, 山西临猗人, 副教授, 硕士, CCF 会员, 主要研究方向:人工智能、理论计算机科学; 余航(1998—), 男, 江西上饶人, 主要研究方向:数据挖掘、随机化算法。



记 A (张艺谋)和标记 C (戏剧片),用户更喜欢带有标记 D (巩俐)而不是带有标记 B (章子怡)主演的电影。所以条件偏好规则为:在两部由张艺谋执导的戏剧片中人们更喜欢巩俐主演的电影而不是章子怡主演的。这里的导演和影片类型就构成了这条规则的上下文。本文提出的用户偏好是由一系列条件偏好规则构成的,这些规则要满足稳定性和简洁性:所谓稳定性是指规则与大量的用户偏好吻合,与少量的不一致;简洁性是指生成的偏好规则集尽可能地小。

TID	属性1	属性2	属性3	属性4	属性5	Pid	用户偏好
t_1	A		C	D		p_1	$\langle t_1, t_3 \rangle$
t_2	A	B		D		p_2	$\langle t_2, t_3 \rangle$
t_3	A	B	C		E	p_3	$\langle t_2, t_4 \rangle$
t_4			C	D		p_4	$\langle t_3, t_4 \rangle$
t_5	A	B				p_5	$\langle t_4, t_5 \rangle$

(a) 影片数据库DB

(b) 偏好数据库P

图 1 影片数据库及偏好数据库

Fig. 1 Movie database and preference database

偏好规则的挖掘通过关联规则实现。尽管关联规则是挖掘算法中的基本方法之一,但是它的改进方法有很多,本文将改进的关联规则算法 MaxClique 用于条件偏好挖掘(Conditional Preference Mining, CPM)算法中来挖掘用户偏好规则。

1 相关工作

挖掘具有条件偏好的偏好规则引起了众多专家学者的兴趣。目前一些主要的研究工作包括:开发偏好建模推理框架的研究,为个性化数据库应用提供说明性和应用性很强的偏好查询语言。然而专注于偏好规则抽取的工作不多。偏好抽取就是让用户知道在一个数据库中什么是他们的偏好并且通过较低的成本获取这些偏好。提取用户偏好代表性的研究方法可以分为以下两类:

1) 通过一个查询界面输入用户偏好。这种情况下用户偏好的表达受到了很大的限制。用户偏好只能用简单模糊词标记元组或属性的方式表达,而且用户要参与到反馈的过程中,如果用户真的能够准确地表达自己的偏好,也就不需要偏好挖掘了,所以通过这种方式获取用户的偏好很困难^[3]。

2) 用挖掘算法获取用户潜在的偏好。一般情况下,偏好受各种因素的影响,用户无法准确地表达自己的偏好^[3]。随着数据挖掘关联规则技术的日益成熟,利用关联规则从用户的历史记录中挖掘隐式用户偏好,抽取一系列具有简洁性和稳定性的偏好规则成为目前的研究热点。文献[4]使用了用户喜欢与不喜欢这样的语义表达帮助个性化查询,对上文考虑不多。文献[5]是利用贝叶斯网络(Bayesian Network, BN)结合上下文挖掘偏好,计算量较为复杂。文献[6]将偏好公式嵌入到关系代数 SQL(Structured Query Language)中,其优点是查询速度快。文献[7]中提到的条件偏好查询使用规则集挖掘偏好,并对规则集的 top- k 条规则进行了评估。文献[8]提出了基于上下文的偏好挖掘,针对不同数据源产生的偏好规则可能出现重复和矛盾的现象,建立离线先验组存放具有代表性的偏好规则。文献[9]介绍了集格理论可用于搜索空间的压缩,对本文的研究有一定的启示作用。文献[10]通过分析消费者的浏览频率、浏览时间、连接路径以及路径深度作为用户偏好商品的权重,并结合双向关联规则理论挖掘具有相互依赖关系的商品,计算消费者对商品的偏好程度。

但是没有考虑上下文对消费者偏好的影响。文献[11]给出了基于上下文的数据库查询规则排序算法,提出了上下文偏好模型。文献[12]给出了一种利用普适计算技术进行上下文感知的数据库查询系统,但是需要偏好规则样本集。文献[13]则将上下文感知用于移动用户的偏好挖掘中,讨论了上下文独立和非独立条件下的两种偏好挖掘方法。文献[14]提出按属性排序挖掘偏好的方法。文献[15]是在电影数据集中设计的一个推荐系统,利用的是电影导演和演员提供的信息而不是用户的评级,对于用户偏好的挖掘显然有误差。

偏好规则一般是通过关联规则或改进的关联规则挖掘出来的。类关联规则算法(Class Association Rule, CAR)^[16]是一种专注于挖掘特殊子集的关联规则算法,目标是在数据库中找出一组规则进行精确分类。基于划分的算法^[17]将数据划分成内存能处理的块,第一次扫描数据库产生局部频繁集,第二次扫描数据库形成全局频繁项集,缺点是会产生部分假频繁项集。文献[18]提出了一种基于抽样的关联规则抽取算法。文献[19]对基于抽样的关联规则算法的有效性进行了分析。CMAR(Classification based on Multiple Class-Association Rules)算法^[20]是一种基于多关联规则的分类算法,该方法拓展了频繁模式挖掘方法,有效地挖掘大型数据库。

目前尽管有大量的研究工作致力于用户偏好的表达和处理,但是这些用户偏好很难应用到实际中,因为在查询结果和算法执行效率上每种方法都存在自己的不足。本文在挖掘算法中采用生成最大团的图模型算法,旨在减少系统内存的占用,压缩数据集的扫描次数。在此基础上结合上下文(条件),提出了 CPM 算法用于挖掘偏好规则,从而能得出用户的偏好。这一方法提高了偏好规则的产生效率和偏好的准确性。

2 上下文偏好

2.1 上下文偏好规则定义

设 I 为项目集合, X 是 I 的子集, 即 $X \subseteq I$, 事务数据库 D 是 I 的多项集, 每个项集是数据库的一个元组, 如图 1(a) 中的一行。偏好数据库 $P \subseteq D \times D$ 是成对事务, 每条记录代表用户的一个偏好, 如: $\langle t, u \rangle \in P$, 意味着, 与 u 相比用户更偏好 t 。偏好数据库和事务数据库的联系如图 2 所示。

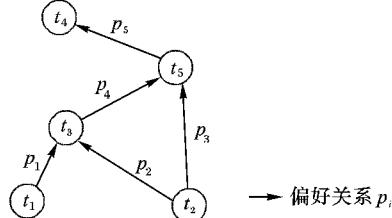


图 2 偏好数据库与事务数据库联系

Fig. 2 Relation of preference database and transaction database

定义 1 上下文偏好。 $i^+ > i^- | X, X \subseteq I, i^-$ 和 i^+ 是 $I - X$ 中的项集, 称为上下文偏好规则。如: $D > E | AB$ 表示在出现上下文 AB 时, 相对于 E , 用户更加偏好 D 。偏好规则 $R = i^+ > i^- | X$ 所关联的偏好数据库元组 $t >_R u$, 由如下方式来构造:

$$(X \cup \{i^+\} \subseteq t) \wedge (X \cup \{i^-\} \subseteq u) \wedge (i^- \notin t) \wedge (i^+ \notin u)$$

例如偏好规则 $D > E | A$ 所关联到的一个元组为: $ACD > ABCE$, 它表示当出现 A 时, 用户对 CD 的偏好优于对 BCE 。



定义 2 偏好规则的覆盖集。设 $agree(R, P) = \{ \langle t, u \rangle \in P \mid t >_R u \}$, 表示 R 满足 P 中的成对集合 $\langle t, u \rangle$, 即用户相对 u 更偏好 t 。设 $contradition(R, P) = \{ \langle t, u \rangle \in P \mid u >_R t \}$ 表示偏好规则 R , 满足成对集合 $\langle t, u \rangle \in P$, 表明用户相对于 t 更偏好 u , 则覆盖集记作:

$$cov(R, P) = agree(R, P) \cup contradiction(R, P)$$

2.2 上下文规则的支持度、可信度和最小偏好规则

定义 3 偏好规则的支持度。一条上下文偏好规则 R 在偏好数据库 P 中的支持度描述如下:

$$sup(R, P) = | agree(R, P) | / | P |$$

如: 上下文关联规则 $D > E \sqcap A$ 与图 1(b) 中的 p_1, p_2 相匹配, $D > E \sqcap B$ 与 p_2 相匹配, 前者的支持度为 $sup(D > E \sqcap A, P) = |\{p_1, p_2\}| / |P| = 2/5 = 0.4$; 而后者的支持度 $sup(D > E \sqcap B, P) = |\{p_2\}| / |P| = 1/5 = 0.2$ 。支持度反映了上下文关联规则 R 对偏好数据库中元组的匹配程度, 同时也反映了一条上下文偏好规则的兴趣度。显然, 规则 $D > E \sqcap A$ 比规则 $D > E \sqcap B$ 更有趣。

定义 4 可信度。一条上下文偏好规则 R 对应于偏好数据库 P 的可信度描述如下:

$$conf(R, P) = | agree(R, P) | / | cov(R, P) |$$

如: $conf(D > E \sqcap A, P) = 2/2 = 1$; $conf(D > E \sqcap \emptyset, P) = 2/3$ 。

可信度用来衡量一个上下文偏好规则与用户偏好相悖的程度, 就这点而言, $D > E \sqcap A$ 比 $D > E \sqcap \emptyset$ 更有价值。如果一条上下文偏好规则 R 的支持度超过某一指定值即阈值(用 min_sup 表示) 称为频繁关联规则。如果一条频繁关联规则的可信度超过某一指定值即阈值(min_conf), 称为强关联规则。可利用支持度阈值和可信度阈值进行剪枝即去掉非频繁项。图 1 中 $D > E \sqcap B$ 和 $D > E \sqcap AB$ 具有相同的支持度和可信度, 希望保留上下文更短的规则, 即最小的上下文规则。本文将采用可扩展的关联规则方法进行频繁规则的挖掘, 第 3 章将描述这一问题。

定义 5 最小偏好规则。与偏好数据库 P 关联的上下文偏好规则 $R: i^+ > i^- \sqcap X$, 如果不存在另一条规则 $R': i^+ > i^- \sqcap Y$, ($Y \subset X$) $\wedge (sup(i^+ > i^- \sqcap Y, P) = sup(i^+ > i^- \sqcap X, P)) \wedge (conf(i^+ > i^- \sqcap Y, P) = conf(i^+ > i^- \sqcap X, P))$, 称 R 为最小上下文偏好规则。利用最小上下文规则可以大幅减少上下文偏好规则的数量, 如 $D > E \sqcap B$ 是最小的上下文偏好规则而 $D > E \sqcap AB$ 不是, 即可剪去。在给定的偏好数据库中, 指定支持度和可信度的阈值, 抽取出超过支持度和可信度阈值的最小的上下文偏好规则, 是本文要解决的主要问题。一组偏好规则的简洁性由其基数的大小衡量, 稳定性则由代价函数来考量。给定一条偏好规则 π , 偏好数据库 p , 上下文偏好规则集 Π , 用户偏好和 Π 匹配可表示为: $agree(\Pi, p) = \bigcup_{\pi \in \Pi} agree(\pi, p)$, 用户偏好和 Π 不一致的可表示为 $contradict(\Pi, p) = \bigcup_{\pi \in \Pi} contradict(\pi, p)$ 。被 Π 覆盖的用户偏好 $cover(\Pi, p) = agree(\Pi, p) \cup contradict(\Pi, p)$ 。利用覆盖关系可以定义用户偏好代价函数。

定义 6 代价函数。给定偏好数据集 p , 上下文偏好规则集 Π , 与 p 关联的 Π 代价函数即

$$Cost(\Pi, p) = (|p| \cdot cover(\Pi, p)) +$$

$$|contradict(\Pi, p)| / |p|$$

直观来看, 用户偏好集 Π 的代价函数表示偏好数据库 p 中的偏好没有被 Π 中的任何一条规则覆盖或者与 Π 中的部分规则相反的偏好在 p 中所占的百分比。按照这一定义, 用户偏好的稳定性是指 Cost 函数的值最小。若某一偏好数据库上 p 的偏好规则为 R , $\Pi \subseteq R$ 满足简洁性和代价最小, 称 Π 是与 p 关联的用户偏好。本文第 4 章会有相应的算法。

3 可扩展关联规则算法

可扩展的关联规则算法是一种发现频繁项集的高效算法, 主要是利用频繁项集的结构性质实现快速查找。采用的方法是将项目集分解成小的独立的子集格从而组成一个子集格搜索空间, 可在内存空间中完成搜索, 可高效地确定长频繁项集及其子集, 即可完成一般规模的数据集处理, 对大数据集也有效, 因此称为可扩展的关联规则算法。它是对传统关联规则作拓展与改进。本文所用的最大团方法就是这种方法。

3.1 集格理论

关于一些集格理论的术语, 文献[8]中有很好的叙述。现在回顾一下主要的内容。

定义 7 集格。 L 在有限的交、并集合的操作中是闭合的, 称之为集格。 (L, \subseteq) 是由子集关系 \subseteq 定义的偏序集格。 $X \vee Y = X \cup Y \sqcap X \wedge Y = X \cup Y$, $A(L)$ 表示 L 原子集。

引理 1 所有频繁集的子集都是频繁的, 非频繁项的超集都是非频繁的。在自底向上的搜索频繁项集的过程中, 这是一个有力的剪枝策略。

引理 2 最大频繁项集唯一决定了所有频繁项集。

引理 3 对于一个有限的布尔集格 L , 若 $x \in L$, $x = \bigvee \{y \in A(L) \mid y \leqslant x\}$, 元素是以原子集的子集的并的形式给出的。

引理 4 对于任意存在的元素 $X \in P(L)$, 令 $J = \{Y \in A(P(L)) \mid Y \leqslant X\}$, $X = \bigcup_{Y \in J} Y$, 且 $\sigma(x)(x$ 的支持度) $= | \bigcap_{Y \in J} L(Y) |$ 。这一引理表明, 若一个项目集以一系列项目交集的形式给出, 该项目集的支持度可由相交项目的 tid_list 的交集获得。可以通过简单地交叉任何两个 $(k-1)$ 长度的子集的 tid_list 来确定任意 k 项集的支持度。

3.2 基于前序的类

定义 8 等价关系。令 P 是一个集合, 一种在 P 上的等价关系是一种二元关系 \equiv , 若 $X, Y, Z \in P$, 这种等价关系满足:

1) 自反性。 $X \equiv X$ 。

2) 对称性。若 $X \equiv Y$, 则 $Y \equiv X$ 。

3) 传递性。若 $X \equiv Y, Y \equiv Z$, 则 $X \equiv Z$ 。

这种等价关系将集合 P 分成不相交的子集, 称作等价类。一个等价类的元素 $X \in P$ 以 $[X] = \{Y \in P \mid X \equiv Y\}$ 的形式给出。定义一个等价前缀函数:

$$p: P(L) \times N \rightarrow P(L)$$

函数 $p(X, K) = X[1:K]$, K 是 X 的前缀长度, 集格 $P(L)$ 上的等价关系 θ_k 定义如下:

$$\forall X, Y \in P(L), X \equiv \theta_k Y \Leftrightarrow p(X, k) = p(Y, k)$$

两个项集若享有一个公共的长度为 k 的前缀, 它们在一个类中, θ_k 称为基于前缀的等价关系。



引理5 所有 θ_k 包含的等价类 $[X]_{\theta_k}$ 是 $P(L)$ 的子集格。

任意的 $[X]_{\theta_k}$ 是一个含有它本身原子的一个布尔集格。设 $[A]_{\theta_k}$ 的原子集是 $\{AC, AD, AT, AW\}$, 通过引理3和引理4的应用可求出所有类的项目集的支持度。应当特别注意的是必须自底向上处理等价类, 按字典顺序的逆序, 即处理 $[W]$ 之后处理 $[T]$, 然后是 $[D]$, $[C]$, $[A]$, 这样才能保证在剪枝的过程中所有子集的信息都是可靠的。应用这一方法, 可以递归地分解不能一次纳入内存的等价类, 使其能被独立地解决且可以在反字典的顺序中进行剪枝。

3.3 最大团方法生成更小的类

定义9 伪等价关系。令 P 为一个集合, \equiv 表示 P 上的一种二元关系, 称为伪等价关系, 对所有 $X, Y \in P$, 满足:

- 1) 自反性。 $X \equiv X$ 。
- 2) 对称性。若 $X \equiv Y$, 则 $Y \equiv X$ 。

这种伪等价关系将 P 分成可能重叠的子集, 称为伪等价类。

定义10 最大团。如果一张图的所有顶点之间均有连线, 称其为完全图。完全图的子图叫作团。令 F_K 表示频繁 K 项集的集合, 定义一张 K 关联图, 以 $G_K = (V, E)$ 的形式给出, 顶点集 $V = \{X \mid X \in F_1\}$, 边集 $E = \{(X, Y) \mid X, Y \in V \text{ 且 } \exists Z \in F_{(k+1)}, X, Y \in Z\}$, 令 M_K 表示 G_K 的最大团的集合, 图3展示了 F_2 所示的关联图 G_1 , 若频繁二项集为: $\{12, 13, 14, 15, 16, 17, 18, 23, 25, 27, 28, 34, 35, 36, 45, 46, 56, 58, 68, 78\}$, 其中最大团 $M_1 = \{1235, 1258, 1278, 13456, 1568\}$ 。

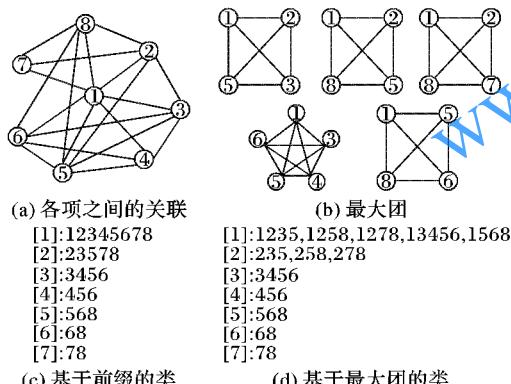


Fig. 3 Prefix classes and MaxClique classes

定义一种基于幂集格 $P(L)$ 的伪等价关系 φ_k 。 $\forall X, Y \in P(L), X \equiv \varphi_k Y \Leftrightarrow \exists C \in M_k$ 同时有 $X, Y \in C$ 且 $p(X, k) = p(Y, k)$ 。这说明 X, Y 是相关的, 它们同为一个最大团的子集, 并且共享长度为 k 的公共前缀, φ_k 为基于最大团的伪等价关系。

引理6 由伪等价关系 φ_k 生成的伪等价类 $[X]_{\varphi_k}$ 是幂集 $P(L)$ 的子集格。

引理7 令 N_k 表示基于最大团关系 φ_k 生成的伪等价类的集合, 由 φ_k 生成的伪等价类 $[Y]_{\varphi_k}$ 是某些由前缀关系 θ_k 生成的伪等价类 $[Y]_{\theta_k}$ 的子集。从另一方面来说, $[X]_{\theta_k}$ 是一组伪等价类 ψ 的并集, 记作 $[X]_{\theta_k} = \bigcup \{[Z]_{\varphi_k} \mid Z \in \psi \subseteq N_k\}$ 。

用 φ_k 来取代 θ_k 可以生成更小的集格。这些小集格对内存的需求更少。从图3中可以看出基于前缀的类 $[1] = \{12345678\}$, 包含了所有的项; 而基于最大团的类 $[1] = \{1235, 1258, 1278, 13456, 1568\}$ 比基于前缀的类小。最大团生成算法对稀疏的 K 阶关联图会产生效果, 对稠密图则不可行。

定义11 覆盖集合。对于类 $[x]$, $y \in [x]$, 称 y 覆盖了 $[x]$ 的子集, 记为 $cov(y) = [y] \cap [x]$ 。对于每个类 C , 首先确定它的覆盖集合: $\{y \in C \mid cov(y) \neq \emptyset \text{ 且对于任意 } z \in C, z < y, cov(y) \not\subset cov(z)\}$ 。计算关联图中顶点2的覆盖集的 $cov(2) = \{3, 5, 7\} = [2]$ 。对图中的例子而言 $cov(y) = [y]$, 因为所有的 y 都包含在类 [1] 中, 类 [1] 的覆盖集为 $[[2], [3], [5]]$, 4 不在集合中, 因为 $cov(4) = [5, 6]$, 它是 $cov(3) = [4, 5, 6]$ 的子集。在最大团的生成算法中, 对当前类只需要考虑它的覆盖集中的元素(算法步骤3)。对每一类覆盖集中的元素递归地生成最大团。每个从覆盖集中获得最大团都用从当前类获得的最大团的类标号作前缀。在插入新的团之前所有的副本或子集都会被淘汰。如果新团是已有团的子集, 新团就不会被插入。

算法1 覆盖集生成算法 Gencov。

输入 电影数据库;

输出 所有类的覆盖集。

```

1) for all tag[i] ∈ AllTag do
2)   for all tag[j] ∈ AllTag do
3)     if (tag[i] ≠ tag[j] ∧ IsFrequent(tag[i], tag[j])) 
4)       insert(ver[i].CoveringSet, j);
5)   end
6) end

```

算法1中将不同的电影标签作为一类, 存储在向量 ver 中, 步骤1)和步骤2)针对所有的电影标签进行频繁集的搜索, 步骤4)生成所有类的覆盖集。

算法2 最大团生成算法 Genmax。

输入 电影数据集;

输出 每一类的最大团。

```

1) Gencov(ver);
2) foreach i; N → 1 do
3)   ver[i].CliqList = ∅;
4)   for all x ∈ ver[i].CoveringSet do
5)     for all c ∈ ver[x].CliqList do
6)       M = c ∩ ver[i].CoveringSet;
7)       if (!M.empty)
8)         M = M ∪ [i];
9)       if (Not Exist X ∥ Y ∈ ver[i].CliqList, X ⊆ Y ∥ Y ⊆ X)
10)        insert(M, ver[i].CliqList);
11)      endif
12)    endfor
13)  endfor
14) endfor

```

算法中的 ver 是一个向量, 它的最大容量为 N 即电影标签的总数量, 步骤1)中调用算法1, 产生所有的覆盖集, 步骤5)~8)描述了最大团的生成过程。

算法3 所有团的生成算法 Genall。

输入 每一类的最大团;

输出 所有的团。

```

1) allCliq = ∅;
2) foreach i; 1 → N do
3)   for allcliq ∈ ver[i].CliqList do
4)     for allsubCliq of cliq do
5)       if (Not Exist Cliq ∈ allCliq == subCliq)
6)         insert(subCliq, allCliq);
7)       endif
8)     endfor

```



```

9)    endfor
10)   endfor

```

算法中 $allCliq$ 是一个用于存储所有团的集合, 步骤 2) ~ 6) 描述了利用所有类的最大团分解为不同的子团的过程, 值得注意的是, 所有的这些子团即为所有团。

4 挖掘偏好规则算法

4.1 偏好规则挖掘算法

为了保证规则的简洁性, 通过减少深度优先搜索解空间的剪枝方法来达到这一目的。

引理 8 剪枝标准。如果一个上下文偏好规则 $i^+ > i^- \mid X$ 是非频繁且非最小化的, 则任何规则 $i^+ > i^- \mid Y$ (X 是 Y 的前缀) 都是非频繁且不是最小的上下文偏好规则。

算法 4 用户偏好规则挖掘算法 CPM。

输入 电影数据库, 偏好数据库, 最小支持度阈值 $minsuppt$, 最小可信度阈值 $minconf$;

输出 偏好规则集。

```

1) rules = ∅;
2) Genall(Genmax());
3) for all Cliq1 ∈ allCliq do
4)   for all Cliq2 ∈ allCliq do
5)     flag = 0;
6)     if (Cliq1.length == Cliq2.length)
7)       for (k = 1; k ≤ NumOfTag; k++) do
8)         if (Cliq1.tag[k] ≠ Cliq2.tag[k]) flag++;
9)       endif
10)      endfor
11)    endif
12)    if (flag == 1)
13)      suppt1 = suppt(Cliq1, Cliq2);
14)      suppt2 = suppt(Cliq2, Cliq1);
15)      if (suppt1 > minsuppt && suppt1 / (suppt1 + suppt2) >
           minconf)
16)        insert(rules, Cliq1, Cliq2);
17)      endif
18)    endif
19)  endfor
20) endfor
21) return rules

```

$NumOfallCliq$ 是所有团的数目, 标记量 $flag$ 表示两个团是否符合规则交叉的条件, 算法步骤 7) 描述了长度相等且只有一项不等。步骤 3) ~ 14), 描述了如何判断两个团符合交叉生成规则的条件的过程, 算法步骤 15) ~ 21) 则描述了对两个满足交叉条件的团进行支持度判断以及生成规则的过程。由算法得知, 根据其前缀上下文, 一个最小前缀上下文规则是最小的。前缀最小是有约束的最小。用深度优先的方法可以相对容易地获得最小前缀规则集合。

4.2 频繁项集搜索策略

1) 自底向上的搜索。

自底向上的搜索是基于等价关系 θ_k , 采用递归的策略将类分解为更小的类。等价类格可以由深度优先搜索和广度优先搜索策略生成。

假如原子集为 $\{A, C, D, T, W\}$ 由广度优先搜索生成的类为 $\{[AC], [AT], [AW]\}$, 继续往下生成的类应该是 $\{[ACT], [ATW], [ACW]\}$, 最后是 $[ACTW]$ 。计算任何项集的

支持度, 只需要简单地把前一阶段生成的它的两个子集做项目列表的交运算就可以了。算法输入的是集格 S 中原子项通过对不同的成对项做交集来产生频繁项, 并通过递归的方法产生出某一阶段的频繁项集, 并将上一阶段重复的项集删除。

算法 5 CONTENUM 算法。

输入 偏好数据库 P , 前缀为 X 的规则;

输出 偏好规则集 S 。

```

1) S = ∅
2) for all i ∈ C do
3)   s12 = supp(i1 > i2 | X ∪ {i}, P)
4)   s21 = supp(i2 > i1 | X ∪ {i}, P)
5)   if ((s12 ≥ σ) ∨ (s21 ≥ σ)) ∧
        ((s21 < supp(i1 > i2 | X)) ∨
         ((s12 < supp(i2 > i1 | X)) then
6)     if ((s12 ≥ σ) ∧ (s12 / (s12 + s21) ≥ κ)
        then S = S ∪ {i1 > i2 | X ∪ {i}}
7)     if ((s21 ≥ σ) ∧ (s21 / (s12 + s21) ≥ κ)
        then S = S ∪ {i2 > i1 | X ∪ {i}}
8)     S = S ∪ CONTENUM((i1, i2), X ∪ {i},
           {c ∈ C | i < ic}, P, σ, κ)
9)   end if
10)  end for
11) return S

```

2) 自顶向下的搜索。

自顶向下的搜索方法从集格的最顶层元素开始, 如果顶层元素为 K 项集, 则需要进行 K 轮的交集运算。这种策略的优势是如果最大元素相当长, 可以避免计算它的子集的支持度。搜索从顶层开始, 如果顶层项集是频繁的, 算法结束; 否则进行下一阶段 $k - 1$ 项的判断, 直到所有的最小频繁项被找到为止。算法构造的过程中可采用哈希表, 记录存储非频繁项集, 避免重复检索。自顶向下的搜索策略只需要将项目列表存入内存即可。

5 实验结果及分析

本章将用实验评价 CPM 算法在抽取上下文偏好规则中的表现。5.1 节描述了实验中所使用的真实的数据集。在此基础上描述了不同类型的实验结果。5.2 节的实验主要用于评价 CPM 算法在抽取有趣条件偏好规则中的表现。5.3 节对用最大团算法挖掘频繁集与传统的 Apriori 算法以及 CONTENUM 算法在挖掘效率上进行了对比。所有的实验程序用 C++ 完成。所有实验在计算机上运行通过, 计算机为 Windows 操作系统, 8 GB 内存, CPU 为 i7-4710hq, 4 核心 8 线程。

5.1 实验使用的数据集

在本节的中实验采用的第 1 个真实数据集来源于 www.movieLens.org。数据集由 71 567 个用户对 65 133 部电影的 10 000 054 个评分(评分的范围为 1 ~ 5 分), 每个用户至少标记了 20 部电影, 每部电影的属性描述包括类型、导演、年份、主演等。第 2 个数据集为 last.fm 数据集, 包含 1 892 个用户对 17 632 个音乐家作品的 92 834 条播放信息, 用户共对音乐家添加了 186 479 个标签(含 11 946 个不重复的标签)。

在电影实验数据集中提取了所有的用户的评分数据。用于比较 CPM 算法和基于 Apriori 的算法^[21] 以及 CONTENUM 算法对偏好规则的抽取效率。在表 1 中, 可以明显地发现数



据集的主要特征,即每个用户的 ID 以及每个用户评价的电影数目(电影数据集 D),每一部被评价的电影的不同特征(属性值集合 L)和由该数据集产生的偏好(两部电影若评分之差超过一个阈值则认为用户对这两部电影存在偏好)构成的成对的电影偏好(偏好数据集 P)。

表 1 真实世界的电影偏好数据集

Tab. 1 Real world preference dataset over movies

User ID	D	L	P	User ID	D	L	P
user16277	525	1243	20237	user6338	634	1448	37026
user3200	537	1262	42175	user34	638	1580	56843
user351	539	1289	43579	user2517	661	1855	50271
user58050	584	1308	29635	user27006	700	1878	75079
user12280	609	1538	43992	user1075	729	1757	26715

用不同规模的数据集训练 CPM 算法,从而发现算法的效率差异是极为重要的实验环节,可以通过修改评分差的阈值来简单地生成不同规模的偏好数据集。

5.2 电影数据集中的偏好规则抽取

图 4 的(a)和(b)中展示了随着支持度和可信度的逐渐增加,CPM 算法从所有用户的数据集中提取偏好规则的数目会随着支持度和可信度的增加而不断减少。当最小可信度接近 1 时,提取到的规则数目急剧下降,这是由于实验采用的数据集极为庞大,很难保证在极高的可信度下规则不被噪声数据影响。

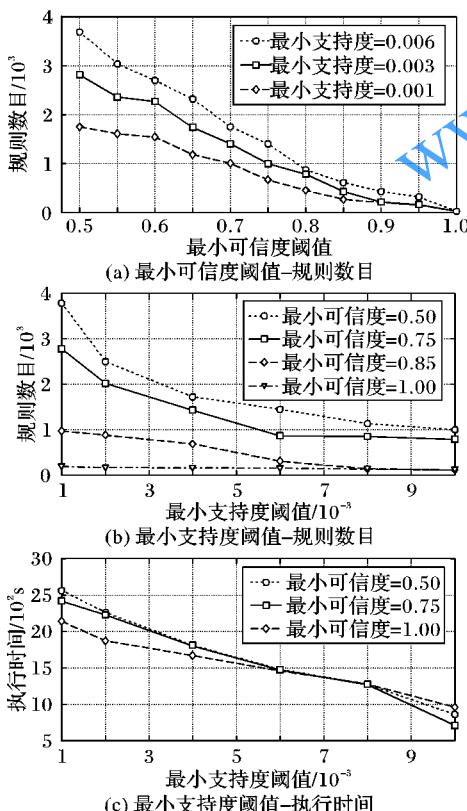


图 4 支持度和可信度变化时的规则数目和执行时间

Fig. 4 Numer of rules and execution time

according to variation of confidence and support

图 4 的(c)中描述了 CPM 算法在 0.1% 到 1% 的最小支持度阈值(对应三种不同的最小可信度)下的执行时间。从图中可以看出,最小可信度阈值并不是影响 CPM 算法执行时

间的主要因素,这是因为 CPM 算法基于最小支持度生成候选上下文前缀,通过候选上下文的交叉运算来生成规则,而当候选上下文前缀的数目确定后,算法所需执行的时间便仅与数据集的规模有关。

通过 CPM 挖掘得到的偏好规则(其中最小支持度 = 0.1%),如表 2 所示,本文中按支持度列出了前 10 条偏好规则,这些偏好规则是从 400 万条电影偏好数据库中发现的。例如,规则显示了用户普遍不太偏好动作电影(规则 1,2,3,5,9),规则 4 表明在惊险类的电影中用户普遍偏好犯罪类型的影片而不喜欢恐怖片;规则 6 则显示了用户比起纪录片更喜欢探险类的影片;规则 7 说明了同样是喜剧类电影,用户更喜欢犯罪类型电影,而不是奇幻类电影。规则 8 中显示用户更喜欢惊险类电影中的动作片而不是喜剧片,在魔幻类电影中,用户更喜欢正剧而不是喜剧。

表 2 挖掘出的偏好规则

Tab. 2 Preference rules discovered from database

上下文偏好规则	支持度/%	可信度/%
1. Mystery > Action NULL	1.20	100.00
2. Film-Noir > Action NULL	0.69	100.00
3. Crime > Action NULL	0.42	100.00
4. Crime > Horror Thriller	0.34	100.00
5. Thriller > Action Crime Drama	0.19	100.00
6. Mystery > Documentary NULL	0.15	100.00
7. Crime > Fantasy Comedy	0.14	100.00
8. Action > Comedy Thriller	0.13	100.00
9. Comedy > Action Crime	0.11	100.00
10. Drama > Horror Fantasy	0.11	100.00

图 5 采用了一个固定的支持度阈值 K (即用规则的最小出现次数代替占数据集的最小比例),展示了在不同 K 值下,从所有偏好数据集中挖掘出的上下文长度不同的规则(在不同长度上下文约束下)以及占所有规则的比例(最小可信度为 0.5)。

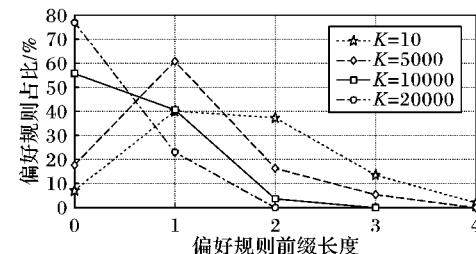


图 5 不同长度上下文的偏好规则占比

Fig. 5 Proportion of preference rules with different length of context

从图 5 的实验结果中可以看出,所有的规则基本集中在上下文长度为 0 到 4 的区间上。当 $K=10$ 时,76.6% 的规则的上下文长度为 1 和 2,上下文长度为 0 的规则仅占 7.1%;而当 $K=5000$ 时,上下文长度为 1 和 2 的规则占比为 77%,但不存在上下文长度为 4 的规则,上下文长度为 0 的规则占比提升至 17.7%。值得注意的是,随着 K 值的不断提高,长上下文规则的比例不断减小,当 $K=20000$ 时,规则的类型减少到仅有两种(上下文长度为 0 和上下文长度为 1 的规则),其中上下文长度为 0 的规则占比 76.9%。这一结果的出现是十分自然的,这是 CPM 算法挖掘规则的特性,同时说明大部分长上下文规则都是在低出现次数下产生的冗余规则,也



进一步地说明最短上下文规则是最具有简洁性和稳定性的。

5.3 相关算法的比较

5.3.1 MovieLens 数据集

图 6(a) 比较了 CPM 算法和基于 Apriori 的算法以及 CONTEMUM 算法在固定的最小支持度(0.6%)和最小可信度(0.75)下从不同规模的偏好数据库中提取规则所花费的时间。

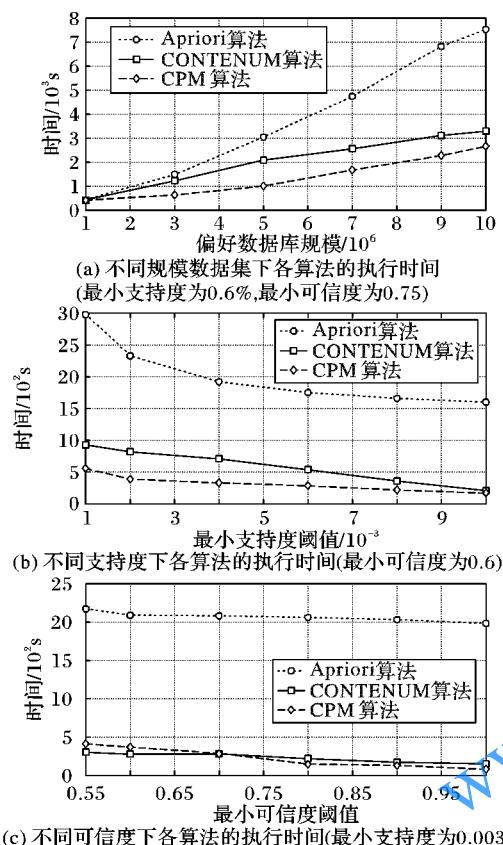


图 6 在 MovieLens 数据集下各算法的执行时间

Fig. 6 Execute time of different algorithm on MovieLens dataset

随着数据集的规模增大, 基于 Apriori 的算法所提取规则耗费的时间以指数形式增长, 这是由于基于 Apriori 的算法需要大量的数据库扫描, 同时随着数据集的规模增大, 基于 Apriori 的算法生成了更多的候选上下文前缀, 这加剧了偏好数据库的规模对算法执行时间的影响。而 CONTEMUM 算法在数据库规模增大的情况下, 时间开销呈线性增长, 数据库规模较为巨大的情况下, 算法消耗的时间甚至逼近 CPM 算法, 这是由于 CONTEMUM 算法不预先生成候选的上下文前缀, 而采用基于短上下文规则扩展的方式。

基于最大团的算法优于以上两种算法, 这是因为 CPM 算法采用基于最大团的混合搜索的方式生成上下文前缀, 这种方式不同于 Apriori 算法请求大量的数据库扫描, 只需要两次数据库扫描即可生成候选的上下文前缀, 从而快速通过上下文前缀交叉运算生成可能存在的规则。

图 6(a) 和(c) 描述了在相同的偏好数据库的规模下, 不同的支持度阈值和可信度阈值对算法运行时间的影响。基于 Apriori 的算法随着最小支持度阈值的提升执行时间明显缩短, 而 CPM 算法则几乎没有受到影响(见图(b)), 这是因为生成上下文规则前缀的过程仅占时间开销的极小的一部分, 故即便使用混合搜索进行剪枝也难以提升性能, 而基于

Apriori 的算法则受益于此, 大幅提升算法的性能(但时间开销仍为 CPM 算法的 6 倍和 CONTEMUM 算法的 4 倍以上)。CONTEMUM 算法随着支持度和可信度的提高, 算法的执行时间均有较大比例的减少, 这是因为自底向上的算法仅依赖高于支持度和可信度的规则进行递归挖掘。在最小可信度阈值高于 0.76 时, CONTEMUM 算法甚至优于 CPM 算法, 这一结果很好地吻合了当支持度和可信度高于某个阈值后, 规则大量集中在较短的上下文前缀的类型上。CONTEMUM 算法没有生成频繁项集的过程, 当规则仅有长度为 0 或长度为 1 的上下文前缀时, 它的效率非常高。

5.3.2 Last.fm 数据集

三种算法在 Last.fm 上的比较结果与其在 MovieLens 上的结论基本一致。需要注意的是, 图 7(a) 中所有的算法的执行时间都有所增加, 原因是 Last.fm 所提供的数据在一条条目下具有更多的标签, 且标签的种类更为丰富。图 7(b) 和(c) 比较了不同最小支持度阈值和不同最小可信度阈值下各算法的执行时间, 无论是 Apriori 算法还是 CONTEMUM 算法的执行时间都有较大的增加, 但 CPM 算法的执行时间变化不大。其中 Apriori 算法的时间增加得最为明显, 这是由于 Apriori 算法生成上下文前缀的时间呈指数增长。CONTEMUM 算法的时间开销的增长也值得关注, CONTEMUM 算法基于自底向上的方式生成上下文, 在这种长前缀且低重复的状况下其剪枝的效率下降极为明显。而 CPM 算法, 只匹配条件符合的团, 比在短前缀有较多重复的情况下更加高效。

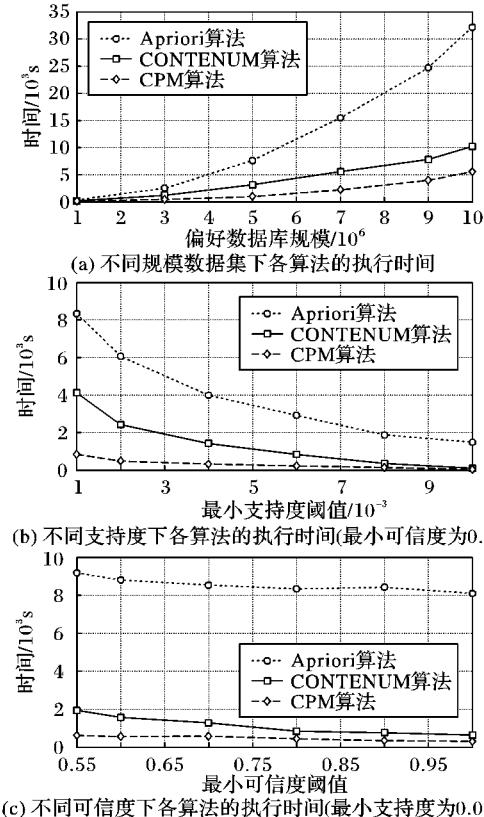


图 7 在 Last.fm 数据集下各算法的执行时间

Fig. 7 Execute time of different algorithm on Last.fm dataset

6 结语

本文提出了偏好规则挖掘算法 CPM, 用于从数据库中挖



掘用户偏好，并在真实世界电影用户偏好集上进行了一系列实验，实验证明本文算法是高效的。实验得到的条件偏好规则建立在上下文的基础上，以可读的用户偏好规则的形式呈现。

在个性化的查询领域中，偏好挖掘技术将变得越来越重要，可以把从用户偏好样本中抽取的偏好模型存入知识库中，这些模型可以满足用户通过SQL语句进行的个性化查询，现已有很多文献中都提到了用可扩展的SQL即带有用户偏好查询操作的SQL语言来进行个性化查询。

当然在电子商务普遍应用的推荐系统的发展过程中，偏好挖掘的技术也是必不可少的。实际上随着被推荐商品的数量急剧增长，传统推荐算法的思想即基于由其他用户提供的对商品的评价与用户对购买商品评价之间的相似性计算，越来越受到可扩展性的挑战。偏好技术的发展，使得以简洁偏好形式出现的用户偏好的自动推理变得更有意义，有望解决可扩展性的问题。

下一步将研究如何利用评测规则对挖掘出的偏好规则进行过滤；对用户偏好的评价要能跳出一般的布尔判断；尝试建立以偏好挖掘算法为核心的偏好挖掘框架模型。

参考文献 (References)

- [1] ZAKI M J. Scalable algorithms for association mining [J]. IEEE Transactions on Knowledge and Data Engineering, 2000, 12(3): 372–390.
- [2] 张志宏, 寇纪淞, 陈富赞, 等. 基于遗传算法的顾客购买行为特征提取[J]. 模式识别与人工智能, 2010, 23(2): 256–266. (ZHANG Z H, KOU J S, CHEN F Z, et al. Feature extraction of customer purchase behavior based on genetic algorithm[J]. Pattern Recognition and Artificial Intelligence, 2010, 23(2): 256–266.)
- [3] AMO S, DIALLO M S, DIOP C T, et al. Contextual preference mining for user profile construction [J]. Information Systems, 2015, 49(C): 182–199.
- [4] HOLLAND S, ESTER M, KIEBLING W. Preference mining: a novel approach on mining user preferences for personalized applications [C]// Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. Berlin: Springer, 2003: 204–216.
- [5] AMO S D, MARCOS L P, ALVES G, et al. Mining user contextual preferences [J]. Journal of Information and Data Management, 2013, 4(1): 37–46.
- [6] CHOMICKI J. Preference formulas in relational queries [J]. ACM Transactions on Database System, 2003, 28(4): 427–466.
- [7] PEREIRA F S F, AMO S D. Evaluation of conditional preference queries [J]. Journal of Information and Data Management, 2010, 1(3): 521–536.
- [8] AGRAWAL R, RANTZAU R, TERZI E. Context-sensitive ranking [C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM, 2006: 383–394.
- [9] DAVEY B A, PRIESTLEY H A. Introduction to Lattices and Order [M]. Cambridge: Cambridge University Press, 2002: 1277–1286.
- [10] 刘政莲, 刘同存, 肖吉军. 基于双向关联规则的网络消费者偏好挖掘研究[J]. 微电子与计算机, 2013, 3(3): 21–26. (LIU M L, LIU T C, XIAO J J, Web consumer preference mining based on bidirectional association rules[J]. Microelectronics and Comput-
- er, 2013, 3(3): 21–26.)
- [11] 孟祥福, 马宗民, 李昕, 等. 基于上下文偏好的数据库查询结果Top-K排序方法[J]. 计算机学报, 2014, 37(9): 1986–1998. (MENG X F, MA Z M, LI X, et al. A Top-K query results approach based on contextual preferences for Web database[J]. Chinese Journal of Computers, 2014, 37(9): 1986–1998.)
- [12] AMO S D, DIALLO M S, DIOP C T, et al. Mining contextual preference rules for building user profiles [C]// Proceedings of the 14th International Conference Data Warehousing and Knowledge Discovery. Berlin: Springer, 2012, 7448: 229–242.
- [13] ZHU H, CHEN E, YU K, et al. Mining personal context-aware preferences for mobile users [C]// Proceedings of the IEEE 12th International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2012: 1212–1217.
- [14] DEMBCZYNSK K, KOTLOWSK W, SLOWINSKI R. Learning of Rule Ensembles for Multiple Attribute Ranking Problems [M]. Berlin: Springer, 2011: 217–247.
- [15] CHOI S M, KO S K, HAN Y S. A movie recommendation algorithm based on genre correlations [J]. Expert Systems with Applications, 2012, 39(9): 8079–8085.
- [16] LIU B, HSU W, MAY. Integrating classification and association rule mining [EB/OL]. [2016-11-20]. http://ckcckc.myweb.hinet.net/paper/Integrating_Classification_and_Association_Rule_Mining.pdf.
- [17] SAVASERE A, OMIECINSKI E, NAVATHE S B. An efficient algorithm for mining association rules in large databases [EB/OL]. [2016-11-20]. http://omega.sp.susu.ru/books/acm_sigmod/vol1/is5/VLDB95/P432.PDF.
- [18] TOIVONEN H. Sampling large databases for association rules [C]// Proceedings of the 22nd International Conference on Very Large Data Bases. San Francisco, CA: Morgan Kaufmann Publishers, 1996: 134–145.
- [19] ZAKI M J, PARTHASARATHY S, LI W, et al. Evaluation of sampling for data mining of association rules [C]// Proceedings of the 7th International Workshop on Research Issues in Data Engineering. Washington, DC: IEEE Computer Society, 1997: 42–50.
- [20] LI W, HAN J, PEI J. CMAR: Accurate and efficient classification based on multiple class-association rules [C]// Proceedings of the 2001 IEEE International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2001: 369–376.
- [21] AGRAWAL R, MANNILA H, SRIKANT R. Fast discovery of association rules [J]. Advances in Knowledge Discovery and Data Mining, 1996, 32(3): 307–328.

This work is partially supported by the National Natural Science Foundation of China (61572419, 61572418, 61403328).

TAN Zheng, born in 1968, M. S., associate professor. His research interests include data mining, natural language processing.

LIU Jinglei, born in 1970, M. S., associate professor. His research interests include artificial intelligence, theoretical computer science.

YU Hang, born in 1998. His research interests include data mining, randomized algorithm.