doi:10.11959/j.issn.1000-436x.2017160

支持推荐非空率的关联规则推荐算法

何明[1]，刘伟世[1]，张江[2]
(1. 北京工业大学信息学部计算机学院，北京 100124；2. 国网英大国际控股集团有限公司信息化工作部，北京 100005)

摘 要：现有的关联规则推荐技术在数据提取时主要侧重于关联规则的提取效率，缺乏对冷、热门数据推荐平衡性的考虑和有效处理。为了提高个性化推荐效率和推荐质量，平衡冷门与热门数据推荐权重，对关联规则的 Apriori 算法频繁项集挖掘问题进行了重新评估和分析，定义了新的测评指标推荐非空率以及 k 前项频繁项集关联规则的概念，设计了基于 k 前项频繁项集的剪枝方法，提出了优化 Apriori 算法且适合不同测评标准值的 k 前项频繁项集挖掘算法，降低频繁项集提取的时间复杂度。理论分析比较与实验表明，k 前项剪枝方法提高了频繁项集的提取效率，拥有较高的推荐非空率、调和平均值和推荐准确率，有效地平衡了冷、热门数据的推荐权重。

关键词：关联规则；推荐系统；推荐非空率；数据挖掘
中图分类号：TP319 **文献标识码**：A

Association rules recommendation algorithm supporting recommendation nonempty

HE Ming[1], LIU Wei-shi[1], ZHANG Jiang[2]
(1. College of Computer, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;
2. State Grid YingDa International Holdings Co., Ltd., Beijing 100005,China)

Abstract: Existing association rule recommendation technologies were focus on extraction efficiency of association rule in data mining. However, it lacked consideration of recommendation balance between popular and unusual data and efficient processing. In order to improve the quality and efficiency of personalized recommendation and balance the recommendation weight of cold and hot data, the problem of mining frequent itemset based on association rule was revaluated and analyzed, a new evaluation metric called recommendation *RecNon* and a notion of *k*-pre association rule were defined, and the pruning strategy based on *k*-pre frequent itemset was designed. Moreover, an association rule mining algorithm based on the idea was proposed, which optimized the Apriori algorithm and was suitable for different evaluation criteria, reduced the time complexity of mining frequent itemset. The theoretic analysis and experiment results on the algorithm show that the method improved the efficiency of data mining and has higher *RecNon*, *F-measure* and *precision* of recommendation, and efficiently balance the recommendation weight of cold data and popular one.
Key words: association rule, recommender system, recommendation nonempty, data mining

1 引言

随着信息技术特别是互联网、物联网和云计算等技术的迅猛发展，网络空间中所蕴含的信息量呈几何式增长。面对大量丰富多彩的数据信息，人们往往很难快速、准确地获取符合个性化需求的内容，出现了“信息过载”这一问题，导致用户无所适从。在这种背景下，推荐系统（recommender system）[1]应运而生，它可以有效地过滤处理海量数据并改善信息过载问题，为用户提供个性化信

收稿日期：2017-04-10；**修回日期**：2017-07-10
基金项目：国家自然科学基金资助项目（No.91646201, No.91546111）；北京市自然科学基金资助项目（No.4153058, No.4113076）；北京市教委面上基金资助项目（No.KM201710005023）
Foundation Items: The National Natural Science Foundation of China (No.91646201, No.91546111), The Natural Science Foundation of Beijing (No.4153058, No.4113076), General Project of Beijing Municipal Education Commission (No.KM201710005023)

息推荐。推荐系统通过挖掘用户的兴趣偏好，进行自动个性化计算来帮助用户有效获取所需要的信息。

推荐系统的研究方法有很多种，如基于内容的推荐、协同过滤推荐和基于效用的推荐等。在各种推荐技术中，关联规则作为推荐系统中重要的数据挖掘技术之一[2]，关注用户的行为模式，根据计算分析找出用户物品集合中的项目关联性，建立关联规则，并基于用户的实际购买行为进行推荐。其主要思想是以关联规则为基础，通过挖掘物品间的关联关系，把已购物品作为规则头，推荐对象作为规则体，通过数据挖掘发现物品之间潜在的联系以完成关联推荐。关联分析可以发现不同物品在销售过程中的相关性，向具有相似行为爱好的用户提供可能感兴趣的项，具有行为属性自动归类，系统自动学习推荐的优点，适合当前线上电子商务平台、线下的零售平台及其他领域的市场营销。当前，针对关联规则方法的研究主要包括关联规则中频繁项集挖掘效率的研究、关联规则的存储模型研究和减少效用较低的关联规则研究，而高效的挖掘频繁项集算法是关联规则推荐研究的重点。

Agrawal 等[3]提出的 Apriori 算法是最经典的关联规则挖掘算法，该算法的主要思路是在逐层迭代过程中使用低维频繁项集生成高维频繁项集。而在实际应用中，Apriori 算法主要受限于数据计算复杂度，因此，一些研究人员提出了很多改进的算法来进一步提高 Apriori 的有效性。在国外的研究工作中，Park[4]使用散列技术优化了 Apriori 算法，该算法改进了频繁项集连接产生候选项集的过程，大大降低了 2-项集产生的复杂度。后来，Toivonen 等[5]使用采样技术优化了 Apriori 算法，从事务数据库抽取部分数据作为样本，只挖掘样本数据中的频繁项集，虽然该算法减少了系统 I/O 操作，但是数据挖掘结果不准确的可能性也比较大。国内学者对 Apriori 算法的研究也取得了较大的进展，魏玲等[6]通过 Bigtable 技术与 MapReduce 模型优化了 Apriori 算法。刘兴彬等[7]提出了一种基于 Apriori 算法自动提取协议识别特征的方法。王大玲等[8]提出了最大关联规则，虽然该方法提高了关联规则挖掘准确率、覆盖率，但是该方法未考虑频繁项集生成效率。Sandving 等[9]提出了一种基于关联规则挖掘的协同过滤推荐算法，该算法通过牺牲推荐精度与覆盖面提高了算法的顽健性。Hong 等[10]从用户历史上下文、兴趣偏好出发提取关联规则为用户提供个性化服务，在挖掘关联规则时，把相应的上下文看作频繁项来处理，但当问题量比较大时，其计算量增加比较大。文献[11]提出利用关联规则来解决推荐系统中的冷启动问题。文献[12]通过关联规则挖掘算法抽取在线产品描述的公共特征以及这些特征之间的关联关系，然后使用 KNN 协同过滤算法为具有部分特征的新软件产品推荐其缺失的特征。

虽然以上这些关联规则提取算法都各具优点，但是它们在数据处理过程中只专注于数据的提取效率，忽略了对冷、热门数据的推荐平衡性的考虑和有效处理，缺少对推荐系统个性化推荐质量评估。

基于上述问题，在已有研究的基础上，为了有效提高关联规则挖掘过程中频繁项集的产生效率与合理平衡推荐过程中冷、热门数据覆盖度，本文的主要贡献包括以下 3 个方面。

1) 研究分析了推荐非空率 *RecNon* 的概念和计算方法，作为推荐系统中一种新的效用评价指标，用来衡量推荐项目所占给定数据集的比例。

2) 基于 Apriori 算法，提出了一种支持非空率 *k*-pre 方法，通过剪枝 *k* 前项频繁项集生成候选项集，提高了频繁项集的生成效率。

3) 在亚马逊购物记录数据集上进行了实验，并与 Apriori 算法对比，*k*-pre 算法不仅提高了频繁项集提取效率，而且提高了系统推荐质量。

2　推荐非空率

对于推荐系统的评价，一般采用通用的评价指标包括覆盖率（*coverage*）、准确率（*precision*）以及调和平均值（*F-measure*）。覆盖率是在推荐的内容中用户喜欢的项占用户喜欢的所有项的百分比，准确率是在推荐的内容中用户喜欢的项占推荐的所有项的百分比。设用户喜欢的内容集合为 *UP*，系统推荐内容的集合为 *RP*，根据定义，*coverage*、*precision* 和 *F-measure* 分别由式(1)~式(3)给出。

$$coverage = \frac{|UP \cap RP|}{|UP|} \tag{1}$$

$$precision = \frac{|UP \cap RP|}{|RP|} \tag{2}$$

$$F\text{-}measure = \frac{2coverage \times precision}{coverage + precision} \tag{3}$$

coverage 和 *precision* 分别从推荐的广泛性和精确性方面对推荐方法进行衡量，而 *F-measure* 则是两者的调和平均值。由式(3)可见，忽视 *coverage* 和 *precision* 中的任意一个，都将导致 *F-measure* 降低。本文认为，*coverage*、*precision* 和 *F-measure* 还不够全面地评价一个推荐系统的质量。基于上述考虑，并借鉴文献[8]的思想，本文提出了一种新的效用评价指标：推荐非空率，简称非空率。

定义 1 推荐非空率（*RecNon*）。在用户所访问的项集中，能够给出推荐项占被访问集合项的比例。设 *UI* 为用户访问过项的集合，*RI* 为具有推荐项的集合，则 *RecNon* 可用式(4)计算。

$$RecNon=\frac{|UI\cap RI|}{|UI|} \tag{4}$$

RecNon 与 *coverage* 具有相似之处，但 *RecNon* 强调的是在一个被访问的项中是否有推荐项，而覆盖率表示的是推荐项占数据集合的比重，非空率从新的角度对推荐系统做出评价。如一个推荐系统，如果只在用户访问的几个项中给出了推荐，而在其余的许多访问项中却没有给出任何推荐，它的 *coverage* 也可以达到较大的值，但 *RecNon* 却很小，而 *RecNon* 过小的推荐系统同样不是一个高质量的系统。显然，一个推荐系统是否能够在用户访问的页面中给出推荐内容，取决于规则集中是否存在与当前用户访问模式相匹配的规则。对使用关联规则而言，*coverage* 涉及的是规则的数量，而 *RecNon* 涉及的是规则的分布。

3 *k*-pre 算法描述及分析

3.1 关联规则基本概念

关联规则可以用蕴涵表达式 $X\rightarrow Y$ 表示，其中，$X\subset I$，$Y\subset I$ 且 $X\cap Y=\varnothing$。为了评测关联规则，定义了支持度（*support*）和置信度（*confidence*）这2个参数，支持度用来衡量一个项集事务数据库中出现次数占总事务数目的比重，而置信度表示在项集 *X* 存在或发生时，项集 *Y* 一定出现的概率。支持度与置信度如式(5)和式(6)所示。

$$support(X\rightarrow Y)=\frac{\sigma(X\cup Y)}{N} \tag{5}$$

$$confidence(X\rightarrow Y)=\frac{\sigma(X\cup Y)}{X} \tag{6}$$

式(5)中，*N* 为事务总数目。在提取关联规则时，会设置最小支持度 *minSup* 与最小置信度 *minConf*，如果项集的支持度大于或等于最小支持度，那么该项集为频繁项集，当频繁项之间的置信度大于或等于最小置信度，那么该关联规则为强关联规则。而关联规则的主要任务就是从事务数据库中提取强关联规则，主要由以下2个步骤组成。

1) 根据 $support(X)\geqslant minSup$ 找到所有的频繁项集 *X*。

2) 根据 $confidence(X\rightarrow Y)\geqslant minConf$ 产生强关联规则。

步骤1)是步骤2)的基础，影响关联规则提取效率问题主要发生在步骤1)，故大多数研究都集中于提取频繁项集，如经典的 Apriori，这也是本文的研究重点。

3.2 Apriori 算法描述

该算法是寻找频繁项集的经典算法，基本原理在于逐层搜索，迭代地用 *n*-项集去探索(*n*+1)-项集，即使用频繁项集先验知识，找出频繁 1-项集的集合 L_1，再基于 L_1 找频繁 2-项集的集合 L_2，直到不能找到满足条件的更大宽度频繁项集为止。为了清晰地描述频繁项集的提取流程，分为2个步骤。

1) 频繁项连接

为了找出频繁 *n*-项集，需要先组合频繁(*n*−1)-项集得到候选 *n*-项集，为了避免(*n*−1)-项集中的项连接时生成重复项集，必须保证(*n*−1)-项集至少拥有 *n*−2 个相同项。

2) 候选项集剪枝

由 Apriori 原理知，如果某个项集是非频繁的，那么它所有的超集也是非频繁的，因此，在迭代产生候选项集过程中，扫描事务数据库就可以判定候选项集是否为频繁的，不满足最小支持度 *minSup* 剪枝掉即可，避免了项集数据量的指数增长，能够在合理的时间内找出所有频繁项集。

该算法虽然通过剪枝候选项集提高了数据挖掘效率，但依然存在以下问题。

1) 多次扫描数据库。在迭代生成频繁项集的过程中，每次计算项集的支持度都需要扫描一遍数据库中的所有记录，计算机系统 I/O 开销比较大。

2) 产生大量候选项。当数据库中的项数非常多时，在每次生成候选项集时都会产生过多组合，候选项集数量非常庞大。

综上，Apriori 算法存在多次扫描数据库、产生大量的候选项集、系统 I/O 开销较大等问题，严重影响了关联规则的提取效率。对此，本文提出以下理论分析：在数据处理过程中，可以通过调节支持度的参数值来控制剪枝力度，但是当支持度设置较大时，虽然提高了频繁项集生成效率，但是通过数据分析可知，最终得出频繁项集主要趋向于热度较高的项，而热度较低的项在频繁项集提取过程中被剪枝，最终得到的强关联规则项集非常集中，而整体推荐效果较差；而当支持度较小时，由于 Apriori 算法扫描数据库次数过多，时间复杂度与空间复杂度将非常高，因此，最终得到热度较小项的强关联规则所占比重也非常小，*RecNon* 较小，从而推荐系统的推荐质量较低。

3.3 *k*-pre 前项关联规则

定理 1　如果一个项集的支持度不大于 *minSup*，该项集所有的超集 *T* 的支持度不大于 *minSup*。

证明　由式(5)可知，支持度的大小为包含某一项的事务个数与总事务数量的比值，由于包含某一项超集的数量不大于包含该项子集的数量，因此，包含某一项超集的支持度不大于子集的支持度。

定义 2　*k*-pre 前项关联规则。对于前项相同的规则，其中，最大 *k* 个支持度的频繁项集称为 *k* 前项关联规则，记作 *k*-pre。

定理 2　在频繁项集的处理过程中，支持度越小，频繁项集越分散，所得推荐结果对应的 *RecNon* 越大。

证明　根据式(4)，令 *I* 是项的集合，$J=2^I$ 是 *I* 的幂集。*RecNon* 是单调递增的，即 $\forall X,Y\in J$: $(X\subseteq Y)\rightarrow RecNon(X)\leqslant RecNon(Y)$。另外，在关联规则提取频繁项集时，根据式(5)，如果最小支持度 $minSupX\leqslant minSupY$，则频繁项数量 *freitem* 是反单调的，即 $|freitem(minSupX)|\geqslant|freitem(minSupY)|$，则推荐项集 $|RI(minSupX)|\geqslant|RI(minSupY)|$，推荐项集所占数据库项集的比例越高，$RecNon(minSupX)\geqslant RecNon(minSupY)$。

Apriori 算法迭代生成候选项集的过程中，当某一项集支持度较高时，该项集和其他项集结合的概率非常高，而以此项集作为前项的候选项集个数较多，在实际应用中，项集的推荐一般为几个到几十个，数据库项较大时，某一项集与其他项集结合的候选项集个数可达几百上千，而每个在判断是否为频繁项集时需要每个区扫描数据库，在数据集较大时，非常耗时。

因此，本文提出 *k*-pre 算法，其基本思想是：在频繁项集连接生成候选项集时根据频繁项的支持度对 L_{n-1}(m 个子项)排序，然后根据设定阈值由前向后支持度较大的与次之的结合，结合到第 *k* 个时结束本次循环进行下次迭代，因为（$n-1$）频繁的支持度已经计算过也排序过，在生成候选项集时不需要判断每个候选项的支持度，而是直接将后续支持度较低的候选组合项省略掉。从 Apriori 原理可知，剪枝掉的也是可信度相对较小的项，对实验结果的影响可以忽略。

k-pre 算法的具体案例分析如下：设关联规则事务数据库如表 1 所示，数据库中部分项集支持度如表 2 所示。设 *k* 前项关联规则 *k* 值为 2，最小支持度 *minSup* 为 0.25，在候选项集生成过程中 A、B、C、D 为 1-项集的频繁项。

表 1　事务数据库

事务编号（tid）	项集（items）
101	ABC
102	AB
103	AC
104	AD

表 2　部分频繁项集支持度

项（item）	支持度（sup）
A	1
B	0.5
C	0.5
D	0.25

在组合生成 2-items 候选项时，项 AB、AC、AD 的支持度都大于或等于 *minSup*，但是根据 *k* 前项关联规则，它们都拥有相同的前项 A，由于生成候选项时需要遵从前 *k* 项结合规则，$sup(B)\geqslant sup(C)\geqslant sup(D)$，只需将 A 与 B、C 结合生成候选项集，AD 为 *k* 前项关联规则剪枝掉的项集，如图 1 中深色所示；由表 2 知 BD、CD 为支持度较小剪枝掉的项集，如图 1 中浅色所示，*k*-pre 频繁 2-项集剪枝如图 1 所示。

k-pre 挖掘算法如下。

输入　数据库集合 *Data*，最小支持度 *minSup*，最小置信度 *minConf*，前项规则 *k* 值

输出　频繁项集数组 *LD*，强关联规则集合 *LG*

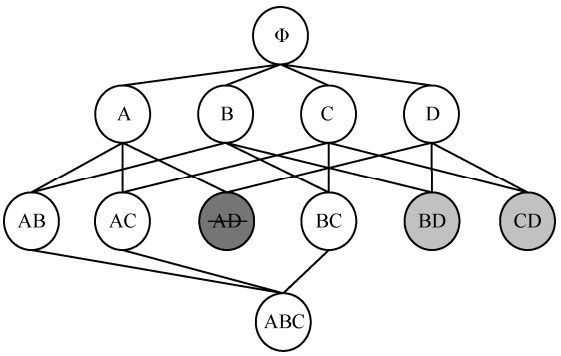


图 1　k=2，$minSup$=0.25 时 k-pre 数据剪枝流程

step1　扫描数据库集合 $Data$，得到 1-项频繁项集字典集合 $LD_{[1]}$。

首先扫描数据库得到 1-项频繁项集合 C_1，扫描并计算候选项的支持度，如果该项支持度大于 $minSup$，则加入频繁项集合，得到 1-项频繁项集字典集合 $LD_{[1]}$，作为后续步骤的操作基础。

step2　将频繁项字典集 $LD_{[n]}$（n=1, 2, …）按照支持度从大到小排序。

在 $LD_{[n]}$自连接生成候选项集时，在 n−1 步已经得到(n−1)-频繁项集与每个频繁项所对应的支持度，将前项频繁项集按照支持度从大到小排序，返回排序后的频繁项集合 $LD'_{[n]}$。

step3　根据 $LD'_{[n]}$生成前 k 项关联规则候选项集合（伪代码如下）。

```
begin
    LEN=len (LD'[n]);
    retList =[ ];
    for i=1 to len
    {for j=i+1 to len
        if j−i>k //判断是否超出前项规则，降低时间复杂度
            break;
            else
                CONNECT LD'[n][i] with LD'[n][j]
                //结合项得到新项集
                retList.append(item);
                //将结果添加到结果集中
    }
    return retList;
end
```

step4　重复 step2 和 step3 得到所有的 k 前项关联规则频繁项集追加到 LD 中。

遍历频繁项集 $LD_{[i]}$，如果 $LD_{[i]}$不为空，调用 step2 和 step3 生成对应的 i+1 长度的 k 前项关联规则频繁项集，将 step3 返回的结果更新到 LD 中，作为 i+1 项，直到 $LD_{[i]}=\varnothing$ 结束。

step5　根据频繁项集 LD 得到强关联规则。

根据式(6)计算频繁项集之间的依赖关系，得到关联项置信度 $Conf$ 大小，由置信度判断是否为强关联规则，得到满足最小置信度的推荐结果 LG，返回频繁项集数组 LD 与强关联规则 LG。

3.4　算法分析与评价

基础的 Apriori 在生成第 n 个大小的候选项集时，L_{n-1} 的频繁项的个数为 m 个，生成候选项集的个数为 $\frac{mm!}{(m-n)!n!}$，n<m，在判断候选项集支持度时，需要遍历数据库项集次数为 $\frac{m(m-1)}{2}$；k-pre 算法通过设定 k 值大小，生成候选项集的个数为 $mk(k<\frac{m!}{(m-n)!n!})$，因为 k 的值大小一般几十到上百，而数据量庞大时，$\frac{m!}{(m-n)!n!}$ 的值较大。因此，k-pre 算法时间复杂度小于 Apriori。

由于 k-pre 算法对候选项集的剪枝，降低了数据挖掘的时间复杂度，为了得到相同数据量，降低支持度的大小，提高了 $RecNon$ 和相对冷门数据的推荐权重。

4　实验及分析

4.1　实验设置

本文的实验环境为一台 PC 机，机器配置为 Intel core 2 duo，2.66 GHz，4 GB 内存，操作系统为 Windows 7。本文选择 UCI 网站的亚马逊购物记录数据作为实验的测试数据集。其中，包含了本文所需要的用户购买记录及商品信息条目。实验通过 k 值变长，并设定支持度阈值来分析算法时间复杂度、$RecNon$、$coverage$ 和 $precision$。该数据集具有 88 162 条购买记录（事务），16 470 个项（商品）。处理后，将数据集的 $\frac{2}{3}$ 作为训练集，进行商品挖掘以生成推荐的关联规则，将其余的 $\frac{1}{3}$ 作为测试集进行测试，在实验过程中对算法时间复杂度、数据挖掘效率、推荐测度等方面做了对比。

4.2　评价标准

实验使用非空率（式(4)）、覆盖率（式(1)）、准确率（式(2)）、调和平均值（式(3)）作为评估推荐质量的评价指标。另外，实验选择 Apriori 算法作为对比算法与本文 *k*-pre 算法在频繁项集生成效率（式(7)）进行比较与分析。

4.3　实验结果及分析

实验 1　该组实验主要针对数据集 *k* 前项关联规则提取频繁项集效率，比较频繁项集生成效率受支持度和 *k* 值的影响。图 2 为项集支持度在 0.002 0~0.010 0 时，原始 Apriori 算法（AP）与 *k*-pre 剪枝（*k*=25，*k*=50）（AP_*k*25，AP_*k*50）的时间比较。

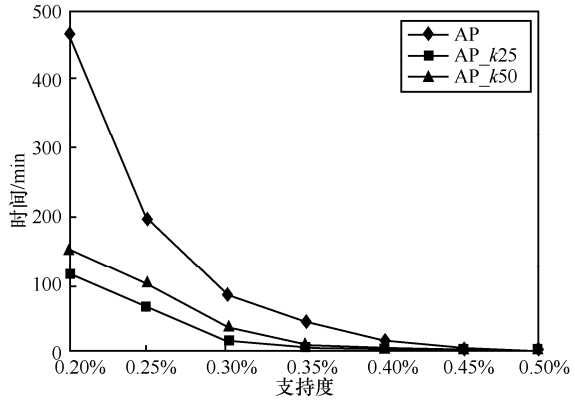


图 2　频繁项集生成时间随支持度阈值的变化情况

从图 2 可以清楚地看到满足最小支持度频繁项的提取时间随着支持度的减小数量增长非常快。当支持度为 0.20%时，*k*-pre 要比原始算法时间缩短了一半以上；当支持度大于 0.50%，由于频繁项数量少，*k*-pre 算法对候选项集的过滤度降低，*k*-pre 算法的数据挖掘提取时间接近 Apriori 算法。*k*-pre 方法在支持度相同的情况下，*k* 值较大时运行时间要比 *k* 值较小时的运行时间多，即 *k* 值越小的剪枝力度越大。通过实验 1，在相同支持度条件下，可以验证 *k*-pre 算法在相同支持度的条件下频繁项集的生成效率要高于 Apriori 算法的生成效率，但是分析中可以发现，*k*-pre 算法在剪枝了一部分数据的情况下提高了频繁项集提取效率，因此，该实验还不能完全证明 *k*-pre 算法的数据挖掘效率高于原始 Apriori 算法。

实验 2　为了验证 *k*-pre 算法绝对提高频繁项集提取效率，该组实验主要分析了频繁项集数量受 *support*、*k* 值的影响。

如图 3 所示，相同支持度下，Apriori 的频繁项集数量高于 *k*-pre 算法的数量，频繁项集的数量随着支持度的减小呈线性增加。通过图 2 与图 3 对比，可以看出数据的剪枝幅度相对于时间的变化幅度较小，即单位时间内生成频繁项集的效率得到提升。

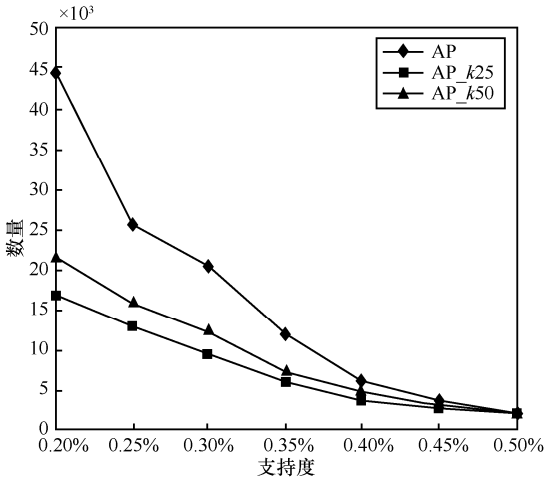


图 3　频繁项集数量随支持度阈值的变化

为了准确描述上述结果，本实验根据式(7)对数据做了进一步处理，不同的支持度、*k* 值下的单位时间内频繁项集提取数量与消耗时间比值得到 2 种算法的频繁项集生成效率（*Efficiency*），如图 4 所示。

$$Efficiency = \frac{Num}{T} \tag{7}$$

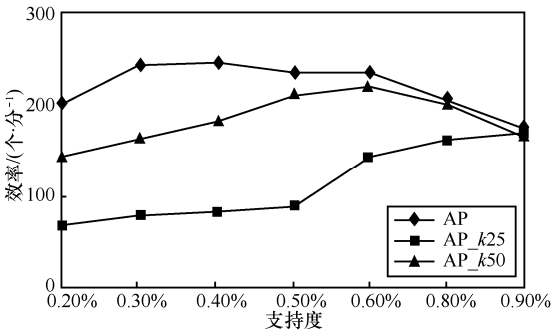


图 4　不同支持度下频繁项集生成效率的比较

从图 4 可以看出，*k*-pre 算法的频繁项集生成效率高于 Apriori 算法。实验对比和算法分析表明，通过 *k*-pre 算法挖掘数据时，候选项集产生了剪枝支持度较小的数据，在扫描数据库判断候选项支持度时提高了命中率，提升了频繁项集生成效率。由以上 2 个实验可知，通过控制支持度与前 *k* 项值参数，在提取相同数量频繁项集的情况下，达到了平衡冷、热数据的权重目的。

实验 3　本实验主要是为了证明推荐结果的质

量，即非空率。在提取相近数量的频繁项集中统计了被推荐项的数量。

如图 5 所示，随着支持度降低，频繁项集数量增加，推荐项的数量逐渐增大，当频繁项集数量大于 3 000 时，*k*-pre 算法的非空率要大于 Apriori 算法的。从实验结果中可以看出，*k*-pre 提高了整体推荐集合中推荐项的比重、系统推荐质量。

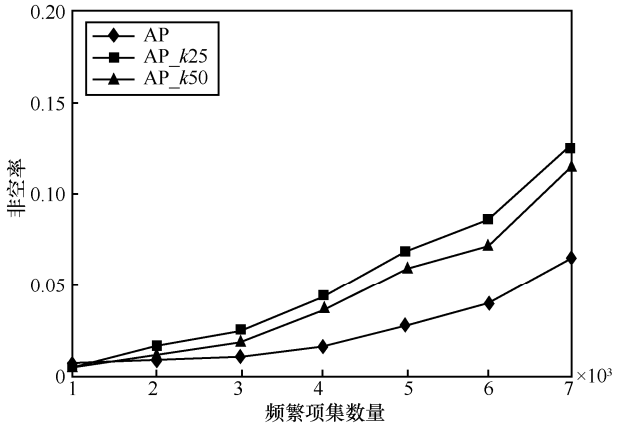


图 5　推荐非空率比较

实验 4　本实验主要比较 *k*-pre 算法与 Apriori 算法的覆盖率、准确率。在 *minSup* 为 0.002 时对强关联规则推荐数据进行统计，如图 6 所示，Apriori 算法在该支持度下推荐覆盖率为 0.53；*k*-pre 算法受前项过滤的影响，在指定支持度情况下，推荐覆盖率整体小于基础方法的覆盖率，该算法的覆盖率随着 *k* 值变大而呈线性增长。在图 7 中可以看出，在相同体积的推荐项中，*k*-pre 算法的覆盖率随频繁项数量增大而线性增长，增长到 0.7 之后，平缓上升。

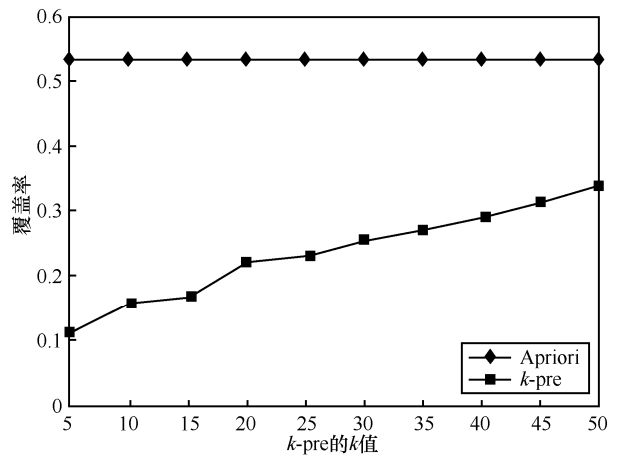


图 6　推荐覆盖率随 *k* 值变化的比较

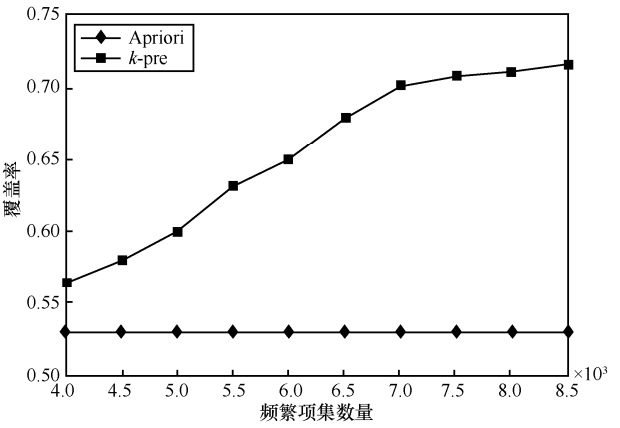


图 7　不同数量频繁项集下推荐覆盖率比较

k-pre 算法的准确率如图 8 所示，在固定支持度下，*k*-pre 算法的覆盖率虽然不占优势，但是准确率明显要高，且当 *k* 值越小，*k*-pre 算法的准确率越高；*k* 大于 50 时，准确率接近基础方法。

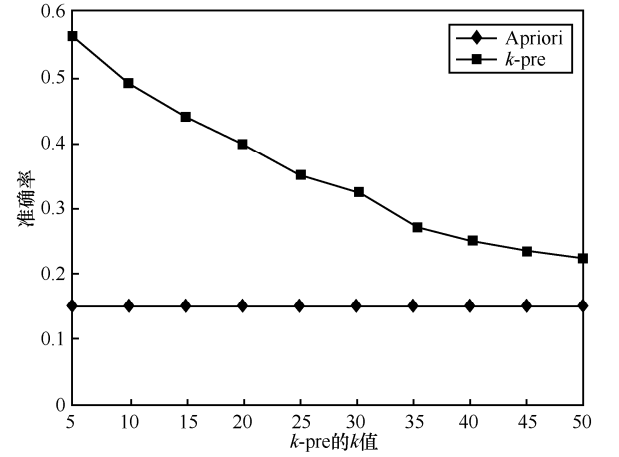


图 8　推荐准确率随 *k* 值变化的比较

从图 9 中可以看出，*k*-pre 算法中除 *k* 值小于 10 以外，其余 *k* 值下的调和平均值均大于基础方法设定支持度下的最大调和平均值。

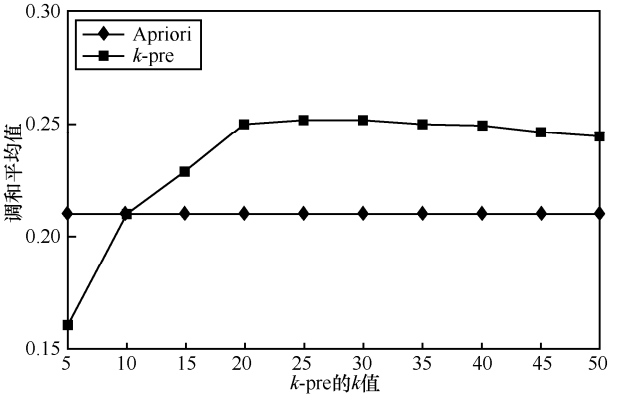


图 9　不同 *k* 值下推荐系统调和平均值比较

综合上述实验结果与分析，*k*-pre 算法无论在推荐效率，还是在推荐准确率、覆盖率与推荐平衡性等方面都具有一定优势，也验证了本文最初设想的合理性，即在频繁项集提取时，对候选项集前 *k* 项关联剪枝，提高系统非空率、频繁项集生成效率，给予那些数据集中权重较低的相应推荐，提高系统推荐质量。

5　结束语

个性化推荐作为信息大数据时代信息处理系统的重要研究方向之一，在很多领域得到了广泛应用。针对如何利用用户历史购物记录，挖掘用户购物行为、兴趣和习惯等，为用户推荐可能感兴趣的物品，本文做了以下工作：1) 提出并分析了推荐非空率 *RecNon* 这一新的评价标准，并将该评价标准应用于本文的关联规则挖掘方法；2) 定义了 *k* 前项关联规则 *k*-pre 概念，提出了相关定义和定理，并进行了证明；3) 设计了 *k*-pre 挖掘算法，详细描述了该算法的执行流程，并进行实例分析；4) 分析了 *k*-pre 算法的时间复杂度，并与 Apriori 算法做了对比。实验结果表明，*k*-pre 关联规则的方法不仅具有更好的推荐测度，并且提高了系统非空率、频繁项集生成效率，平衡了冷、热门数据所占权重。

为了更加高效准确地实现关联规则个性化推荐，还可以从以下方面继续深入完善：1) 将关联规则挖掘方法与其他数据挖掘方法结合，如协同过滤、聚类分析、内容推荐等，从而达到更加准确、全面的个性化推荐；2) 分析相关商品访问、浏览记录和商品属性、标签，并结合多方面内容分析挖掘关联规则，提高系统的推荐质量。

参考文献：

[1] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6):734-749.

[2] LIN W, ALVAREZ S, RUIZ C. Efficient adaptive-support association rule mining for recommender systems[J]. Data Mining and Knowledge Discovery, 2002, 6(1): 83-105.

[3] AGRAWAL R, IMIELINSKI T, SWAMI A. Mining association rules between sets of items in large databases[C]//The 1993 ACM SIGMOD International Conference on Management of Data. Washington, 1993: 207-216.

[4] PARK J S, CHEN M S, YU P S. An effective hash-based algorithm for mining association rules[J]. ACM SIGMOD Record, 1997, 24(2): 175-186.

[5] TOIVONEN H. Sampling large databases of association rules[C]//The 22th International Conference on Very Large Data Bases, San Francisco: Morgan Kaufmann Publishers Inc, 1996: 134-145.

[6] 魏玲, 魏永江, 高长元, 等. 基于 Bigtable 与 MapReduce 的 Apriori 算法改进[J]. 计算机科学, 2015, 42(10):208-210.
WEI L, WEI Y J, GAO C Y, et al. Improved Apriori algorithm based on Bigtable and MapReduce[J]. Computer Science, 2015, 42(10): 208-210.

[7] 刘兴彬, 杨建华, 谢高岗, 等. 基于 Apriori 算法的流量识别特征自动提取方法[J]. 通信学报, 2008, 29(12):51-59.
LIU X B, YANG J H, XIE G G, et al. Automated mining of packet signatures for traffic identification at application layer with Apriori algorithm[J]. Journal on Communications, 2008, 29(12):51-59.

[8] 王大玲, 于戈, 鲍玉斌. 一种具有最大推荐非空率的关联规则挖掘方法[J]. 软件学报, 2004, 15(8):1182-1188.
WANG D L, YU G, BAO Y B. An approach of association rules mining with maximal nonblank for recommendation[J]. Journal of Software, 2004, 15(8):1182-1188.

[9] SANDVIG J, MOBASHER B, BURKE R. Robustness of collaborative recommendation based on association rule mining[C]//The 2007 ACM Conference on Recommender Systems. 2007: 105-112.

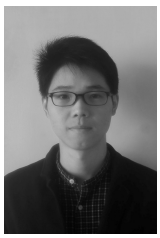
[10] HONG J Y, SUH E H, KIM J, et al. Context-aware system for proactive personalized service based on context history[J]. Expert Systems with Applications, 2009, 36(4): 7448-7457.

[11] CHEN W Y, CHU J C, LUAN J, et al. Collaborative filtering for orkut communities: discovery of user latent behavior[C]//International Conference on World Wide Web. 2009: 681-690.

[12] GARCIA E, ROMERO C, VENTURA S, et al. A collaborative educational association rule mining tool[J]. The Internet and Higher Education, 2011, 14(2): 77-88.

作者简介：

何明（1975-），男，陕西礼泉人，博士，北京工业大学副教授，主要研究方向为大数据、推荐系统、机器学习等。



刘伟世（1989-），男，山东菏泽人，北京工业大学硕士生，主要研究方向为推荐系统。

张江（1980-），女，北京人，博士，国网英大国际控股集团有限公司信息化工作部经理，主要研究方向为网络通信协议、网络自愈、大数据等。