2017 年 11 月 第 11 期
Vol. 38 No. 11 2017

基于概念邻域的 Top-N 推荐算法

陈昊文,王黎明,张 卓

(郑州大学 信息工程学院,郑州 450052)
E-mail: iehwchen@gs.zzu.edu.cn

摘 要: 做为处理信息过载的有效手段,推荐系统在短时间内得到了迅速的发展. 传统的基于邻域的方法忽略了用户与产品间的结构关系,只考虑了同类对象间的相似关系. 随着推荐系统的广泛应用,数据稀疏条件下的推荐问题也亟待解决. 针对推荐系统所面临的关键问题提出了一种面向隐式反馈数据的基于概念邻域的推荐算法. 将用户与产品的评分(关系)矩阵转化为二元形式背景,以此为基础构造出相应的概念格,将用户与产品分别以对象与属性的形式聚集在概念中,并通过概念间的偏序关系,以对象(用户)的起始概念为起点探索其近邻概念并获取候选项集,最后结合所提出的全局偏好度与邻域偏好度过滤出最终推荐结果. 该算法通过在两个公共数据集上的实验,相较于传统的基于邻域的推荐算法,具备较好的推荐效果,并更适用于数据稀疏条件下的推荐.

关 键 词: 推荐算法; 协同过滤; 形式概念分析; 概念格
中图分类号: TP399 **文献标识码:** A **文 章 编 号:** 1000-1220(2017)11-2553-07

Top-N Recommendation Algorithm Based on Conceptual Neighborhood

CHEN Hao-wen, WANG Li-ming, ZHANG Zhuo
(School of Information Engineering, Zhengzhou University, Zhengzhou 450052, China)

Abstract: As an effective mean to deal with information overload, the recommendation system has developed rapidly in a short time. The traditional neighborhood based approaches ignore the structural relationship among users or products, and only consider the similarity among the similar objects. With the recommendation system being applied extensively, the recommendation problem in the condition of data sparse also needs to be solved urgently. To solve the key problems the recommendation system faced with, a recommendation algorithm based on conceptual neighborhood is proposed for the implicit feedback data. The algorithm transforms the matrix of scores (relationships) between users and products into binary formal context, and constructs the corresponding concept lattice on the basis of formal context. so that the users and products are gathered in concepts in the form of objects and attributes respectively. As the starting point, the initial concept index can help users obtain the candidate item sets from their neighborhood through the order relationship among the concepts. Finally, the final recommendation can be generated through combining the proposed global preference and neighborhood preference. After the experiment on two public datasets, the proposed method with a better recommendation effect is more suitable for the recommendation under the data sparse condition, compared to the traditional neighborhood based recommendation algorithm.

Key words: recommendation algorithm; collaborative filtering; formal concept analysis; concept lattices

1 引 言

随着互联网的迅速发展,其用户数量也在急速增长,但伴随而来的是信息过载问题的出现. 海量的信息使用户在获取资源时无从下手. 为了解决这样的问题,推荐系统应运而生. 通过分析用户以及与用户相关的数据来推测出用户的兴趣所在和行为趋势,以此为基础向用户推荐他们所需要的信息和服务. 推荐系统的产生有效地解决了信息过载问题,使数据增长不再成为用户按需获取信息的阻碍. 目前推荐系统已经被广泛应用在了电子商务、视频网站、音乐电台和社交网络等领域.

推荐系统的核心部分主要由推荐算法构成,协同过滤(Collaborative Filtering)是目前应用最广泛的方法之一. 其核心思想是通过分析用户记录,在用户群中找到指定用户的相似(兴趣)用户,综合这些相似用户对某一信息的评价,形成系统对该指定用户对此信息的喜好程度预测. 协同过滤算法主要分为基于模型与基于内存两类[1]. 基于模型的方法通过对已有用户数据建模并进行推荐,例如最具代表性的 SVD 算法[2],之后 Koren Y 等人[3]又在此基础上提出的 LFM 算法,只针对已有的评分数据进行训练,不用对缺失数据预先填充. 后来在此基础上又衍生出了基于回归的协同过滤算法[4]、基于贝叶斯的推荐算法[5]等. 但当面对隐性反馈数据进行建模时,往往需要生成负样本,生成方法的不同直接影响了模型的训练结果,从而导致了模型的不稳定性. 基于内存的

收稿日期: 2016-09-27 收修改稿日期: 2016-11-23 基金项目: 国家青年科学基金项目(61303044)资助. 作者简介: 陈昊文,男,1990 年生,硕士研究生,研究方向为形式概念分析及应用、数据挖掘等; 王黎明,男,1963 年生,博士,教授,CCF 高级会员,研究方向为现代软件工程技术、分布式人工智能和数据挖掘等; 张 卓(通信作者),男,1978 年生,博士,副教授,研究方向为形式概念分析及应用等.

协同过滤算法主要依赖于邻域的偏好信息,分为基于用户(user-based)和基于物品(item-based)[6],基于物品的协同过滤算法则是目前业界应用最为广泛的算法,无论是亚马逊网[7],还是 Netflix,都是以该算法为基础构建的. 此外,也有一些算法[8,9]将两种方法结合在了一起. 但是传统的基于邻域的协同过滤算法仅仅挖掘用户或产品之间的相似关系,而忽略了两者间存在的关系结构. 并且在数据稀疏条件下,由于信息的大量缺失通常获取不到充足的邻域信息,直接影响了推荐效果. 随着形式概念分析的不断发展,其核心数据结构概念格已经广泛应用于数据挖掘、信息检索、数据抽取、推荐系统等领域[10,11],与其相关的构造维护算法[12]也不断更新. 2006 年 Boucher Ryan 等人[13]首次提出了将形式概念分析与协同过滤算法结合的思想,该文献将概念格作为用户记录信息的存储载体,为之后的进一步应用提供了方向. Tomohiro Murata 等人[14]结合形式概念分析提出了一种基于知识的推荐模型,Dmitry I. Ignatov 等人[15]则将概念格中的关联规则挖掘应用在了广告推荐领域. 也有研究者为了应对多值背景将模糊形式概念分析应用在了推荐问题中[16,17].

本文为了解决以上问题提出了基于概念邻域的协同过滤算法(concept neighborhood-based collaborative filtering,简称 CNCF),通过在原始数据基础上构造概念格,并根据概念之间的偏序关系寻找近邻概念来获取初始推荐候选项,再结合概念相似度对推荐候选项进行二次过滤,最终获得用户的推荐项. 通过实验验证,相较于传统的基于邻域的协同过滤算法,该算法能够取得较好的推荐效果,并且能更好地适应数据稀疏条件下的推荐.

2 形式概念分析

概念在哲学中被理解为外延与内涵所组成的思想单元,德国数学家 Wille 在 1982 年首先提出了形式概念分析,用于概念的发现、排序和显示[18]. 概念格作为形式概念分析(Formal Concept Analysis)的核心数据结构,是根据形式背景中对象与属性之间的二元关系建立的一种概念层次结构. 概念格能够通过 Hasse 图清晰地体现概念间的泛化和特化关系,因此被认为是进行数据分析的有力工具.

定义 1.[19](形式背景)一个形式背景 $K=(G,M,I)$ 是由两个集合 G 和 M 以及 G 与 M 间的关系 I 组成. G 的元素称为对象,M 的元素称为属性. $(g,m)\in I$ 或 gIm 表示对象 g 具有属性 m.

定义 2.[19](伽罗瓦联系)设 A 是对象集合 G 的一个子集,定义 $f(A)=\{m\in M\mid \forall g\in A, gIm\}$($A$ 中对象共同属性的集合). 相应地设 B 是属性集合 M 的一个子集,$g(B)=\{g\in G\mid \forall m\in B, gIm\}$(具有 B 中所有属性的对象的集合). 若 $A=g(B)$,$B=f(A)$,则称集合 A,B 满足迦罗瓦联系.

定义 3.[19](形式概念)背景 (G,M,I) 上的一个形式概念是二元组 (A,B),其中 $A\subseteq G$,$B\subseteq M$,且 A 与 B 满足迦罗瓦联系,则称 A 是概念 (A,B) 的外延,B 是概念 (A,B) 的内涵.

定义 4.[19](层次序)设 $C_1=(A_1,B_1)$ 和 $C_2=(A_2,B_2)$ 是格中的两个概念,且 $A_1\subseteq A_2$($B_1\supseteq B_2$),称 C_1 是 C_2 的子概念,C_2 是 C_1 的超概念,记为 $C_1\leqslant C_2$,关系 $\leqslant$ 称为概念格层次序.

定义 5.(邻域概念)当不存在概念 (A_3,B_3) 满足 $(A_1,B_1)<(A_3,B_3)<(A_2,B_2)$ 时,称 (A_1,B_1) 是 (A_2,B_2) 的子节点. 当概念 (A_1,B_1) 与 (A_2,B_2) 具有相同的上邻(或下邻)时,称 (A_1,B_1) 是 (A_2,B_2) 的兄弟节点.

定义 6.(起始概念)给定一个对象 a,且 $a\in A_1$,当不存在概念 $(A_2,B_2)<(A_1,B_1)$,并且 $a\in A_2$,概念 (A_1,B_1) 是对象 a 的起始概念.

根据定义 6,本文提出命题如下:

命题 1. 一个对象具有的属性集即为此对象起始概念的内涵集.

证明: 设对象 a 的属性集 $H=\{m_1,m_2,\cdots,m_k\}$,对象 a 的起始概念 $C=(A,B)$,由于 $a\in A$,可知 $B\subseteq H$. 当 $B\subset H$ 时,根据定义 2 可以构造概念 $C'=(g(H),H)$,再结合条件 $B\subset H$,得出 $C'<C$,显然这与起始概念的定义相矛盾,所以 $B=H$,命题 1 得证.

由命题 1 可知,对象 a 的起始概念的内涵集即为对象 a 的属性集,结合层次序的定义,对象 a 将不再出现在其起始概念的子节点概念的外延集中. 所以对象 a 的起始概念是包含对象 a 的概念中唯一能够完整描述对象 a 的特性的. 在第 3 节中会结合推荐问题详细阐述起始概念在算法中的作用.

3 概念格及起始概念索引的构造

3.1 基于用户记录的概念格的生成

推荐系统一般通过日志系统获取用户行为数据,并按照一定的格式进行再处理. 一条用户记录通常表示某用户对某物品在某一时刻进行了某种操作,包含用户标识、物品标识以及其他一些辅助信息. 例如一个 7×4 的用户-产品评分矩阵,其中包含了 7 位用户对 4 件产品的所有打分,如果分值为 0 则说明该用户未访问过该产品,如表 1 所示.

表 1 用户 - 产品评分矩阵

Table 1 User - item rating matrix

	I_1	I_2	I_3	I_4
U_1	0	3	0	2
U_2	0	0	1	5
U_3	0	0	0	2
U_4	5	0	0	0
U_5	1	3	4	0
U_6	0	0	1	0
U_7	4	2	0	0

虽然用户-产品矩阵能够清楚地反映用户与产品之间的关系,但并不能从中直接提取用户或产品的邻域信息. 而概念格具有明显的聚类特性,概念间的层序关系也能清楚反映对象(用户)集间的泛化与例化关系,通过概念之间的关系可以直接获取邻域信息.

根据概念格的定义,对照表 1,可以直接将用户-产品矩阵转化为形式背景. 用户集合作为形式背景中的对象集合,产品集合作为形式背景中的属性集合,用户与产品间的关系(评分)直接映射为对象与属性间的关系,如果某对象具备某种属性,则表示该用户对该产品进行了打分. 由于是在经典概念格范畴内研究问题,对象与属性之间只存在 0 和 1 的关系(二元关系),即对象

是否具有这个属性. 所以只关注用户是否对产品进行了评分,而不考虑评分值. 由表 1 中用户-产品矩阵转化而来的形式背景如表 2 所示. 由于需要利用概念格

表 2　形式背景

Table 2　Formal context

	I_1	I_2	I_3	I_4
U_1	0	1	0	1
U_2	0	0	1	1
U_3	0	0	0	1
U_4	1	0	0	0
U_5	1	1	1	0
U_6	0	0	1	0
U_7	1	1	0	0

的结构关系,所以采用渐进式算法中的 Godin 算法[20] 来构造概念格. 此算法是在有新的对象插入时,在原有概念格基础上进行调整. 基于表 2 中形式背景生成的概念格如图 1 右侧部分所示,节点上两排字符分别表示概念中的外延与内涵,即概念的属性以及包含这些属性的对象,节点与节点间的连线表示概念之间的关系.

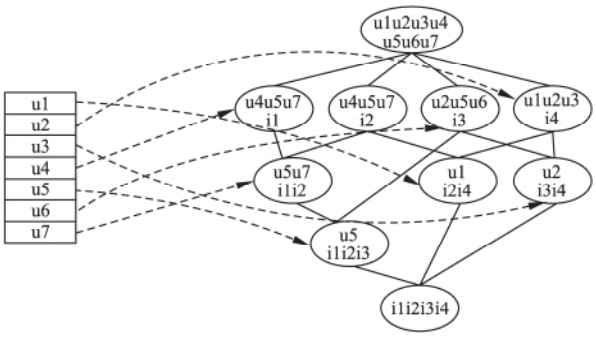

图 1　起始概念索引分概念格

Fig. 1　Initial concept index and concept lattices

3.2　起始概念索引的构造

假设对用户 u 进行推荐,虽然每个用户具有唯一标识,但是在概念格中,外延中包含用户(对象) u 的概念通常不止一个,所以需要从这些概念中确定唯一一个概念作为探索邻域的起点. 由定义 5 可知,对象 u 的起始概念包含了该对象具有的所有属性,也是所有包含该对象的概念中唯一包含其所有属性的概念,所以起始概念恰好描述了该对象的所有特性,更适合作为探索的起点. 然而在推荐的过程中,如果每次都要从概念格中定位目标对象的起始概念,则会产生大量重复性探索,显然这会影响整个算法的效率. 在数据库技术中,索引的引入显著提高了数据检索的效率. 如果可以一次性的建立所有用户(对象)与其起始概念之间的对应关系,那么当对用户进行推荐时,就可以通过索引直接确定与之对应的起始概念位置,有效避免了大量的重复检索. 借助哈希索引的思想,可将对象看作索引键,哈希函数表示对象与起始概念间的映射,定义如下:

定义 7. (起始概念索引) 设基于形式背景 $K=(G,M,I)$ 下的概念格为 C,对象 $a\in G$ 为索引键,则映射 $I=\{a\rightarrow c \mid \forall a\in G, c$ 为对象 a 的起始概念$\}$ 为联系对象与起始概念的哈希函数.

根据命题 1 可知,一个对象的起始概念的内涵集是唯一的,即该对象的属性集,所以该对象的起始概念在概念格中也具备唯一性. 根据以上理论基础,构造包含所有对象的起始概念索引的方法如下:

函数 1. ConstrutIndex( )
输入: 概念格 L
输出: 起始概念索引集 ICindex
说明: 构造所有对象的起始概念索引

```
for C in L
    if len(Map) == 所有对象的个数:
        break //构造完成
        for j in C.extent:
            if j in ICindex.keys( ):
                continue
            //设置标记变量 f,表示当前概念 C 是否是对象 j 的起始概念
            f = True
            for k in C.chd
                //判断概念 C 是否为对象 j 的起始概念
                if j in L[k].extent:
                    f = False
                    break
            if f is True:
                //将映射 j→C,即对象 j 与起始概念 C 的对应关系添加至索引集中
                ICindex.append(j→C)
```

函数 1 中对概念格的所有概念逐个遍历,因为在概念格的存储结构中每个概念都包含了其父节点及子节点的所有信息,所以可以直接获取概念的子节点,并根据起始概念的定义找出符合条件的对象. 图 1 左侧部分代表索引键(对象),虚线则代表对象与该对象起始概念的对应关系. 通过构造起始概念索引,使得在推荐时可以以目标对象作为索引键直接访问起始概念.

4　基于概念格邻域的 Top-N 推荐算法

4.1　问题描述

假设用户(对象) u 的起始概念为 C,结合之前定义不难分析出,对于概念 C,只有其兄弟节点和子节点的内涵集中可能包含未在概念 C 的内涵集中出现过的属性项,也就是一些潜在的推荐项. 根据以上分析将候选项集定义如下:

定义 8. (候选项集) 设对象 u 的起始概念为 $C=(A,B)$,概念 C 的所有兄弟节点,子节点为 $C_1=(A_1,B_1)$,$C_2=(A_2,B_2)$,$\cdots$,$C_k=(A_k,B_k)$ $(k\geqslant 0)$,则对象 u 的候选项集 $CS=B_1\cup B_2\cdots\cup B_k/B$.

结合文献[21]中对推荐系统的形式化定义,可以将基于概念邻域的 Top-N 推荐算法的形式化定义描述如下:

设 C 是所有用户的集合,S 是所有产品的集合,转化后的形势背景 $K=(C,S,I)$,G 为基于形势背景 K 生成的概念格,CS 为探索概念邻域获取的推荐候选项集,且 $CS\subseteq S$,设效用

函数 Pre( ) 可以计算用户 c 对产品 s 的偏好度,即 $Pre: C \times S \rightarrow R$,R 是一定范围内的全序非负实数,则推荐问题所要获取的结果是使偏好度值 R 最大的那些产品 s^*,如式(1)所示.

$$\forall c \in C, s^* = argmax_{s \in CS} Pre(c, s) \quad (1)$$

4.2 构造候选项集

根据概念间的偏序关系,越是处于概念格下层的概念中包含的对象越特殊,因为这些对象具有更多的属性. 在确定推荐对象即目标用户的起始概念后,本文利用概念格的结构优势,通过探索邻域概念直接获取推荐候选项. 为了充分构造候选项集,这里采用递归的方法探索起始概念的子节点及兄弟节点:

函数 2. GetItemOfChildren ( )
输入: 起始概念 Ci,递归深度 n
输出: 起始概念 Ci 的子节点候选项集 CSchd
说明: 递归探索子节点

```
1.  def  GetItemOfChildren ( Ci,n) :
2.       n = n - 1  //记录递归深度
3.       for c in CL [Ci] . chd:
4.            if  n! =0:
5.             //向下一层探索
6.             CSchd ← CSchd ∪ GetItemOfChildren( c, n)
7.            else:
8.              CSchd ←CSchd ∪ c. intent
9.       return  CSchd
```

函数 2 从起始概念出发,通过探索子节点获得候选项集 CSchd. 根据定义 4、定义 5 可知,一个概念的内涵集一定是它任意子节点所包含内涵集的真子集. 所以在第 5、6 行可以看出当递归进入下一层时并未将当前探索到的子节点中内涵集的项目添加到候选项集中. 只需如第 9 行所示,在递归进行至最后一层时将所有当前探索到的子节点中内涵集项添加到候选项集中,就能够能保证候选项集充分扩展.

函数 3. GetItemOfSiblings( )
输入: 起始概念 Ci,递归深度 n
输出: 通过起始概念 Ci 的兄弟节点获得候选项集 CSsib
说明: 递归探索兄弟节点

```
1.  def GetItemOfSiblings( Ci,n) :
2.       n = n - 1  //记录递归深度
3.       for i in CL [Ci] . fah:
4.         for j in CL [i] . chd:
5.           if  n! =0:
6.            CSsib ← CSsib ∪ CL [j] . intent
7.            CSsib ← CSsib ∪ GetItemOfSiblings( u,n)
8.           else:
9.             CSsib ← CSsib ∪ CL [j] . intent
10.      return CSsib
```

函数 3 的执行过程与函数 2 类似. 但不同的是,函数 3 第 6 行中将当前递归层中探索到的兄弟节点内涵集中的项目添加到了候选项集中,原因是兄弟节点的内涵集之间并不存在包含与被包含的关系,所以在每层递归需要将当前兄弟节点的内涵集项添加至候选项集才能保证构造充分.

通过将由函数 2 与函数 3 获得的候选项集 CSchd 与 CSsib 合并,并剔除目标对象(用户) 曾经访问过的产品项,获得最终的推荐候选项集.

4.3 用户对产品的偏好度

为了得到最终的推荐项,需要确定效用函数 Pre( ) 以计算产品对于用户的推荐度,并对每个用户的候选项集进行再次过滤,从而得到最终推荐项. 在 Top-N 推荐中需要从中筛选出用户最有可能感兴趣的 N 个物品,传统的基于邻域的协同过滤算法通常利用用户相似度或物品相似度的方法来计算产品对于用户的推荐度. 本文以概念格作为数据载体,结合提出的两种偏好度,定义了两种产品对于用户推荐度的计算方法. 首先定义两种偏好度: 全局偏好度与邻域偏好度.

4.3.1 全局偏好度

设 C = (U,I) ,C = (U′,I′) 为同一概念格中的两个概念,并且 i∈I,i′∈I′,那么基于全局偏好度计算方法如下:

$$G(u, i') = \frac{\sum_{i \in I} w(i, i')}{|I|} \quad (2)$$

在这里我们只考虑概念中的所包含的内容,用户 u 对物品 i′的全局偏好度是基于内涵集 I 中属性(产品) 与 i′之间的平均相似度定义的. 公式 2 中|I|表示概念 C 内涵集中的属性(产品) 个数,w(i,i′) 表示 i 与 i′的 Jaccard 系数,Jaccard 系数主要用于计算符号度量或布尔值度量的个体间的相似度,计算方法如公式(3) 所示:

$$w(i, i') = \frac{|U_i \cap U_{i'}|}{|U_i \cup U_{i'}|} \quad (3)$$

U_i,$U_{i'}$分别表示与物品 i,i′存在关系的用户集合,在全局偏好度中,通过计算起始概念中与目标用户相关的产品与推荐项的平均相似度定义了用户 u 对产品 i′偏好度.

4.3.2 邻域偏好度

设 C = (U,I) ,C′ = (U′,I′) 为同一概念格中的两个概念,且 i∈I,i′∈I′,基于概念格的邻域偏好度计算公式如下:

$$N(u, i') = \frac{1}{|N_{i'}|} \sum_{C' \in N_{i'}} sim(C, C') \quad (4)$$

公式(4) 中 $N_{i'}$代表概念 C 的子节点和兄弟节点中内涵中包含属性 i′的概念集合,sim(C,C′) 表示概念 C 与 C′之间的相似度. 概念实际上是由外延集与内涵集两个集合组成,通常的计算概念相似度方法会先分别求得两个概念中外延集间的相似度与内涵集间的相似度,之后通过对这两个相似度加权求和来计算. 这里我们采用 Faris Alqadah 等人提出的一种没有参数引用的概念相似性度量方法,此方法在计算过程中更好地考虑了对象与属性间的关联性,而不是将其看作两个独立的群体. 计算方法如公式(5) [23] 所示:

$$sim(c, c') = 1 - \frac{zero(D)}{|D|} \quad (5)$$

其中$|D| = |U \cup U'| \times |I \cup I'|$,D 表示 U∪U′为对象集,I∪I′为属性集的子形式背景,zero(D) 表示了形式背景 D 中零的个数. 在邻域偏好度的计算中,所有涉及的信息均为邻域概念(子节点或兄弟节点) 中的内容. 结合以上对全局偏好度与邻域偏好度的定义,下面给出两种效用函数 Pre( ) 的定义方式:

$$Pre1(u, i') = G(u, i') \quad (6)$$

$$Pre2(u, i') = G(u, i') \times N(u, i') \quad (7)$$

可以看出,Pre1 仅使用了公式(2)即全局偏好度来计算产品对用户的推荐度,Pre2 则通过全局偏好度与邻域偏好度的乘积来计算推荐度.

为了区分使用这两种效用函数的 CNCF 算法,将使用公式(6)作为效用函数的 CNCF 算法记为 CNCF-1,使用公式(7)作为效用函数的 CNCF 算法记为 CNCF-2.

5　实验结果与分析

5.1　实验设计

针对提出的基于概念邻域的协同过滤算法,为了评估其推荐效果,本文选取了基于邻域的协同过滤算法 userkNN(基于用户 k 近邻)和 itemkNN(基于物品 k 近邻)作为对比算法进行实验,并对召回率和准确率等推荐算法衡量标准进行计算比较.

5.2　实验环境与实验数据

实验在处理器为四核 3.40GHz,内存 4GB 的计算环境下进行. 所有算法均由 python 语言所实现. 实验数据来自 Movielens100k 与 Bookcrossing 两个公共数据集. 前者包含 943 个用户对 1682 部电影的 100000 条用户评分记录,后者则是 bookcrossing 图书社区 278858 个用户对 271379 本图书的 1149780 条评分记录. 实验采用 5 折交叉验证法,为方便表述,标记 Movielens 数据集为 T1,而 Bookcrossing 数据集本身数据量较大,从中随机抽取了 5064 个用户对 36145 本图书的评分记录构成数据子集,记为 T2.

5.3　评价标准

Top-N 推荐算法的效果评估一般通过准确率(precision)和召回率(recall)来度量. 令 R(u)是算法针对用户产生的推荐列表,T(u)是用户在测试集上的行为列表,那么推荐结果的准确率定义为:

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \tag{8}$$

推荐结果的召回率定义为:

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \tag{9}$$

覆盖率(coverage)能够描述一个推荐算法对物品长尾的发掘能力,本文通过以下方法计算覆盖率:

$$Coverage = \frac{|\cup_{u \in U} R(u)|}{|I|} \tag{10}$$

5.4　实验结果分析

5.4.1　召回率与准确率

为了全面评测 CNCF 算法推荐的召回率和准确率,选取不同的推荐列表长度 N = 1,5,10,并分别在每个推荐列表长度下进行实验.

表 3、表 4 所示为四种算法在 T1、T2 上的召回率与准确率. 通过比较可以看出,itemkNN 算法的召回率与准确率是最低的,这也与其近邻产品的选择方式有着密切关系. 由于本文集中讨论基于邻域的推荐方法,并未考虑产品属性的相关信息,而主要通过同时访问过两个产品的用户数来定义产品间的相似度,加上用户-产品矩阵通常都具有较大的稀疏度,导致了其近邻产品选择的局限性,影响了推荐结果. 同样是基于邻域的方法,从用户角度出发的 userkNN 则取得了较高的召回率,通过聚集具有相似访问的用户,将其中用户访问较多的产品进行推荐. 由于 T1 与 T2 分别为电影与书籍类的数据集,比较容易在具有相思特质的群体内传播共享,所以 userkNN 的效果要明显优于 itemkNN.

表 3　不同算法在 T1 与 T2 上的召回率

Table 3　Recalls of different algorithms on T1 and T2

数据集	N	userkNN	itemkNN	CNCF-1	CNCF-2
T1	1	0.032	0.025	0.028	0.034
	5	0.119	0.098	0.111	0.121
	10	0.197	0.161	0.183	0.199
T2	1	0.0078	0.0075	0.0078	0.0076
	5	0.0261	0.0246	0.0258	0.0264
	10	0.0369	0.0343	0.0381	0.0395

对于本文提出的 CNCF-1 与 CNCF-2 算法,其两项指标均优于 itemkNN 算法,相比 CNCF-1,CNCF-2 算法的召回率与准确率更有优势,与 userkNN 算法相近. CNCF-1 算法在计算产品推荐度时没有考虑概念间的相似度,仅基于候选产品

表 4　不同算法在 T1 与 T2 上的准确率

Table 4　Precisions of different algorithms on T1 and T2

数据集	N	userkNN	itemkNN	CNCF-1	CNCF-2
T1	1	0.284	0.214	0.248	0.292
	5	0.2112	0.17	0.194	0.21
	10	0.176	0.1398	0.1598	0.1732
T2	1	0.0277	0.0265	0.0275	0.0268
	5	0.0184	0.0173	0.0182	0.1086
	10	0.0139	0.0121	0.0134	0.0139

与用户已访问产品间的 jaccard 系数来定义其推荐度,再加上用户访问数据的缺失,对最后的推荐效果还是产生了一定的影响. 而 CNCF-2 在综合考虑了产品之间与概念之间的相似度后,明显取得了更好的推荐效果.

5.4.2　覆盖率

覆盖率能够描述一个推荐系统对物品长尾的发掘能力. 一个好的推荐系统不仅需要有较高的用户满意度,也要有较高的覆盖率,这样才能避免出现越是热门的产品越容易被推荐,而越是冷门的物品越无人问津的情况出现. 如表 5 所示,userkNN 算法的覆盖率在两个数据集中都是最低的,这也反应了 userkNN 算法的根据相似用户进行推荐的核心思想,往往推荐的是在这个相似用户群体中比较热门的物品,从而弱化了对物品长尾的发掘. 相反,ItemkNN 的覆盖率在两个数据集中都维持在较高水准. CNCF-1 与 CNCF-2 算法的覆盖率均高于 userkNN 算法,而 CNCF-1 算法的覆盖率均低于 itemkNN 与 CNCF-2 算法. 可以看出,CNCF-1 与 CNCF-2 算法在获得较高的准确率和召回率的同时,也能保持较高的覆

表 5　不同算法在 T1 与 T2 上的覆盖率

Table 5　Coveroge rates of different algorithms on T1 and T2

	userkNN	itemkNN	CNCF-1	CNCF-2
T1	0.189	0.256	0.239	0.248
T2	0.184	0.228	0.192	0.219

盖率.

5.4.3 在稀疏数据下的性能评测

为了评测算法在数据稀疏情况下的推荐性能,在数据密度较低的 T2 上,通过按一定概率删除数据集中记录的方法,来模拟数据密度逐渐降低过程. 选取消减概率 p = [0.2,0.4,0.6,0.8],随着概率 p 的增加,数据稀疏度逐渐增大,在推荐列表长度为 10 的条件下,对 4 种算法的推荐结果进行评估计算.

图 2、图 3 展示了 T2 数据集上的四种算法的在不同消减概率下的召回率与准确率. 可以明显看出 itemkNN 算法对于稀疏数据的适应性是最差的,userkNN 在 p = 0.2 之后的准确率与召回率的下降速率虽然减缓,但其召回率与准确率均低于在 CNCF-1 和 CNCF-2 算法. 随着消减概率的增加,传统的 itemkNN 和 userkNN 算法都会因为数据愈发稀疏而获取不到足够的邻域信息,从而直接影响了最终的推荐效果. 由图 1 可以看出,如果把概念看做顶点,则其 hasse 图为连通图,从其中任意一概念出发,都能到达其他任意概念,再结合函数 2、函数 3 的递归探索过程,能够很容易访问到邻域概念,并构造候选项集,保证了在相对稀疏的形式背景下,依然能够获取足够的候选项进行推荐. 综上所述,本文提出的 CNCF-1 与 CNCF-2 算法在数据稀疏的情况下能够取得更好的推荐效果.

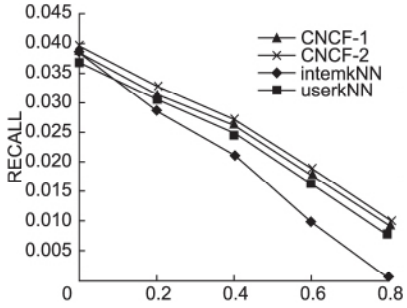

图 2 T2 中不同消减概率下的召回率

Fig. 2 Recalls of different reduction probabilities in T2

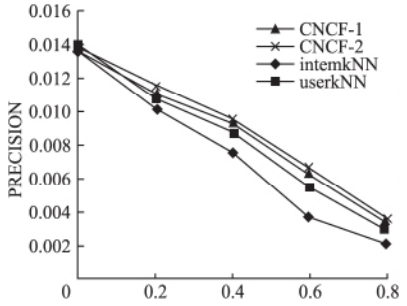

图 3 T2 中不同消减概率下的准确率

Fig. 3 Precisions of different reduction probabilities in T2

6 总 结

协同过滤算法作为推荐系统产生之初就出现推荐算法之一,时至今日依然在不同的推荐背景下发挥着重要作用,本文结合形式概念分析的知识,将概念格应用于推荐问题中,提出了一种基于概念邻域的 Top-N 推荐算法(CNCF). 通过其格结构以及概念间的关系,快速获得候选产品. 并结合所提出的全局偏好度与邻域偏好度定义了两种全新的推荐度度量方法,用于对候选项集的过滤. 通过实验,验证了 CNCF 算法相较于传统的基于邻域的协同过滤算法能够在保持较高的推荐质量同时,并且对数据稀疏的推荐环境具有更好的适应性. 由于 CNCF 算法所做的推荐仅仅面向隐式反馈数据,即形式背景下的概念格. 而现实中往往存在着一些类似于用户对产品的评分等能够直接体现用户对产品的喜爱程度的数据. 如何利用这些数据进行推荐,将形式背景下 CNCF 算法推广到多值模糊形式背景下是今后研究的方向.

References:

[1] John S Breese, David Heckerman, Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering [J]. Fourteenth Conference on Uncertainty in Artificial Intelligence, 2013, 7(7): 43-52.

[2] Polat H, Du W. SVD-based collaborative filtering with privacy [C]. Acm Symposium on Applied Computing, 2005, 1: 791-795.

[3] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42(8): 30-37.

[4] Vucetic S, Obradovic Z. Collaborative filtering using a regression-based approach [J]. Knowl Inf Syst, 2005, 7(1): 1-22.

[5] Rendle S, Freudenthaler C, Gantner Z, et al. Bayesian personalized ranking from implicit feedback [C]. Proceedings of the Twentyfifth Conference on Uncertainty in Artificial Intelligence, 2009: 452-461.

[6] Deshpande M, Karypis G. Item-based top-n recommendation algorithms [J]. ACM Trans Inf Syst, 2004, 22(1): 143-177.

[7] Linden G, Smith B, York J. Amazon. com recommendations item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76-80.

[8] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C]. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008: 426-434.

[9] Xue G-R, Lin C, Yang Q, et al. Scalable collaborative filtering using cluster-based smoothing [C]. Sigir: International Acm Sigir Conference on Research & Development in Information Retrieval, 2005: 114-121.

[10] Wang Li-ming, Zhang Zhuo. An algorithm for mining closed frequent itemsets based on apposition assembly of iceberg concept lattices [J]. Journal of Computer Research and Development, 2007, 44(7): 1184-1190.

[11] Zhang Zhuo, Du Juan, Wang Li-ming. Formal concept analysis approach for data extraction from a limited deep-web database [J]. Journal of Intelligent Information Systems, 2013, 41(2): 211-234.

[12] Jiang Qin, Zhang Zhuo, Wang Li-ming. Algorithms of constructing concept lattice based on deleting multiple attributes synchronously [J]. Journal of Chinese Computer Systems, 2016, 37(4): 646-652.

[13] Boucher-Ryan P D, Bridge D, et al. Collaborative recommending using formal concept analysis [J]. Knowledge-Based Systems, 2006, 19(5): 309-315.

[14] Li X, Murata T, et al. A knowledge-based recommendation model utilizing formal concept analysis [J]. International Conference on Computer & Automation Engineering, 2010, 4: 221-226.

[15] Dmitry I Ignatov, Sergei O. Kuznetsov, et al. Concept based recommendations for Internet advertisement [C]. 6th International Conference on Concept Lattices and Their Applications, 2008.

[16] Maio C D, Fenza G, Gaeta M, et al. Rss based elearning recommendations exploiting fuzzy FCA for knowledge modeling [J]. Applied Soft Computing, 2012, 12(1): 113-124.

[17] Fang P, Zheng S. A research on fuzzy formal concept analysis based collaborative filtering recommendation system [J]. 2nd International Symposium on Knowledge Acquisition and Modeling, 2009, 3: 352-355.

[18] Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts [M]. Berlin: Springer, 2009.

[19] Ganter B, Wille R. Formal concept analysis: mathematical foundations [M]. Berlin: Springer Science & Business Media, 2012.

[20] Godin R. Incremental concept formation algorithm based on Galois (concept) lattices [J]. Computational Intelligence, 1995, 11(2): 246-267.

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

[22] Alqadah F, Bhatnagar R. Similarity measures in formal concept analysis [J]. Ann Math Artif Intell, 2011, 61: 245-256.

附中文参考文献:

[10] 王黎明, 张　卓. 基于 iceberg 概念格并置集成的闭频繁项集挖掘算法 [J]. 计算机研究与发展, 2007, 44(7): 1184-1190.

[12] 姜　琴, 张　卓, 王黎明. 基于多属性同步消减的概念格构造算法 [J]. 小型微型计算机系统, 2016, 37(4): 646-652.

征 稿 简 则

一、征稿范围 《小型微型计算机系统》杂志刊登文章的内容涵盖计算技术的各个领域(计算数学除外). 包括计算机科学理论、体系结构、计算机软件、数据库、网络与通讯、人工智能、信息安全、多媒体、计算机图形与图像、算法理论研究等各方面的学术论文.

二、来稿要求: 本刊主要刊登下述各类原始文稿:

1. 学术论文: 科研成果的有创新、有见解的完整论述. 对该领域的研究与发展有促进意义, 论文字数最好在 10000 字左右.
2. 综述: 对新兴的或活跃的学术领域或技术开发的现状及发展趋势的全面、客观的综合评述.
3. 技术报告: 在国内具有影响的重大科研项目的完整的技术总结.

三、注意事项

1. 来稿务求做到论点明确、条理清晰、数据可靠、叙述简练, 词义通达.
2. 来稿必须是作者自己的科研成果, 无署名和版权争议. 引用他人成果必须注明出处.
3. 本刊采用在线投稿方式, 可登陆 http://xwxt.sict.ac.cn/进行在线投稿.
4. 格式要求: 题目(中、英文)、摘要(中、英文)、作者的真实姓名(中、英文)、作者的单位、城市(中、英文)、邮政编码、E-mail(便于联系的)、关键词(中、英文 4～7 个), 中图分类号、作者简介、基金项目.

(1) 英文部分的作者姓名使用汉语拼音, 单位英文名称须给出英文全称, 不要使用缩略语;

(2) 作者简介包含作者姓名、性别、出生年、最高学历、技术职称、研究方向(若作者中有中国计算机学会(CCF)会员, 请注明, 并给出会员号). 凡第一作者为 CCF 会员/高级会员/学生会员者, 将享受八五折的版面费优惠;

(3) 基金项目的类别与项目编号.

5. 中、英摘要: 文章摘要具有独立性和自明性, 含正文等量的主要信息, 一般为 250～300 字, 采用第三人称表述.
6. 参考文献: 未公开发表的文献不得列入. 文后所列参考文献统一排序, 且必须在正文中引用. 中文参考文献应给出对应的英文译文.

(1) 图书　[编号] 作者姓名(姓在前, 名在后), 书名, 出版社地址, 出版社, 出版年.

(2) 期刊　[编号] 作者姓名、文章题目、刊物名称, 出版年, 卷号(期号): 起止页码.

(3) 会议论文　[编号] 作者姓名. 论文题目. 见: 编者、论文集全名、出版地: 出版者, 出版年, 起止页码.

(4) 网络文献: 请给出文献作者或单位名, 文章题目、网址、发布日期.

7. 插图和表: 插图必须精绘并用计算机激光打印, 一般不超过 7 幅. 图应结构紧凑, 不加底纹, 不要做成彩色的, 图宽最好不超过 8 厘米, 图内字号统一使用 6 号宋体, 字迹、曲线清晰, 必要时给出坐标名称和单位. 每个图、表均给出中英文标题.
8. 计量单位: 稿件中一律使用《中华人民共和国法定计量单位》. 外文和公式中应分清大、小写和正、斜体, 上、下角的字母、数码位置准确. 易混淆的字母或符号, 请在第一次出现时标注清楚.
9. 本刊在收到作者稿件经初审后立即给作者电子邮箱发"稿件收到通知". 除作者另有明确要求外, 本刊原则上只与第一作者联系, 作者投稿后若 4 个月无消息, 可自行改投它刊. 通过初审的稿件将收到本刊给予的编号. 录用后的定稿不允许修改作者姓名等涉及著作权的内容, 以避免不必要的纠纷。
10. 本刊对不拟录用的稿件只发给"退稿通知", 恕不退回原稿, 请自留底稿.
11. 稿件一经发表, 将酌致稿酬, 并寄送样刊.

本刊文章现被国内外多家数据库收录, 作者著作权使用费与本刊稿酬一并给付, 作者若不同意将文章收录, 请在投稿时说明.

编辑部地址: 沈阳市东陵区南屏东路 16 号《小型微型计算机系统》编辑部　邮政编码: 110168

电　话: (024) 24696120　E-mail: xwjxt@sict.ac.cn　网　址: http://xwxt.sict.ac.cn