DOI：10.13196/j.cims.2008.07.203.yaozh.012

文章编号: 1006 – 5911(2008) 07 – 1449 – 08

集成项目类别与语境信息的协同过滤推荐算法

姚 忠[1], 吴 跃[1], 常 娜[2]
(1. 北京航空航天大学 经济管理学院, 北京 100083; 2. 国家信息中心, 北京 100045)

摘 要: 为改进基于项目的协同过滤推荐算法的推荐效果, 在项目相似性计算时引入项目类别因素的影响, 得出新的推荐算法, 即基于项目类别的修正条件概率相似性, 并在此基础上提出集成语境信息的多维推荐模型。通过与相关相似性、余弦相似性和修正余弦相似性的数值实验对比, 证明在数据比较稀疏的情况下, 改进算法所获得的推荐效果有较大提高。

关键词: 电子商务推荐系统; 协同过滤; 项目相似性; 项目类别; 语境信息; 条件概率; 数据稀疏性
中图分类号: TP39 文献标识码: A

Collaborative filtering recommender algorithm for integrating item category and contextual information

YAO Zhong[1], *WU Yue*[1], *CHANG Na*[2]
(1. School of Economics & Management, Beihang University, Beijing 100083, China;
2. State Information Center, Beijing 100045, China)

Abstract: To improve recommendation result of the item-based collaborative filtering algorithm, influence of product category in product similarity computation was introduced, and a new recommendation algorithm, i. e. Category-based Adjusted Conditional Probability similarity (CACP), was proposed. Base on this algorithm, multi-dimension recommendation model for integrated contextual information was also presented. Experiment was conducted to compare correlation similarity, cosine similarity and adjusted cosine similarity. Results showed that recommendation result of CACP was greatly improved especially in sparse data environment..

Key words: e-commerce recommender system; collaborative filtering; item similarity; item category; contextual information; conditional probability; data sparsity

0 引言

电子商务的飞速发展对企业服务提出了诸多新要求, 包括商品质量的保证、送货及时性、商品选购舒适度、退货便利性等, 其中最为突出的就是商品选购的个性化推荐[1]。信息检索技术满足了人们一定的需要, 但由于通用性问题, 仍不能满足不同背景、不同目的和不同时期的信息需求[2]。在此背景下, 推荐系统(recommender systems) 应运而生。电子商务推荐系统是为解决 Internet 上的信息过载问题而提出的一种智能代理系统, 利用电子商务网站向客户提供商品信息和建议, 帮助用户决定购买何种产品, 模拟销售人员帮助用户完成购买过程[1]。电子商务推荐系统收集用户购买商品的历史记录、点击记录、商品信息等相关可利用资源, 通过推荐算法来预测用户的偏好, 将最符合用户偏好的商品(或者若干商品) 推荐给用户。当前对电子商务推荐系统的研究最多而且最核心的是推荐算法的研究, 目前

收稿日期: 2007-09-06; 修订日期: 2007-11-21。 Received 06 Sep. 2007; accepted 21 Nov. 2007.
基金项目: 国家自然科学基金资助项目(70401001)。 **Foundation item:** Project supported by the National Natural Science Foundation, China(No. 70401001).

作者简介: 姚 忠(1964 –), 男, 河北张北人, 北京航空航天大学经济管理学院副教授, 博士, 主要从事决策支持系统与智能系统、供应链管理、基于知识管理的信息系统体系结构的研究。E-mail: iszhyao@buaa.edu.cn。

主要的推荐方法包括基于内容的过滤、协同过滤, 以及和数据挖掘技术融合的推荐方法[3]。随着电子商务系统的发展, 推荐算法也面临一系列问题和挑战。本文针对推荐算法存在的相关问题, 在算法中引入项目类别和多维语境信息, 对基于项目的协同过滤推荐算法进行了改进, 用改进的基于项目的协同过滤推荐算法进行评分预测及项目推荐。

1 集成项目类别的基于项目的协同过滤推荐算法

1.1 基于项目的协同过滤算法

Sarwar[4]于 2001 年提出基于项目的协同过滤推荐算法, 其算法中至关重要的一个步骤是计算项目之间的相似性, 选择与目标项目最相似的项目来预测用户对目标项目的评分, 将预测评分最高的项目推荐给用户。基于项目的协同过滤推荐算法, 是作为基于用户的协同过滤推荐算法的补充而发展起来的, 最初的目的是为了解决基于用户算法的数据稀疏性和扩展性两大问题。

(1) 数据稀疏性 推荐系统中, 每个用户一般都只对很少的项目做出评分, 从而造成评价矩阵数据相当稀疏, 难以找到相似用户集或者项目集, 大大降低了推荐效果[5]。

(2) 扩展性 现有大部分协同过滤推荐算法的计算量是随着用户和项目数目的增加而急剧增加的, 对于上百万的数据量, 传统的算法将遇到极大的扩展性问题, 严重影响推荐系统的实时性能[6]。

基于项目的协同过滤算法的基本思想是: 根据用户—项目评分矩阵求出不同项目之间的相似关系, 然后利用这些相似关系, 通过发现与用户喜欢的项目的相似项目, 来预测出目标用户最可能感兴趣的项目。该算法的核心概念是假设项目与项目之间有某种关联, 项目之间的关联越紧密, 则其相似性就越大[7]。项目相似性计算与评分预测是协同过滤算法最重要的两个步骤。

在大多数协同过滤系统中, 项目信息的更新比较慢, 项目之间的关系相对比较稳定, 所以可以预先离线计算好项目之间的相似性, 然后用查找表进行快速查找, 既节省了在线计算时间, 一定程度上也解决了系统的实时性问题, 同时又解决了基于用户的协同过滤可能存在的扩展性问题。

1.2 改进算法提出的原因

当前, 国内外学者对基于项目的协同过滤推荐算法的研究主要集中于两个方面: ① 对基于项目的协同过滤推荐算法的思想及具体算法的改进, 特别是项目相似性计算的改进; ②基于项目的协同过滤推荐算法同其他推荐算法的结合。

基于项目的协同过滤推荐算法通过与目标项目最相近的若干邻居项目的已有历史评分来预测目标项目的评分, 查询目标项目的若干最近邻居项目是通过度量不同项目之间的相似性来实现, 因而项目相似性计算是最近邻居项目查询的基础, 是基于项目的协同过滤推荐算法研究的最热点。

当前度量项目相似性的方法主要有余弦相似性、相关相似性、修正的余弦相似性和条件概率等[4, 8-9]。

(1) 余弦相似性(cosine-based similarity)

将项目 i 与项目 j 作为 m 维用户空间中的两个矢量, 项目之间的相似程度用这两个矢量之间的夹角余弦来衡量。设项目 i 和 j 在 m 维用户空间上的评分分别表示为向量 $\boldsymbol{a}$, $\boldsymbol{b}$, 则项目 i 和项目 j 之间的相似性

$$sim(i,j)=\cos(\boldsymbol{a},\boldsymbol{b})=\frac{\boldsymbol{a}\times\boldsymbol{b}}{\|\boldsymbol{a}\|\times\|\boldsymbol{b}\|}。\quad(1)$$

式中分子为两个项目评分矢量的内积, 分母为两个矢量模的乘积, 夹角越小, 相似度越高。

(2) 相关相似性(correlation-based similarity)

集合 U 表示同时对项目 i 和 j 评分过的用户, 通过两个项目 i 和 j 之间的 Pearson 系数作为它们之间的相似系数:

$$sim(i,j)=\frac{\sum_{u\in U}(R_{u,i}-\overline{R_i})(R_{u,j}-\overline{R_j})}{\sqrt{\sum_{u\in U}(R_{u,i}-\overline{R_i})^2}\sqrt{\sum_{u\in U}(R_{u,j}-\overline{R_j})^2}}。\quad(2)$$

式中: $R_{u,i}$ 表示用户 U 对项目 i 的评分, $\overline{R_i}$ 和 $\overline{R_j}$ 分别表示集合 U 内用户对项目 i 和 j 评分的均值。

(3) 修正的余弦相似性(adjusted cosine similarity)

余弦相似性有一个严重的缺陷即它没有考虑不同用户评分尺度的不同。修正的余弦法通过从评分中减去相应用户的评分的平均值来克服这个缺陷。

$$sim(i,j)=\frac{\sum_{u\in U}(R_{u,i}-\overline{R_u})(R_{u,j}-\overline{R_u})}{\sqrt{\sum_{u\in U}(R_{u,i}-\overline{R_u})^2}\sqrt{\sum_{u\in U}(R_{u,j}-\overline{R_u})^2}}。\quad(3)$$

式中 $\overline{R_u}$ 是用户 $u\in U$ 对全部项目评分的平均值。

协同过滤推荐算法度量用户之间的相似性, 最近邻居的选择是否合理, 直接影响推荐的准确率, 然而随着电子商务系统规模扩大, 用户数目和项目数据急剧增加, 导致用户评分数据呈现极端稀疏性, 造成传统的相似性度量方法得到的目标用户的最近邻居不准确, 算法质量低下[10]。此外, 由于用户对于所有评分为 0 的项目的偏好程度不可能完全相同, 余弦相似性并不能有效地在没有经过处理的用户—项目矩阵的基础上度量项目之间的相似性[11]。

此外, 传统项目相似性算法忽略了项目之间本身存在的项目类别关系, 即同属于某一个类别的项目之间应该有更高的相似性。如果项目 i 和 j 属于共同类别的数目越多, 这两者应该有更高的相似性, 这种考虑在数据比较稀疏的情况下会有更现实的意义。

1.3　改进算法步骤

本文在对项目进行分类的基础上采用条件概率来计算项目之间的相似性, 即基于项目类别的修正条件概率相似性(Category-based Adjusted Conditional Probability similarity, CACP)计算方法。项目相似性和评分预测直接决定推荐系统的推荐质量, CACP 算法对项目相似性计算和评分预测公式都有所改进, 目的是引入项目类别因素对推荐算法的影响, 以求达到更好的推荐效果和性能上的提高。

对项目进行分类是对项目相似性算法进行改进的出发点。在各种不同的系统中, 对于自身所提供的项目往往以大的类别来划分(如电影网站将所有的电影按照动作片、喜剧片等进行不同的归类)。显然, 一个项目可以同时属于多个项目类别(如电影既是历史剧, 又是言情剧)。考虑项目之间的类别差异, 能够使得同属于一个类别的项目间的相似性更大, 从而提高推荐算法的推荐精度。

用户偏好信息的获取是推荐算法的前提。用户信息的获取主要是通过用户对给定信息的评价, 包括显式评价和隐式评价两类。显式评价基于用户有意识地表达对项目的认可程度, 通常使用特定区间的整数值来表达用户的偏好程度, 用户数据库中的信息随着用户不断使用而随时更新; 隐式评价不需要用户主动参与, 推荐系统通过 Agent 等技术自动跟踪并分析用户浏览记录、购物记录等行为来获取信息。本文的用户信息获取基于显式评价, 通过用户对项目的评分记录(用户, 项目, 评分)转化形成用户—项目评分矩阵 $\boldsymbol{R}$。用户初始评分确定后, 推荐系统将依据评分函数 R_a 来预测用户对未评分项目的评分, 进而产生推荐。

$$R_a: Users \times Item \rightarrow Ratings。 \tag{4}$$

CACP 算法的基本步骤如下:

输入信息: 用户—项目评价矩阵 $\boldsymbol{R}$、项目类别属性表、项目类别调和参数 α, 以及项目最近邻居个数 k。其中, 参数 α 的作用是调和项目类别对项目相似性计算结果的影响; 参数 k 的作用是设定了对未评分项目 i 进行评分预测时所需要的最近邻居项目个数。

输出信息: 预测目标用户 u 评分最高的项目 i 或者评分最高的 n 个项目(*Top-N* 推荐)。

步骤 **1**　根据项目类别属性表中各项目所属类别, 建立项目所属类别矩阵 $\boldsymbol{S}$[12]。其中: n 行表示有 n 个项目, h 列表示总共有 h 个类别属性。

$$\boldsymbol{S}=\begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1h} \\ S_{21} & S_{22} & \cdots & S_{2h} \\ \vdots & \vdots & & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nh} \end{bmatrix}。 \tag{5}$$

矩阵 $\boldsymbol{S}$ 中的元素

$$S_{ij}=\begin{cases} 1 & \text{项目 } i \text{ 属于类别 } j \\ 0 & \text{项目 } i \text{ 不属于类别 } j \end{cases}。 \tag{6}$$

步骤 **2**　根据项目类别矩阵 $\boldsymbol{S}$, 建立修正的项目类别矩阵 $\boldsymbol{P}$, 其中 p_{ij} 表示项目 i 和 j 属于共同的项目类别的数目。例如, 如果项目 i 和 j 同时属于两个项目类别, 则 $p_{ij}=2$; 如果项目 i 和 j 不同时属于任何一个项目类别, 则 $p_{ij}=0$。

$$\boldsymbol{P}=\begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix}。 \tag{7}$$

建立项目类别矩阵是为了计算不同的项目是否属于同一个类别, 以及属于共同的类别的多少。如果这些项目属于不同类别的数目越多, 则相似性就应该越大。显然, 在修正的项目类别矩阵 $\boldsymbol{P}$ 中, 其元素的值以主对角线为轴对称分布, 即 $p_{ij}=p_{ji}$(项目 i 与项目 j 所属的共同的项目类别的数目等于项目 j 和项目 i 所属的共同的项目类别的数目)。在矩阵 $\boldsymbol{P}$ 中, 主对角线的元素(即项目 i 与自身所属的共同的项目类别的数目)是没有意义的, 也不参与最终相似度的计算。

步骤 **3**　根据用户—项目矩阵 $\boldsymbol{R}$ 及修正的项目类别矩阵 $\boldsymbol{P}$, 应用改进的基于项目类别的修正条件

概率式(8),计算项目 i 和项目 j 的相似性:

$$sim(i,j)=\frac{Freq(ij)+p_{ij}\times\alpha}{Freq(i)+Freq(j)-Freq(ij)+\sum_{u=1}^{m}|r_{u,i}-r_{u,j}|}。\tag{8}$$

式中: $Freq(i)$ 表示对项目 i 评分的用户数目; $Freq(j)$ 表示对项目 j 评分的用户的数目; $Freq(ij)$ 表示同时对项目 i 和 j 有过评分的用户的数目; $r_{u,i}$ 表示用户 u 对项目 i 的评分,在分母上添加了表达式 $\sum_{u=1}^{m}|r_{u,i}-r_{u,j}|$,以弱化评分差距过大带来的负面影响。为区别不同类别项目之间的联系,加入了 $p_{ij}\times\alpha$, p_{ij} 的值来源于修正的项目类别矩阵 $\boldsymbol{P}$。在式(8)中用调和参数 α 与项目类别数目的积来调节两个项目 i 和 j 的相似性,这将使得属于共同项目类别数目越多的两个项目之间的相似性越大。

步骤 **4** 根据步骤 3 得到的项目相似性计算结果 $sim(i,j)$,得出项目相似矩阵

$$\boldsymbol{R}_{sim}=\begin{bmatrix}sim(1,1) & sim(1,2) & \cdots & sim(1,n)\\ sim(2,1) & sim(2,2) & \cdots & sim(2,n)\\ \vdots & \vdots & & \vdots\\ sim(n,1) & sim(n,2) & \cdots & sim(n,n)\end{bmatrix}。\tag{9}$$

步骤 **5** 根据初始的用户 – 项目矩阵 $\boldsymbol{R}$ 统计目标用户 u 的所有已经评价的项目的集合 I_u。

步骤 **6** 设 $I=\{i_1,i_2,\ldots,i_h\}$ 为推荐系统中所有项目的集合,计算目标用户 u 的未选项目集,即没有被该用户评价的项目组成的集合 $I_u'=I-I_u$。

步骤 **7** 根据步骤 4 得到的项目相似性矩阵 $\boldsymbol{R}_{sim}$ 中的相似性计算结果和输入的项目最近邻居个数 k,计算项目 $i(i\in I_u')$ 的 k 个最近邻居的集合 $M_i=\{i_1,i_2,\ldots,i_k\}$, $i\notin M_i$,相似度集合 $\{sim(i,i_1),sim(i,i_2),\ldots,sim(i,i_k)\}$ 以从大到小的顺序排列。如果目标用户 u 已评分的项目总数小于参数 k,则集合 M_i 只选择 $Num(I_u)$ 个最近邻居, $Num(I_u)$ 表示目标用户 u 已评分的项目总数。

步骤 **8** 根据步骤 7 得到的目标用户 u 的最近邻居集合 M_i 和用户—项目评分矩阵 $\boldsymbol{R}$ 内的评分数据,依据用户 u 对目标项目 i 的最近邻居评分的加权平均值预测用户 u 对项目 $i(i\in I_u')$ 的预测评分

$$r_{u,i}=\frac{\sum_{j\in M_i}sim(i,j)\times r_{u,j}}{\sum_{j\in M_i}|sim(i,j)|}。\tag{10}$$

式中 $r_{u,j}$ 表示用户 u 对 M_i 中的项目 j 的评分。

步骤 **9** 重复步骤 7 和步骤 8,预测目标用户 u 对所有未评分项目的评分,选择预测评分最高的项目 i 推荐给该用户;如果是 $Top-N$ 推荐,则选择评分最高的前 N 个项目推荐给用户。

2 基于项目的协同过滤推荐算法中集成语境信息

2.1 语境信息与多维推荐技术

传统的推荐技术都是基于用户×项目的二维空间,仅仅是建立在用户对项目的评价信息上来对未评分项目的评分预测,没有考虑另外的语境信息,而这些语境信息在一些应用中可能很重要。所谓语境信息就是指对人的行为或者事件的发展产生影响的上下文信息或者场景信息,如时间、地点等信息[12]。语境信息在很大程度上会影响用户的偏好和最终的购物决策。这种情况下,如果仅根据用户对项目的评分来进行推荐是不够的,作为个性化推荐的重要工具,电子商务推荐网站应该考虑这些语境信息,以满足不同用户在不同语境信息下的个性化需求。

多维技术是从数据挖掘中发展起来的一种推荐技术,它扩展了传统的二维矩阵,引入了语境信息。集成了语境信息的推荐系统的数据存在于多维空间,这些维度除表示用户和项目信息,还包括影响用户购买行为的语境信息。令 $U\times I\times D_1\times D_2\times\ldots\times D_n$ 表示推荐空间,即与评分相关的用户、项目、语境信息的集合,其中 U 表示用户空间, I 表示项目空间, $D_1,D_2,\ldots,D_n$ 分别表示 n 维语境信息的集合,每一维 D_i 的不同取值称为语境段。

相应的集成语境信息的多维评分模型为:

$$\forall(u,i,d_1,\ldots,d_n)\in U\times I\times D_1\times D_2\times\ldots\times D_n,\quad R_b=R_b(u,i,d_1,d_2,\ldots,d_n)。\tag{11}$$

式中: $u\in U$, $i\in I$, $d_1\in D_1$, $\ldots$, $d_n\in D_n$ 即用户 u 在 $d_1,d_2,\ldots,d_n$ 这些语境段下对项目 i 的评分。

不同于传统的二维推荐模型需要用全部数据进行预测,多维推荐模型在进行评分预测时只会用到与用户指定的语境信息相关的那些数据[13]。比如要推荐一部电影给一个想在周末看该电影的用户,该方法在进行评分预测时,将会只用到在周末看该电影的历史数据来进行分析。

2.2 降维方法

集成了多维语境信息的推荐算法由于存在于多维空间,除了包含用户和项目的信息,还包含影响用

户行为和决策的语境信息。这种情况下, 直接计算项目相似性和评分预测是不可能的, 需要先用其他方法来对这种包含多种数据的初始信息进行处理。

本文降维方法的主要思想是在多维推荐空间中, 将语境信息(语境段) 作为约束条件, 从而使得评分模型只包含用户对项目的评分信息的条件表达式。该方法简单可行, 类似于信息检索, 更具有可解释性。具体模型如下:

$$\begin{aligned} &\forall(u, i, d_1, \cdots, d_n) \in U \times I \times D_1 \times D_2 \times \cdots \times D_n, \\ &\quad R_b = R_b(u, i, d_{1j}, d_{2j}, \cdots, d_{nj}) \\ &\quad \Rightarrow R_b = R_b(u, i), \\ \text{s. t.} \quad &\quad d_1 = d_{ij}, \\ &\quad d_2 = d_{2j}, \\ &\quad \vdots \\ &\quad d_n = d_{nj}。 \end{aligned} \tag{12}$$

在考虑预测某一用户 u 在语境段 $d_{1j}, d_{2j}, \cdots, d_{nj}$ 下对项目 i 的评分时, 模型的主体变成了只含用户和项目两个变量的方程, 而这些相关的语境信息作为方程的约束条件而存在。

2.3　最优语境段的选择

在多维评分模型中的一个重要问题就是哪一维或几维语境信息应该被应用到, 即这些语境信息确实会对用户的购买决策产生明显的影响, 而那些对用户的行为不会或者产生很小影响的语境信息将不会在推荐中用到。最优语境段是指根据已有的用户评分信息, 如果在该语境段内用基于降维的方法比传统的基于项目的协同过滤推荐算法预测结果更加准确, 则该语境段就称为最优语境段[13]。在该语境段内用基于降维的方法进行评分预测, 否则就直接利用基于项目的协同过滤推荐算法进行预测。

最优语境段选择算法如下:

步骤 1　将用户评价记录集 D 中的项目平均分成 n 个互不相交的子集, 选择其中一个做测试集 D_t, 另外 $n-1$ 个做训练集 D_m, $D_t \cap D_m \neq \emptyset$, $D_t \cup D_m = D$。这样做是为了对数据集进行反复的交叉试验, 以使得计算结果更加准确。

步骤 2　设置一个正整数 N(N 的取值可以根据具体的需要来定), 如果 D 中属于某个语境段的评分记录大于 N, 则保留该语境段, 说明该语境段在较大程度上影响用户的评分, 否则将该语境段排除。

步骤 3　设定某一个评价推荐系统质量的评价指标为 $P_{A,X}(Y)$, 对于每一个语境段 $S_j \in S$, 计算 $P_{A,D^m_{S_j}}(D^t_{S_j})$ 以及 $P_{A,D^m}(D^t_{S_j})$, 其中: $D^m_{S_j} \in D^m$ 表示训练集 D^m 中满足语境段 S_j 的记录; $D^t_{S_j} \in D_t$ 表示测试集 D_t 中满足语境段 S_j 的记录。

步骤 4　采用 n-折交叉实验, 重复步骤 3, 计算在每个语境段 $S_j \in S$ 上两种方法的评价指标的平均值 $\overline{P_{A,D^m_{S_j}}(D^t_{S_j})}$ 和 $\overline{P_{A,D^m}(D^t_{S_j})}$, 比较两个性能评价指标, 如果 $\overline{P_{A,D^m_{S_j}}(D^t_{S_j})}$ 优于 $\overline{P_{A,D^m}(D^t_{S_j})}$, 则保留该语境段 S_j; 否则, 语境段 S_j 将不会被用到降维方法中来。设保留下来的语境段为 $S^* = \{S^*_1, S^*_2, \cdots, S^*_h\}$, 即为最优语境段的集合, 其中 $S^*_1, S^*_2, \cdots, S^*_h$ 是按照它们的优劣性顺序排列的, 即 S^*_1 是最优的语境段, S^*_2 次优, 以此类推。

2.4　多维模型下评分预测算法

在基于项目的协同过滤推荐算法中, 集成语境信息建立集成语境信息的多维评分模型, 通过选择最优的语境段, 用基于降维方法将多维的模型降低到传统的二维推荐模型上, 并在此基础上用上文提出的基于项目的协同过滤推荐的改进算法, 进行评分的预测和项目的推荐。该算法主要步骤的描述如下:

输入信息: 用户在考虑了语境信息下对项目的评分记录集、项目初始的类别矩阵 $\boldsymbol{S}$ 和项目类别调和参数 α, 以及项目最近邻居个数 k。

输出信息: 在某一语境段 S' 下, 用户 u 评分最高的项目 i 或者评分最高的 n 个项目(Top-N 推荐)。

步骤 1　根据初始的项目类别矩阵 $\boldsymbol{S}$, 建立修正的项目类别矩阵 $\boldsymbol{P}$。

步骤 2　将记录集 D 分解成训练集和测试集, 进行最优语境段的选择。该过程可以离线进行, 进行实时推荐时只需要查询语境段 S' 是否属于最优语境段的集合 S^*。

步骤 3　利用降维的方法, 预测用户 u 对未评分项目的评分。如果 $S' \in S^*$, 并且用户 u 的历史评分中存在满足语境段 S' 的记录, 则在用降维的方法建立起来的局部数据模型中, 利用改进的基于项目的协同过滤推荐算法进行评分的预测; 如果 $S' \notin S^*$, 或者该用户的历史中不存在满足语境段 S' 的记录, 则直接用改进的基于项目的协同过滤推荐算法(见 1.3 节) 进行评分的预测。

步骤 **4** 选择预测评分最高的项目 i 推荐给用户; 如果是 *Top-N* 推荐, 则选择前 N 个最高的项目列表推荐给用户 u。

3 实验仿真和测试

3.1 数据集

本文采用的数据集来自明尼苏达大学 GroupLens Research 项目组收集的 MovieLens 数据集。MovieLens 站点(http://MovieLens.umn.edu/) 用于接收用户对电影的评分, 并提供相应的电影推荐列表, 其评分尺度是从 1 到 5 的整数, 数值越高, 表明用户对该电影的偏爱程度越高。实验中用到该数据集中的 12 000 条评分数据, 包括 150 个用户和 832 部电影。在实验中还用到描述电影(项目) 的类别文件, 即每一个电影属于哪一个或哪几个类别, 共有 19 个不同的电影(项目) 类别。

3.2 评价指标

本文采用最常见的统计精度方法——平均绝对误差(Mean Absolute Error, MAE) 法作为评价推荐系统推荐效果的度量标准。

MAE 方法通过度量推荐系统产生的目标项目的预测评分与用户的实际评分之间的偏差度量推荐的准确性。算法工作在训练集上, 设预测的用户评分集合表示为$\{p_1, p_2, \cdots, p_n\}$, 对应的实际用户评分集合为$\{q_1, q_2, \cdots, q_n\}$, 则[10]:

$$MAE=\frac{\sum_{i=1}^{n}|p_i-q_i|}{N}。\tag{13}$$

MAE 值越小, 说明预测的评分和实际用户的评分相差越小, 推荐质量也就越高。

3.3 集成项目类别的协同过滤推荐算法实验

本实验采用 5-折交叉测试, 把数据集分为训练集和测试集, 训练集和测试集的比率为 4∶1, 用 MAE 法评价推荐算法质量。主要步骤如下:

步骤 **1** 对电影的类别文件进行分析, 依据 1.3 节 CACP 算法的步骤 2 和步骤 3 得到用户—项目(在本实验中项目为电影) 的项目类别矩阵 ***S***, 进而得到修正的类别矩阵 ***P***。

步骤 **2** 将项目类别考虑到相似性计算过程中, 依据 1.3 节 CACP 算法项目相似性计算公式(式(8)), 计算电影之间的相似性, 得到相似性矩阵 $\boldsymbol{R}_{sim}$。

步骤 **3** 在训练集的基础上对未评分项目(电影) 的评分进行预测, 采用的公式是 1.3 节 CACP 算法提出的集成项目类别的评分预测公式, 即式(10)。在该过程中, 选取最近邻居个数 k 为 5, 10, 15, 20, 25, 30, 35, 40 分别进行计算。为更好地比较参数 k 对推荐效果的影响, 在这一实验环节中把评分预测中的项目类别参数 α 设定为一个固定值, 即 α= 1。

步骤 **4** 计算 *MAE*。因为采用的是 5-折交叉试验的方法, 需要重复步骤 3, 计算不同的训练集和测试集上的 *MAE* 值, 求其平均值作为最终结果。

步骤 **5** 对照实验。不考虑项目类别的影响因素, 采用传统的余弦相似性(式(1))、修正的余弦相似性(式(2))、相关相似性公式(式(3)) 进行项目相似性计算, 同样采用 5-折交叉测试求其 *MAE*, 重复步骤 3 和步骤 4, 同通过 CACP 算法计算得到的 *MAE* 进行对比。

表 1 实验结果显示, 本文提出的 CACP 算法在最近邻居个数等于 5 和 10 时得出的 *MAE* 值没有余弦相似性好, 但是当 $k \geqslant 15$ 时, 均得到比传统方法更小的 *MAE* 值。这是由于在计算项目相似性时, CACP 算法考虑了项目类别的影响, 使得属于共同项目类别数目越多的项目之间的相似性越大, 这和实际情况是非常相符的。

表 **1** 不同的相似性方法得到的推荐结果

k	Cosine	Adjusted Cosine	Correlation	CACP
5	0.760 1	0.968 7	0.797 5	0.786 2
10	0.741 6	0.921 2	0.776 2	0.729 7
15	0.755 5	0.898 4	0.773 2	0.693 1
20	0.739 9	0.880 0	0.772 3	0.682 1
25	0.724 9	0.868 5	0.771 8	0.674 3
30	0.731 3	0.860 6	0.774 8	0.700 4
35	0.733 1	0.855 8	0.775 7	0.714 1
40	0.731 9	0.860 4	0.787 7	0.720 1

3.4 参数灵敏度分析

在 CACP 算法中, 项目类别调和参数 α 和最近邻居个数 k 是非常重要的两个参数, 其取值直接影响到推荐质量。此外, 用户—项目评分数据也对预测结果产生直接的影响。为了对改进算法做进一步的验证和分析, 分别对参数 k、参数 α, 以及评分数据

r_{ij} 在 CACP 算法中对推荐质量的影响进行灵敏度分析。

(1) 最近邻居个数 k 灵敏度分析

固定项目类别参数 α, 取 α= 0. 8, k 的值分别为 10, 15, 20, 25, 30, 35, 40, 50, 60, 70。根据 1. 3 节 CACP 算法步骤, 用 MAE 作为推荐质量度量标准, 实验结果如图 1 所示。

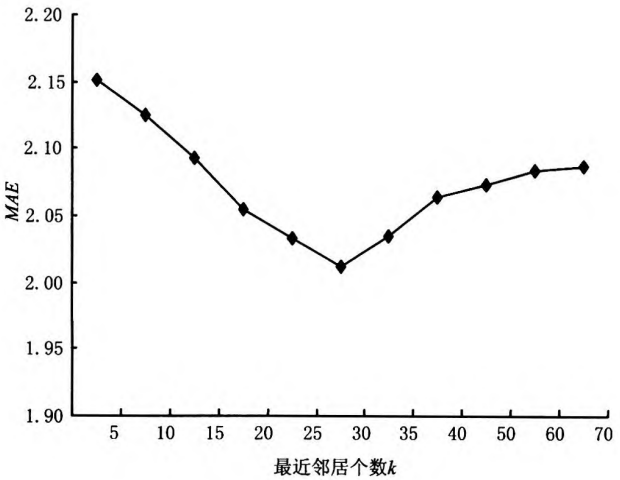

图1　最近邻居个数灵敏度分析

实验结果显示: k= 30时, 算法产生的推荐效果最好。当最近邻居个数 k 从 5 增加到 30 时, MAE 的值逐渐减小, 且下降的速度比较快, k > 30 时, MAE 趋于缓慢上升趋势。这说明在数据集较少的情况下, 增加最近邻居个数并不总能提高推荐精度, 当 k 增加到一定程度时, 反而会使 MAE 的值增加, 因为增加最近邻居个数会使得在评分预测时, 会使用与目标项目相似性并不大的项目, 反而会降低对目标项目的预测的准确性; 反之, 当最近邻居个数取得较少时(如 k= 10), 由于不能够充分利用已有的用户评分信息, 同样不能达到较好的预测效果。

(2) 项目类别调和参数 α 灵敏度分析

固定最近邻居个数 k 为 30, α 的值分别为 0. 2, 0. 4, 0. 6, 0. 8, 1. 0, 1. 2, 1. 4, 1. 6, 1. 8, 2. 0, 4. 0, 6. 0, 依据 1. 3 节 CACP 算法计算得到的 MAE 值如图 2 所示。

从图 2 中可以看到, 当 α< 0. 8 时, MAE 随着 α 值的增加而减少; 当 α= 0. 8 时, MAE 值最小, 推荐效果最佳; 当 α> 0. 8 时, 推荐精度又随 α 值的增加而有所降低。可见, 如果 α 取值过小, 在相似性计算时就不太能体现加入项目类别因素带来的影响, 在评分预测阶段使得预测的评分不够准确; 如果 α 的值取得过大, 在相似性计算时, 会使项目类别因素影响过大, 已有的用户评分信息影响过小, 同样会导致最终预测评分的不合理。

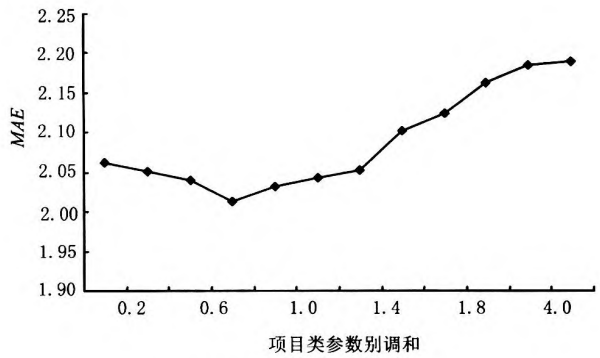

图2　项目类别调和参数灵敏度分析

(3) 用户—项目评分矩阵评分数据 r_{ij} 灵敏度分析

作为算法的基础数据, 评分矩阵的评分数据 r_{ij} 的变化自然会引起推荐结果的变化, 仅以用户 4 为例。在原训练集中, 用户 4 对项目 11 的评分 $r_{4,11}$= 4, 对项目 210 的评分 $r_{4,210}$= 3, 对项目 258 的评分为 $r_{4,258}$= 5, 根据推荐算法求得用户 4 对未知项目 50 的评分预测 $r_{4,50}$ = 3. 900 5。现将 $r_{4,11}$, $r_{4,210}$ 和 $r_{4,258}$ 的取值分别在 $[1, 5]$ 范围内变换, 其中一项变化时, 另外两项保持原值不变。得到的实验结果如表 2 所示。

表 **2**　评分数据参数灵敏度分析

$r_{4,11}$	$r_{4,50}$	$r_{4,210}$	$r_{4,50}$	$r_{4,258}$	$r_{4,50}$
1	3. 677 8	1	3. 413 5	1	3. 296 2
2	3. 752 9	2	3. 658 6	2	3. 449 6
3	3. 827 1	3	3. 900 5	3	3. 601 5
4	3. 900 5	4	4. 139 4	4	3. 751 8
5	3. 973 1	5	4. 375 2	5	3. 900 5

从表 2 中数据可以看出, 项目 11、项目 210、项目 258 同项目 50 都是正相关关系, 随着用户 4 对项目 11、项目 210、项目 258 评分的提高, 根据 CACP 算法得到的用户 4 对项目 50 的预测评分也随之提高。其中, 随项目 11 的提升幅度最大, 这是由于项目 210 与项目 50 的相似度最高, 以项目相似度作为评分预测的权重(式(10)), 用户 4 关于项目 210 的评分对预测结果的影响最大。此外, 用户 4 对未知项目 50 的评分随单个评分数据变化波动相对较大,

这是因为该实验数据集中,用户 4 的评分数据很少。对于评分数据较多的用户,其对未知项目的评分预测受单个值的影响相对较小,特别是当该项目不在未知项目最近邻居项目集中时。

4 结束语

协同过滤方法已经成功应用于电子商务推荐系统,但随着电子商务推荐系统规模的不断扩大,协同过滤算法也暴露出很多问题。本文提出了改进的基于项目的协同过滤推荐算法,首先在计算项目相似性时考虑到项目类别因素的影响,使得属于共同项目类别数目越多的两个项目之间的相似性越大,这一方面是符合项目之间的本身内在联系,另一方面也减少了数据稀疏性的影响。在基于项目的协同过滤推荐算法中集成语境信息,将多维语境信息作为预测评分的重要因素,通过降维方法将多维评价模型降低到传统的用户—项目二维模型上来。

参考文献:

[1] YU Li, LIU Lu. Research on personalized recommendations in e-business[J]. Computer Integrated Manufacturing Systems, 2004, 10(10): 1306-1313(in Chinese). [余 力, 刘 鲁. 电子商务个性化研究[J]. 计算机集成制造系统, 2004, 10(10): 1306-1313.]

[2] ZENG Chun, XING Chunxiao, ZHOU Lizhu. A survey of personalization technology[J]. Journal of Software, 2002, 13(10): 1952-1961(in Chinese). [曾 春, 邢春晓, 周立柱. 个性化服务技术综述[J]. 软件学报, 2002, 13(10): 1952-1961.]

[3] ZHOU Junfeng, TANG Xian, GUO Jingfeng. An optimized collaborative filtering recommendation algorithm[J]. Journal of Computer Research and Development, 2004, 41(8): 1842-1847(in Chinese). [周军锋, 汤 显, 郭景峰. 一种优化的协同过滤推荐算法[J]. 计算机研究与发展, 2004, 41(8): 1842-1847.]

[4] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International World Wide Web Conference. New York, N.Y., USA: ACM, 2001: 285-295.

[5] HUANG Z, CHEN H, ZENG D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering[J]. ACM Transactions on Information Systems, 2004, 22(1): 116-142.

[6] BRECHEISEN S, KRIEGEL H, PFEIFLE M. Efficient density-based clustering of complex objects[C]//Proceedings of the 4th IEEE International Conference on Data Mining. Washington, D.C., USA: IEEE, 2004: 43-50.

[7] NIE Kai. A new user-based collaborative filtering recommendation algorithm[J]. Logistics Sci-Tech, 2006(9): 123-127(in Chinese). [聂 凯. 一种新的基于用户的协作过滤推荐算法[J]. 物流科技, 2006(9): 123-127.]

[8] DESHPANDE L M, KARYPIS G. Item-based Top-N recommendation algorithms[J]. ACM Transactions on Information Systems, 2004, 22(1): 143-177.

[9] KARYPIS G. Evaluation of item-based Top-N recommendation algorithms[C]//Proceedings of the 10th International Conference on Information and Knowledge Management. New York, N.Y., USA: ACM, 2001: 247-254.

[10] ZHANG Guangwei, KANG Jianchu. Context based collaborative filtering recommendation algorithm[J]. Journal of System Simulation, 2006, 18(S1): 595-602(in Chinese). [张光卫, 康建初. 面向场景的协同过滤推荐算法[J]. 系统仿真学报, 2006, 18(S1): 595-602.]

[11] DENG Ailin, ZHU Yangyong, SHI Bole. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Journal of Software, 2003, 14(9): 1621-1628(in Chinese). [邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报, 2003, 14(9): 1621-1628.]

[12] CHO S, LEE M, JANG C, et al. Multidimensional filtering approach based on contextual information[C]//Proceedings of International Conference on Hybrid Information Technology. Los Alamitos, Cal., USA: IEEE Computer Society, 2006: 497-504.

[13] ADOMAVICIUS G, SANKARANARAYANAN R, SEN S, et al. Incorporating contextual information in recommender systems using a multidimensional approach[J]. ACM Transactions on Information Systems, 2005, 23(1): 103-145.

[14] HOFMANN T. Latent semantic models for collaborative filtering[J]. ACM Transactions on Information Systems, 2004, 22(1): 213-238.