

一种改进的结合标签和评分的协同过滤推荐算法

高娜, 杨明

(南京师范大学计算机科学与技术学院, 江苏 南京 210023)

[摘要] 推荐系统由于其数据量庞大的原因, 已经成为大数据领域研究的一个热点. 而协同过滤算法是推荐系统中最著名的算法之一. 传统协同过滤算法在利用评分矩阵进行推荐时, 面临数据稀疏性问题, 从而严重影响推荐的质量. 同时, 推荐系统中存在大量的描述用户和产品属性特征的标签信息, 把这些标签信息融入到传统的推荐算法中是解决稀疏性的一个有效方法. 因此, 针对稀疏性问题, 本文提出了一种结合标签和评分的协同过滤推荐算法. 该算法结合标签信息和评分数据共同计算用户之间或产品之间的相似性, 进而为用户产生推荐. 实验结果表明, 本文提出的算法可以有效解决数据稀疏性问题, 同时可以提高推荐系统的准确性.

[关键词] 协同过滤, 标签, 推荐系统, 稀疏性

[中图分类号] TP391 **[文献标志码]** A **[文章编号]** 1001-4616(2015)01-0098-06

An Improved Unifying Tags and Ratings Collaborative Filtering for Recommendation System

Gao Na, Yang Ming

(School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China)

Abstract: The recommendation system has become the hot topic widely studied in the field of big data due to the massive amounts of data it contains. While the collaborative filtering algorithm is one of the most popular approach in the recommendation system. When making recommendations using the traditional collaborative filtering (CF) algorithms based on ratings matrix, we face the problem of sparsity that seriously impairs the quality of recommendation. Meanwhile, there is a large number of tags information that describe the attribute characteristics of users and items. Integrating these tags information into the traditional recommendation algorithms is a promising means to alleviate the sparsity problem. Therefore, to address the sparsity problem, this paper proposes a new collaborative filtering recommendation algorithm that integrates the tags and ratings, named UTR-CF. This algorithm utilizes the tags information and the ratings data simultaneously to compute the similarity between users or items, and then generate the recommendations. The experimental results indicate that the newly developed algorithm can alleviate the sparsity problem, and improve the accuracy of recommendation system simultaneously.

Key words: Collaborative Filtering, tag, recommendation system, sparsity

近年来信息技术的飞速发展和互联网的迅速普及使用户面临数据过载问题, 已严重制约了电子商务的发展. 从海量数据中获取用户真正感兴趣的信息如同大海捞针. 因此, 如何帮助用户快速找到他们感兴趣的产品或服务, 为用户提供个性化推荐, 成为当今社会关注的焦点和亟待解决的问题. 为此, 研究人员开发了推荐系统(Recommendation System)来解决数据过载这一问题. 推荐系统为不同的用户提供不同的服务, 节省了用户选择产品的时间, 同时也为电子商务提供商寻找自己的客户提供了渠道, 实现了用户和电子商务提供商的双赢.

协同过滤推荐算法(Collaborative Filtering, CF)^[1]是推荐系统中应用最成功的一种推荐算法. 该算法通过挖掘用户的历史行为来建立用户兴趣模型, 进而为用户产生推荐. 其核心思想是搜寻与目标用户有相似兴趣爱好的邻居用户, 根据邻居用户的信息来产生推荐. 此类算法的优势是不需要建立配置文件, 能处理一些难以进行文本描述的复杂对象, 如视频、音乐等. 但其局限性也是很明显的, 即存在稀疏性、可扩

收稿日期: 2014-09-30.

基金项目: 国家自然科学基金重点、面上(61432008、61272222).

通讯联系人: 高娜, 硕士, 研究方向: 机器学习、模式识别. E-mail: gaonahao@126.com

展性和冷启动等问题. 根据推荐方式的不同,协同过滤推荐算法可分为两大类算法:Memory-based 算法^[2]和 Model-based 算法^[3].

Memory-based 算法又被称为基于邻域的推荐算法(Neighborhood-based approach),因为它是基于邻域信息进行推荐的. 根据推荐方式的不同,它又可分为基于用户的协同过滤算法(User-based Collaborative Filtering, UserCF)^[1]和基于产品的协同过滤算法(Item-based Collaborative Filtering, ItemCF)^[4]. 二者的核心分别是计算用户之间和产品之间的相似度. 因此,当评分矩阵十分稀疏时,计算得到的相似度不准确,严重影响推荐的质量.

Model-based 算法首先从用户-产品评分矩阵中获取用户的行为模型,进而预测用户对目标产品的评分. 此类算法主要包括矩阵分解^[5]、支持向量机^[6]、决策树^[7]等推荐算法. 矩阵分解的核心思想是把一个高维矩阵分解为 2 个低维矩阵的乘积形式,进而用 2 个低维矩阵的乘积来预测未知评分.

可以看出,基于协同过滤的推荐系统面临严重的评分数据稀疏性问题. 已有学者提出一些方法来解决这一问题,例如, Sarwar B M 等人在文献[8]中提出一种降维的方法来缓解稀疏性; Wang Z 提出结合基于标签的邻域算法和基于评分的邻域算法^[10]来解决稀疏性问题,文中的标签是网上用户为了表达自己的偏好为产品创建的标签. 针对稀疏性问题,本文在文献[10]的基础上提出一种新的结合标签和评分的协同过滤算法:UTR-CF. 与文献[9]不同的是,本文中的标签是领域专家预先设定的,且标签与评分的结合方式采用联合学习的方式. UTR-CF 算法结合标签信息和评分数据共同计算相似性,弥补了已有的仅利用评分矩阵计算相似性的不足. 实验结果表明,本文提出的方法可有效缓解数据稀疏性问题.

1 已有工作

本文指定特殊的符号来区别用户和产品:用户使用符号 u 和 v 表示;产品使用符号 i 和 j 表示. r_{ui} 表示用户 u 对产品 i 的真实评分, \hat{r}_{ui} 表示用户 u 对产品 i 的预测评分. 评分数据是 1 到 5 的整数,反映了用户对产品偏爱程度的不同.

从不同的角度划分,Memory-based 算法主要包含两大类算法:基于用户的协同过滤算法(UserCF)和基于产品的协同过滤算法(ItemCF). 二者分别需要找出用户的邻居和产品的邻居,并根据其邻居的评分来为用户产生一组推荐.

1.1 基于用户的协同过滤算法

UserCF 算法依据的思想是一个用户会喜欢和他有相似兴趣爱好的好友所喜欢的东西. 其关键是搜寻与目标用户兴趣爱好相似的邻居用户集,进而参考邻居用户集的评分来预测目标用户的未知评分. 该算法的主要步骤为:

- (1) 计算目标用户 u 与其他所有用户 v 之间的相似度;
- (2) 选择与目标用户 u 相似度最大的前 k 个用户作为其邻居集;
- (3) 目标用户 u 对 i 的预测评分就表示为用户 u 的邻居集对 i 的评分的加权和. 其中,权重即为步骤(1)所得到的相似度.

步骤(1)中用户之间相似度的计算是在用户-产品评分矩阵上进行的,是基于共同评分的产品集进行的,多数情况下采用皮尔逊相关系数^[4]来计算相似度.

若 I 表示同时被用户 u 和用户 v 评过分的 product 集, r_{ui} 和 r_{vi} 分别表示用户 u 和用户 v 对产品 i 的真实评分, \bar{r}_u 和 \bar{r}_v 分别表示用户 u 和用户 v 在产品集 I 上的平均评分,则用皮尔逊相关系数^[4]计算用户 u 和用户 v 之间的相似度公式为:

$$\text{sim}(u, v) = \frac{\sum_{i \in I} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{vi} - \bar{r}_v)^2}}. \quad (1)$$

与目标用户 u 相似度最大的前 k 个用户构成它的邻居集,表示为 $S^k(u; i)$. 则利用 UserCF 算法预测用户 u 对产品 i 的评分公式为:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in S^k(u; i)} \text{sim}(u, v)(r_{vi} - \bar{r}_v)}{\sum_{v \in S^k(u; i)} \text{sim}(u, v)}. \quad (2)$$

式(2)中的 $\text{sim}(u, v)$ 即为式(1)计算得到的相似度. 引入用户 u 和 v 的平均评分 \bar{r}_u 和 \bar{r}_v 是为了消除不同用户评分尺度的不同.

1.2 基于产品的协同过滤算法

ItemCF 算法认为, 用户对一个产品的评分与用户对该产品的相似产品的评分接近, 其核心是计算产品之间的相似性. ItemCF 算法的主要步骤为:

- (1) 计算目标产品 i 与其他所有产品 j 之间的相似度;
- (2) 选择与目标产品 i 相似度最大的前 k 个产品作为其邻居集;
- (3) 用户 u 对目标产品 i 的预测评分就表示为用户 u 对产品 i 邻居集的评分的加权. 其中, 权重即为步骤(1)所得到的相似度.

产品之间相似度的计算是基于共同评分的用户集进行的, 如图 1 所示. 图 1 给出了一个计算产品之间相似度的例子, 行表示用户, 列表示产品, 交叉点表示评分值. 图中产品 i 和产品 j 之间的相似度是基于共同评分的用户集 $(1, u, m)$ 进行计算的.

若 U 表示对产品 i 和 j 都有评分的用户集, \bar{r}_i 和 \bar{r}_j 分别表示产品 i 和 j 在 U 上的平均评分, r_{ui} 和 r_{uj} 分别表示用户 u 对 i 和 j 的真实评分, 则采用皮尔逊相关系数计算产品 i 和 j 之间相似度的公式为:

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{uj} - \bar{r}_j)^2}}. \quad (3)$$

与目标产品 i 相似度最大的前 k 个产品构成了它的邻居集, 表示为 $S^k(i; u)$. 则利用 ItemCF 算法预测用户 u 对产品 i 的评分的公式为:

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in S^k(i; u)} \text{sim}(i, j)(r_{uj} - \bar{r}_j)}{\sum_{j \in S^k(i; u)} \text{sim}(i, j)}. \quad (4)$$

引入产品 i 和 j 的平均评分 \bar{r}_i 和 \bar{r}_j 是为了消除不同产品评分尺度的不同.

ItemCF 算法的优点是可扩展性强. 由于推荐系统中的产品集相对稳定, 因此可以提前离线计算出产品的相似度矩阵, 用 ItemCF 算法进行推荐时直接调用相似度矩阵, 节省了大量的推荐时间, 能应付产品数量大幅增长的情况. 而对于 UserCF 算法, 由于推荐系统中的用户数量不稳定, 故不能提前计算出相似度. 因此, 当用户数目大量增长时, UserCF 算法需要实时计算用户之间的相似度, 其可扩展性比较弱.

1.3 结合 UserCF 算法和 ItemCF 算法的集成算法

有学者提出结合 UserCF 算法和 ItemCF 算法的集成算法来进行推荐^[10,11], 它综合了两种协同过滤算法的优点, 取得了很好的效果. 文献[10]对这种集成算法进行了详细描述, 即结合式(2)和式(4), 引入一个平衡因子 α 来调节二者所占的比重, 记为 HybridCF 模型^[10]:

$$\hat{r}_{ui} = \alpha \times \left(\bar{r}_u + \frac{\sum_{v \in S^k(u; i)} \text{sim}(u, v) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in S^k(u; i)} \text{sim}(u, v)} \right) + (1 - \alpha) \times \left(\bar{r}_i + \frac{\sum_{j \in S^k(i; u)} \text{sim}(i, j) \times (r_{uj} - \bar{r}_j)}{\sum_{j \in S^k(i; u)} \text{sim}(i, j)} \right). \quad (5)$$

2 改进的协同过滤算法(UTR-CF)

推荐系统中的信息不仅有评分数据, 还有用户和产品的标签信息. 标签是一种描述事物多重属性的分类工具. 本文实验用到的数据集是 MovieLens 数据集^[12], 产品即数据集中的电影. 电影的标签为电影的类型, 用户的标签有用户的年龄、性别和职业等. 这些标签都是领域专家预先为其设定的. 已有的协同过滤算法都是基于评分矩阵进行推荐的, 因此存在稀疏性问题. 本文针对此问题, 考虑加入用户和电影的标签信息, 提出一种结合标签和评分共同计算相似度的改进的协同过滤算法: UTR-CF 算法.

	1	2		i	j		$n-1$	n
1				R	R			
u				R	R			
$m-1$				R	-			
m				R	R			

图 1 产品相似度计算示例

Fig. 1 A demonstration of items similarity calculation

2.1 利用标签计算相似度

每个产品和每个用户都有一个标签集来描述它的特征属性. 例如,电影推荐系统中一个电影的标签集是 {Action, Children, Comedy}, 它描述了这个电影所属的类型; 而一个用户的标签集 {M, 24, 'educator'} 则描述了这个用户的性别、年龄和职业等人口统计学特征. 那么, 计算用户之间或电影之间的相似度就转化为计算它们所对应的标签集之间的相似度. 在计算标签集相似度之前, 首先要把标签集等文本信息转化成数字信息以方便建模.

假设两个用户 (或两个电影) 的标签集转化成数字信息后分别表示为 m 维空间中的两个向量: $\mathbf{t} = (t_1, t_2, \dots, t_m)$, $\mathbf{s} = (s_1, s_2, \dots, s_m)$, 称之为标签向量. 本文使用余弦相似性方法来计算标签向量之间的相似性:

$$\text{sim}(\mathbf{t}, \mathbf{s}) = \cos(\mathbf{t}, \mathbf{s}) = \frac{\mathbf{t} \cdot \mathbf{s}}{\|\mathbf{t}\|_2 \cdot \|\mathbf{s}\|_2} = \frac{\sum_{k=1}^m t_k s_k}{\sqrt{\sum_{k=1}^m t_k^2} \sqrt{\sum_{k=1}^m s_k^2}}, \quad (6)$$

式中, \cdot 表示两个向量之间的内积. 在余弦相似性度量方法中, 余弦值越大, 向量夹角就越小, 则用户或产品之间的相似程度就越高.

2.2 结合标签和评分共同计算相似度

以计算产品 i 和 j 之间的相似度为例, 本文提出的 UTR-CF 算法的主要思路为:

(1) 基于评分矩阵利用式(3)计算产品 i 和 j 之间的相似度, 表示为 $\text{simR}(i, j)$;

(2) 基于标签信息利用式(6)计算产品 i 和 j 之间的相似度, 表示为 $\text{simT}(i, j)$;

(3) 既考虑利用评分矩阵得到的相似度 $\text{simR}(i, j)$, 又考虑利用标签信息得到的相似度 $\text{simT}(i, j)$, 在总的相似度 $\text{uni_sim}(i, j)$ 中, 二者所占的比重通过参数 λ 来控制:

$$\text{uni_sim}(i, j) = \lambda \text{simR}(i, j) + (1 - \lambda) \text{simT}(i, j). \quad (7)$$

类似地, 用户 u 和 v 之间总的相似度 $\text{uni_sim}(u, v)$ 也可通过式(7)计算得到.

将式(7)应用到模型(2)、(4)、(5)中, 即用总的相似度 $\text{uni_sim}(u, v)$ 和 $\text{uni_sim}(i, j)$ 替换原来的仅利用评分矩阵计算得到的相似度 $\text{sim}(u, v)$ 和 $\text{sim}(i, j)$, 可得到如下 3 个改进的公式:

(1) 基于用户的协同过滤算法 (记为 UTR-UserCF):

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in S^k(u; i)} \text{uni_sim}(u, v) (r_{vi} - \bar{r}_v)}{\sum_{v \in S^k(u; i)} \text{uni_sim}(u, v)}; \quad (8)$$

(2) 基于产品的协同过滤算法 (记为 UTR-ItemCF):

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in S^k(i; u)} \text{uni_sim}(i, j) (r_{uj} - \bar{r}_j)}{\sum_{j \in S^k(i; u)} \text{uni_sim}(i, j)}; \quad (9)$$

(3) 集成基于用户的协同过滤算法和基于产品的协同过滤算法 (记为 UTR-Hybrid CF):

$$\hat{r}_{ui} = \alpha \times \left(\bar{r}_u + \frac{\sum_{v \in S^k(u; i)} \text{uni_sim}(u, v) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in S^k(u; i)} \text{uni_sim}(u, v)} \right) + (1 - \alpha) \times \left(\bar{r}_i + \frac{\sum_{j \in S^k(i; u)} \text{uni_sim}(i, j) \times (r_{uj} - \bar{r}_j)}{\sum_{j \in S^k(i; u)} \text{uni_sim}(i, j)} \right). \quad (10)$$

实验结果表明, 本文提出 UTR-CF 推荐算法可显著提高推荐系统的准确性, 降低推荐的误差.

3 实验结果及分析

3.1 数据集

本文使用美国 GroupLens 研究项目组提供的 MovieLens 数据集^[12]. 数据集中共有 943 个用户, 1 682 部电影, 10 万个评分. 本文从数据集中抽取 80% 作为训练集构建模型, 20% 作为测试集检测算法的效果, 使用 5 折交叉平均实验结果来减少误差.

3.2 度量标准

平均绝对误差 (Mean Absolute Error, MAE) 是广泛应用于评价推荐系统推荐质量的标准, MAE 越小, 推荐系统的精确度越高. MAE 定义如下^[10]:

$$\text{MAE} = \frac{\sum_{u, i} |\hat{r}_{ui} - r_{ui}|}{N}, \quad (11)$$

式中, \hat{r}_{ui} 表示用户 u 对产品 i 的预测评分; r_{ui} 表示用户 u 对 i 的真实评分; N 为测试的总数量. 公式所表述含义为所有预测误差的平均值.

3.3 实验结果分析

本文设置邻居数量(又称邻域大小) k 的变化范围为 20~30, 分别比较 3 种协同过滤算法 UserCF、ItemCF 和 Hybrid CF 在加入标签与未加入标签时的性能.

3.3.1 ItemCF 算法与 UTR-ItemCF 算法

结合标签与评分共同计算相似度时, 需要调节式(7)中的参数 λ . 对于 ItemCF, 实验得到的参数 λ 的最优值为 0.4, 即利用评分矩阵计算得到的相似度所占比重为 0.4, 利用标签信息计算得到的相似度所占比重为 0.6. 因此, 固定参数 $\lambda=0.4$, 使邻域大小 k 从 20 变化到 30, 比较未加入标签信息的传统 ItemCF 算法与加入标签信息的 UTR-ItemCF 算法的性能. 表 1 给出了这两种算法在不同 k 值情况下的详细 MAE 值.

从表 1 可以看出, 随着 k 值的增加, improvement 的值有所下降, 即 UTR-ItemCF 算法相对于 ItemCF 算法性能提高的幅度有所下降. 这是因为当推荐所参考的邻居数越多时, 用 ItemCF 算法计算得到的推荐结果就越准确, 加入标签信息对其改善程度也就变得越微弱.

图 2 为两种算法的 MAE 值随邻域大小 k 的变化趋势图. 从图 2 可以看出, 本文提出的 UTR-ItemCF 算法的性能明显好于传统 ItemCF 算法的性能, 且随着 k 值的增加, 二者的 MAE 值都呈下降趋势.

3.3.2 UserCF 算法与 UTR-UserCF 算法

对于 UTR-UserCF 算法, 实验得到式(7)中参数 λ 的最优值为 0.7, 即利用评分矩阵和标签信息计算得到的相似度所占比重分别为 0.7 和 0.3. 因此, 固定参数 $\lambda=0.7$, 使邻域大小 k 从 20 变化到 30, 观察 UserCF 算法与 UTR-UserCF 算法的 MAE 值随邻域大小 k 的变化情况, 表 2 给出了两种算法在不同 k 值下的详细 MAE 值. 为清晰观察其间的关系, 将表 2 中的数据用折线图的形式呈现, 如图 3 所示.

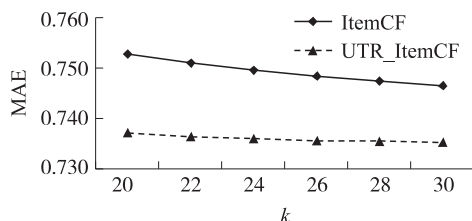


图 2 ItemCF 算法与 UTR-ItemCF 算法的 MAE 随邻域大小 k 的变化情况

Fig. 2 The MAE values of ItemCF algorithm and UTR-ItemCF algorithm with the change of neighborhood size k

由图 3 可知, 本文提出的 UTR-UserCF 算法的性能明显优于传统 UserCF 算法的性能, 且二者的 MAE 值随 k 的增加都呈下降趋势. 说明在进行推荐时所参考的邻居数目越多, 得到的推荐结果越准确.

3.3.3 Hybrid CF 算法与 UTR-Hybrid CF 算法

对于 Hybrid CF 算法, 调节式(5)中的参数 α , 得到 α 的最优值为 0.4. 固定 UTR-ItemCF 算法中的参数 $\lambda=0.4$, 固定 UTR-UserCF 算法中的参数 $\lambda=0.7$, 比较未加入标签信息的 Hybrid CF 算法与加入标签信息的 UTR-Hybrid CF 算法的性能, 表 3 列出了两种算法的详细 MAE 值随邻域大小 k 的变化情况. 从图 4 可以看出, 本文提出的 UTR-Hybrid CF 算法明显优于传统的 Hybrid CF 算法, 且 UTR-

表 1 ItemCF 算法与 UTR-ItemCF 算法的 MAE 值

Table 1 The MAE value of ItemCF algorithm and UTR-ItemCF algorithm

k	ItemCF	UTR-ItemCF	Improvement
20	0.752 8	0.737 2	0.015 6
22	0.751 0	0.736 4	0.014 6
24	0.749 6	0.736 0	0.013 6
26	0.748 4	0.735 6	0.012 8
28	0.747 4	0.735 5	0.012 0
30	0.746 5	0.735 3	0.011 2

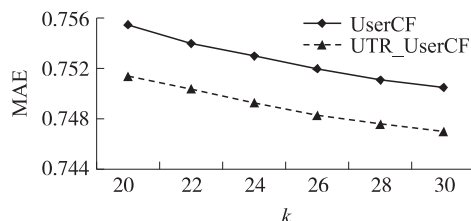


图 3 UserCF 算法与 UTR-UserCF 算法的 MAE 随邻域大小 k 的变化情况

Fig. 3 The MAE values of UserCF algorithm and UTR-UserCF algorithm with the change of neighborhood size k

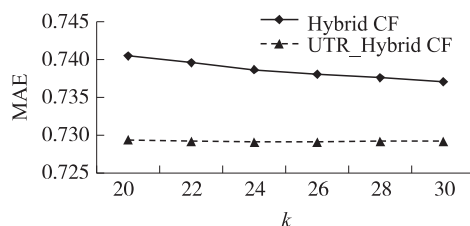


图 4 Hybrid CF 算法与 UTR-Hybrid CF 算法的 MAE 随邻域大小 k 的变化趋势

Fig. 4 The MAE values of Hybrid CF algorithm and UTR-Hybrid CF algorithm with the change of neighborhood size k

Hybrid CF 算法的 MAE 值随 k 的变化趋势不明显,趋于平稳.

从以上实验结果可以看出,本文提出的结合标签和评分共同计算相似度的协同过滤算法(UTR-CF)明显好于传统的仅利用评分计算相似度的协同过滤算法.

表 2 UserCF 算法与 UTR-UserCF 算法的 MAE 值

Table 2 The MAE values of UserCF algorithm and UTR-UserCF algorithm

k	UserCF	UTR-UserCF	Improvement
20	0.755 5	0.751 4	0.004 1
22	0.754 0	0.750 4	0.003 6
24	0.753 0	0.749 3	0.003 7
26	0.752 0	0.748 3	0.003 7
28	0.751 1	0.747 6	0.003 5
30	0.750 5	0.747 0	0.003 5

表 3 Hybrid CF 算法与 UTR-Hybrid CF 算法的 MAE

Table 3 The MAE values of Hybrid CF algorithm and UTR-Hybrid CF algorithm

k	Hybrid CF	UTR-Hybrid CF	Improvement
20	0.740 4	0.729 3	0.011 1
22	0.739 5	0.729 1	0.010 4
24	0.738 6	0.729 0	0.009 6
26	0.738 0	0.729 0	0.009 0
28	0.737 5	0.729 1	0.008 4
30	0.737 0	0.729 1	0.007 9

4 结语

目前,协同过滤算法已取得很大进展,但仍存在稀疏性、冷启动、可扩展性等问题.本文针对数据稀疏性问题提出融合标签和评分的协同过滤算法,即 UTR-CF 算法.实验结果表明,本文提出的算法可以缓解稀疏性问题并提高推荐系统的准确性.由于推荐系统中数据量庞大,系统的可扩展性问题尤为突出,因而把大数据领域知识和云计算技术应用到推荐系统中来减少推荐时间将是一个很好的选择.因此,今后的工作将致力于把协同过滤算法迁移部署到云计算平台中来改善推荐系统的实时性.

[参考文献]

- [1] Resnick P, Iacovou N, Suchak M, et al. Grouplens: an open architecture for collaborative filtering of netnews[C]//Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. New York, USA: Chapel Hill Press, 1994: 175-186.
- [2] Delgado J, Ishii N. Memory-based weighted-majority prediction for recommender systems[C]//Proceedings of ACM SIGIR'99 Workshop on Recommender Systems. UC, USA: Berkeley Press, 1999: 251-257.
- [3] Marlin B. Modeling user rating profiles for collaborative filtering[C]//Advances in Neural Information Processing Systems. Toronto, Canada: MIT Press, 2003.
- [4] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International Conference on World Wide Web. New York: ACM, 2001: 285-295.
- [5] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [6] Xu J A, Araki K. A SVM-based personal recommendation system for TV programs[C]//Proceedings of Conference on Multi-Media Modeling. Beijing: IEEE, 2006: 4.
- [7] Nikovski D, Kulev V. Induction of compact decision trees for personalized recommendation[C]//Proceedings of the 2006 ACM Symposium on Applied Computing. Dijon, France: ACM, 2006: 575-581.
- [8] Sarwar B, Karypis G, Konstan J A, et al. Application of dimensionality reduction in recommender system-a case study[R]. Minneapolis: Minnesota Univ Minneapolis Dept of Computer Science, 2000.
- [9] Wang Z, Wang Y, Wu H. Tags meet ratings: improving collaborative filtering with tag-based neighborhood method[C]//Proceedings of the Workshop on Social Recommender Systems. Hongkong, China: ACM, 2010.
- [10] Ji H, Li J, Ren C, et al. Hybrid collaborative filtering model for improved recommendation[C]//2013 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). Dongguan: IEEE, 2013: 142-145.
- [11] Wang J, de Vries A P, Reinders M J T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion[C]//Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Washington, USA: ACM, 2006: 501-508.
- [12] Riedl J, Konstan J. Movielens dataset[EB/OL]. (1998-10-19) [2008-07-18] <http://www.grouplens.org/data/>.

[责任编辑:黄 敏]