

基于贝叶斯理论的协同过滤推荐算法

孟宪福, 陈 莉

(大连理工大学 电子与信息工程学院, 辽宁 大连 116024)

(goodchenli2004@163.com)

摘 要:考虑到在协同过滤算法中邻居集合的有效性是影响推荐质量的重要因素,提出了基于贝叶斯理论的协同过滤推荐方法,该方法利用贝叶斯理论分析用户对项目特征值的喜好度。在计算相似度时,考虑用户喜好度,在此基础上计算目标项目的最近邻居。实验结果表明该算法可以提高推荐系统的推荐质量。

关键词:贝叶斯理论;用户喜好度;协同过滤;项目特征;相似度度量

中图分类号: TP18; TP393 **文献标志码:** A

Collaborative filtering recommendation algorithm based on Bayesian theory

MENG Xian-fu, CHEN Li

(School of Electronic and Information Engineering, Dalian University of Technology, Dalian Liaoning 116024, China)

Abstract: Considering that the effectiveness of the collection of neighbors is an important factor to influence the quality of the recommendation in collaborative filtering recommendation algorithm, a collaborative filtering recommendation method based on Bayesian theory was proposed in this paper. It got the value of the users' preference for a certain characteristic by using Bayesian theory. In the calculation of the similarity degree, it considered users' preferences. And then it calculated the collection of the neighbors. The result shows that it can provide better recommendation quality than traditional item-algorithm.

Key words: Bayesian theory; preference value; collaborative filtering; item characteristic; similarity measurement

0 引言

协同过滤算法的基本思想是通过参考与目标项目相似度高邻居集合的评分来预测目标项目的评分,从而产生最终的推荐。随着站点结构变化,内容复杂度增加和用户增多,基于协同过滤技术的系统面临以下问题:如何提高算法的可扩展性以及如何提高协同过滤算法的推荐质量。研究人员提出了各种解决方法,文献[1]作者提出了基于时间的数据权重和基于资源相似度的数据权重两种改进度量,此方法更好地反应用户兴趣的变化,从而提高推荐质量。文献[2]作者提出以用户的评价个数和为他人提供推荐的次数作为要素的信任模型,并将信任因子引入到协同过滤算法中。文献[3]作者提出了通过减少项目空间的维数,使用户在降维后的项目空间上对每个项目都有评分来解决数据的稀疏性问题,以此提高推荐质量,但是降维会导致信息损失。文献[4]作者提出基于 K-means 聚类算法的近邻预选择算法,对用户的评分相似性进行聚类,在一定程度上提高了实时响应速度。其他的技术如 Bayesian 网络技术^[5]、Hortig 图技术^[6]、关联规则技术^[7]也被用来改进协同过滤技术中的某些缺陷。

考虑到构造有效的邻居集合是提高推荐质量的关键。传统的相似度度量方法采用统计方法衡量项目之间的相似性,尤其在评分数据较稀疏时,存在各种弊端,导致邻居集合不准确,从而影响推荐系统的推荐质量。本文提出基于贝叶斯理论的协同过滤算法。该方法首先利用贝叶斯理论对用户爱好进行学习,分析用户对项目的固有特征的爱好度,然后在此基础上采用一种新颖的相似度度量方法计算项目的相似度,由此得到更有效的邻居集合。实验表明,本文算法能获得较好

的推荐精度。

1 传统的基于项的协同过滤技术

基于项的协同过滤推荐根据用户对相似项的评分预测该用户对目标项的评分,它基于这样一个假设:如果大部分用户对一些项的评分比较相似,则当前用户对这些项的评分也比较相似。基于项的协同过滤推荐系统使用统计技术找到目标项的若干最近邻居,由于当前用户对最近邻居的评分与对目标项的评分比较类似,所以可以根据当前用户对最近邻居的评分预测当前用户对目标项的评分。

设用户 u 未评分的项目集合为 P_u ,对于项目 $i \in P_u$,传统基于项的协同过滤技术使用如下方法计算用户 u 对项目 i 的评分。首先得到项目 i 与项目 j 共同评分的用户集合,利用其上的评分向量,计算项目 i 和项目 j 之间的相似度。相似度的度量方法^[8]主要有余弦相似性、Pearson 相关系数、调整余弦相似性等,具体计算公式如下所示。

余弦相似性:

$$\text{sim}(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 \times \|\vec{j}\|_2}$$

Pearson 相关系数:

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}}$$

其中: U 是对项目 i, j 共同评分的用户集合, \bar{R}_i 是项目 i 的平均得分, $R_{u,i}$ 是用户 u 对项目 i 的评分。

调整余弦相似性:

收稿日期:2009-04-17;修回日期:2009-06-01。

作者简介:孟宪福(1960-),男,辽宁大连人,副教授,主要研究方向:信息系统、分布式计算;陈莉(1986-),女,湖南邵阳人,硕士研究生,主要研究方向:电子商务推荐系统、P2P 推荐系统。

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}$$

其中: U 是对项目 i, j 共同评分的用户集合, \bar{R}_u 是用户 u 所有评分的平均值。

利用上述公式计算项目 i 与其他项目之间的相似度, 将与项目 i 相似度最高的若干项目组成 i 的邻居集合。得到邻居集合后, 利用如下公式计算用户 u 对项目 i 的评分, 设评分为 $P_{u,i}^{[9]}$, 那么:

$$P_{u,i} = \frac{\sum_{\text{all_similar_items}, N} (S_{i,N} \times R_{u,N})}{\sum_{\text{all_similar_items}, N} (|S_{i,N}|)}$$

其中: $S_{i,N}$ 为项目 i, N 之间的相似度, $R_{u,N}$ 为用户 u 对相似度最高的项目集合中项目 N 的评分。

2 基于贝叶斯理论的协同过滤技术

2.1 项目的特征属性

在实际的商务系统中, 用户对项目某类属性的偏好在其短期内是稳定的, 比如对服装的颜色、质地或对电影的情节、演员等。分析单个用户对项目特征的偏好, 有利于更精确地向目标用户推荐项目。

在电子商务领域中, 项目可以用其典型的一些属性来表示, 如表 1 所示, 我们将其抽象为 $\{M_1, M_2, \dots, M_n\}$, 并假设各属性之间是相互独立的。

表 1 项目特征值表

项目	M_1	M_2	...	M_n
Item ₁	10010	20001	...	30001
Item ₂	10011	20010	...	30010
...
Item _n	10100	20100	...	30100

将项目中的属性可能具有的值, 都进行编码。比如颜色属性, 用 8 bit 的二进制可以表示 256 种颜色。每种颜色对应一种编码。每个项目中每个特征属性的值为经过编码后的值。

$$P_u^j(A | t_1, t_2, \dots, t_n) = \frac{P_u^j(A | t_1) P_u^j(A | t_2) \dots P_u^j(A | t_n)}{P_u^j(A | t_1) P_u^j(A | t_2) \dots P_u^j(A | t_n) + [1 - P_u^j(A | t_1)][1 - P_u^j(A | t_2)] \dots [1 - P_u^j(A | t_n)]}$$

2.3 基于贝叶斯理论的推荐算法

2.3.1 寻找项目邻居集合

为了更有效地计算用户 u 上项目 j 的邻居集合, 在计算项目 j 和项目 i 的相似度时, 首先计算两个项目之间的评分相似度, 然后利用贝叶斯公式计算用户 u 喜欢项目 j 的概率。最后利用如下公式计算最终的相似度:

$$sim_u(j, i) = \alpha \cdot sim(j, i)$$

其中: $sim(j, i)$ 为项目 i, j 的评分相似度, $sim_u(j, i)$ 为最终相似度, α 为协调因子, 其计算公式如下:

$$\alpha = \begin{cases} p_u^j, & p_u^j > 0.5 \text{ 且 } u_i > \bar{u} \\ 1 - p_u^j, & p_u^j > 0.5 \text{ 且 } u_i < \bar{u} \\ p_u^j, & p_u^j < 0.5 \text{ 且 } u_i > \bar{u} \\ 1 - p_u^j, & p_u^j < 0.5 \text{ 且 } u_i < \bar{u} \end{cases}$$

其中: p_u^j 表示用户 u 喜欢项目 j 的概率, \bar{u} 表示用户 u 的平均分。

上述的相似度计算方法, 可以使得同时属于用户喜欢的两个项目或者同时属于用户不喜欢的两个项目的相似度具有更高的值。取相似度最高的若干项目组成目标项目的邻居集合, 因此邻居集合内的项目都是在用户喜好程度上以及项目特征

2.2 利用贝叶斯理论分析项目特征值

把用户的评分项目分成两类, 用户喜欢的项目集合和用户不喜欢的项目集合, 利用贝叶斯理论分析用户喜欢的项目集合中某项特征值出现的概率, 以及用户不喜欢的项目集合中某项特征值出现的概率, 以此计算未评分项目具有某些特征值时用户喜欢的概率。

将用户 u 的所有评分项目集合划分为 U^{like} 和 $U^{dislike}$, U^{like} 表示用户 u 喜欢的项目集合, $U^{dislike}$ 表示用户 u 不喜欢的项目集合。把用户 u 评分的平均分记为 \bar{u} , u_i 表示用户 u 对项目 i 的评分, 如果 $u_i > \bar{u}$, 则 $i \in U^{like}$, 否则 $i \in U^{dislike}$ 。设事件 A 为属于用户喜欢的项目集合, 事件 B 为属于用户不喜欢的项目集合, 用 m_1, m_2, \dots, m_n 代表特征值, $P_u(m_i | A)$ 表示用户 u 喜欢的项目集合中出现特征值 m_i 的概率, $P_u(m_i | B)$ 表示用户 u 不喜欢的项目集合中出现特征值 m_i 的概率。对每个用户建立一个哈希表, hash_like 对应用户喜欢的项目集合, hash_dislike 对应用户不喜欢的项目集合, 两个哈希表分别存储特征值及其在对应的集合中出现概率的映射关系。 $P_u^j(A | m_i)$ 表示项目 j 具有特征值 m_i 时, 用户 u 喜欢的概率。 $P_u^j(A | m_1, m_2, \dots, m_n)$ 表示项目 j 同时具有特征值 m_1, m_2, \dots, m_n 时, 用户 u 喜欢的概率。根据贝叶斯公式有:

$$P_u^j(A | m_i) = \frac{P_u(m_i | A) \times P_u(A)}{P_u(m_i | A) \times P_u(A) + P_u(m_i | B) \times P_u(B)} \quad (1)$$

在等同的概率情况下, $P_u(A)$ 和 $P_u(B)$ 都取为 0.5, 那么式(1)可以简化为:

$$P_u^j(A | m_i) = \frac{P_u(m_i | A)}{P_u(m_i | A) + P_u(m_i | B)}$$

设未评分项目 j 具有的特征值集合为 m_1, m_2, \dots, m_n , 如果 m_i 在已经计算出的 hash 表中, 则把 m_i 并入集合 $T = \{t_1, t_2, \dots, t_n\}$ 中。如果 T 为空则 $P_u^j(A | t_1, t_2, \dots, t_n)$ 取值为 0.5, 如果 T 不为空, 则由复合概率公式可得:

上与目标项目更相似的项目, 从而构造出更有效的邻居集合。

2.3.2 优化后的推荐算法

为了计算用户对未评分项目喜欢的概率值, 对每个用户的偏好进行学习, 建立该用户的 hash_like 表和 hash_dislike 表, 当用户给出新的评分项时, 更新 hash_like 表 hash_dislike 表。

设用户 u 的未评分项目集合为 N_u , 对任意项目 $j \in N_u$, 使用如下步骤计算用户 u 对项目 j 的评分 $P_{u,j}$ 。

1) 把用户的评分并入集合 U^{like} 和 $U^{dislike}$ 中。

2) 对于待预测的项目 j , 计算 $P_u^j(A | t_1, t_2, \dots, t_n)$ 。

3) 用上文介绍的相似性度量方法计算其他项目 $i (i \neq j)$ 和项目 j 之间的相似性 $sim_u(j, i)$ 。

4) 将相似性最高的若干项目作为项目 j 的邻居项目集合, 记为 M 。

5) 采用式(2) 计算用户 u 对项目 j 的评分 $P_{u,j}$:

$$P_{u,j} = \frac{\sum_{i \in M} sim_u(j, i) \times P_{u,i}}{\sum_{i \in M} |sim_u(j, i)|} \quad (2)$$

经过上述方法处理后, 把集合 N_u 中预测评分值最高的前

r 个项目推荐给用户 u 。

3 实验与结果分析

3.1 实验环境及比较标准

实验采用的 PC 配置为 Celeron CPU 3.06 GHz, 1 G RAM, Windows XP 操作系统, 算法实现语言为 Java。采用 Movielens 站点提供的数据集, Movielens 是一个基于 Web 的研究型推荐系统, 依据用户对电影的评分向用户提供相应的电影推荐列表^[8]。数据集中包含了 6 040 名用户对 3 900 部电影的评分, 大约 100 万个评分数据。为了分析不同实验数据集对本文算法的影响。我们从中随机抽取了 300 位用户的评分, 分成三个数据集: 第一组数据集为这些用户对电影 ID 号小于 1 000 的电影的评分数据组成, 第二组为这些用户对电影 ID 号 $\geq 1 000$ 且 $< 2 000$ 的评分数据组成, 其他的评分为第三组, 记为 D1, D2, D3。表 2 和图 1 描述了数据集的用户数、电影数、评分数、稀疏度^[8]以及评分值分布情况。实验将数据集中数据分为训练集和测试集。项目的特征属性为数据集中给出的电影的所有属性, 包括电影的年代以及电影的 18 种类别。

表 2 实验数据集

评分组	用户数量	电影数量	评分数量	稀疏度
D1	300	754	10 753	0.952
D2	300	728	12 961	0.940
D3	300	1 529	19 958	0.956

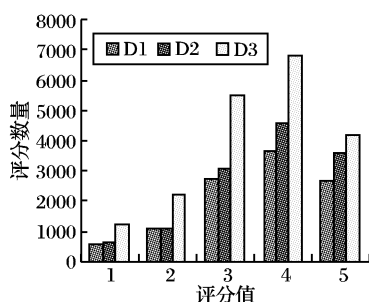


图 1 评分值分布情况

采用平均绝对偏差 (Mean-Absolute-Error, MAE)^[9] 作为度量标准验证算法的有效性。MAE 是通过计算预测的用户评分和实际的用户评分之间的偏差来度量预测的准确性。MAE 越小, 推荐质量越高。

设预测的用户评分集合为 $\{p_1, p_2, \dots, p_n\}$, 对应的实际用户评分集合为 $\{q_1, q_2, \dots, q_n\}$, 则 MAE 定义为^[9]:

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

3.2 实验结果及分析

3.2.1 推荐质量 (MAE) 比较

第一组实验采用评分分数以及评分电影数最多的 D3 数据集, 用户相似性度量方法采用 Pearson 相关系数法, 邻居集合大小分别取 5 ~ 40, 间隔为 5。用本文算法跟传统 Item-based 算法进行比较 (如图

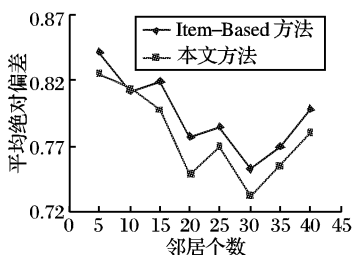


图 2 算法推荐质量 (MAE) 的比较

2)。由图 2 可知, 对于给定的不同邻居数本文提出的基于贝

叶斯理论的评分算法基本具有较小的 MAE 值。这是由于用户对项目感兴趣的原因在于项目具有用户喜爱的某些特性, 根据用户爱好对项目进行分类, 那么在同一类中的项目的特征更相似, 用户喜爱的程度更接近。通过分析未评分项目的特征得到用户喜好度, 计算相似度时加上喜好度权重, 使得与未评分项目属于同一类别中的项目具有与其更大的相似度。更多地参考这些项目的评分, 能更好地预测未评分项目的评分。当邻居集合数目大于 30 时, MAE 值开始逐渐增大, 其原因可能是随着邻居集合的扩大, 其中包括了相似度不太大的项目, 导致邻居集合不精确。

3.2.2 不同数据集对本文算法的影响

本组实验分别在 D1, D2, D3 三个不同的数据集上进行, 实验比较了不同数据集对推荐质量的影响。邻居集合大小从 5 增加到 40, 间隔为 5, 实验结果如图 3 所示。由图 3 可知, 本文算法在 D3 上具有最小的 MAE 值。由此可知尽管评分数据增多时, 数据稀疏度在一定程度上可能有所增加, 但是评分数据的增加有利于推荐精度的提高。经分析认为, 这是由于同一用户的评分数量增加, 可以更好地分析用户对特征的爱好, 使得可以查找到更好的最近邻居, 从而使 MAE 值降低。

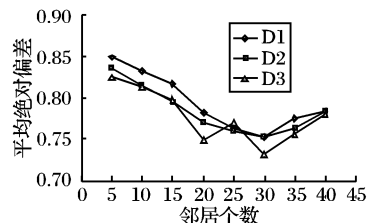


图 3 不同数据集对推荐质量的影响

4 结语

基于贝叶斯理论的协同过滤算法, 分析用户对项目特征的爱好, 在此基础上采用一种新的相似度计算方法, 从而使最近

邻居集合更加准确。实验结果表明, 本文提出的算法可以在一定程度上提高系统的推荐质量。

参考文献:

- [1] 邢春晓, 高凤荣, 战思南, 等. 适应用户兴趣变化的协同过滤推荐算法[J]. 计算机研究与发展, 2007, 44(2): 296-301.
- [2] 郭艳红, 邓贵仕, 雒春雨. 基于信任因子的协同过滤推荐算法[J]. 计算机工程, 2008, 34(20): 1-3.
- [3] SARWAR B M, KARYPIS G, KONSTAN J A, et al. Application of dimensionality reduction in recommender system - A case study[C]// Proceedings of the ACM SIGMOD International Conference on Knowledge Discovery and Data Mining. San Diego: ACM Press, 1999: 201-212.
- [4] 邓爱林, 左子叶, 朱扬勇. 基于项目聚类的协同过滤推荐算法[J]. 小型微型计算机系统, 2004, 25(9): 1665-1670.
- [5] CHICKERING D, HECHERMAN D. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables[J]. Machine Learning, 1997, 29(2/3): 181-212.
- [6] WOLF J, AGGARWAL C, WU K-L, et al. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering[C]// Proceedings of the ACM SIGMOD International Conference on Knowledge Discovery and Data Mining. San Diego: ACM Press, 1999: 201-212.
- [7] SARWAR B, KARYPIS G, KONSTAN J, et al. Analysis of recommendation algorithms for E-commerce[C]// Proceedings of the ACM Conference on Electronic Commerce. New York: ACM Press, 2000: 158-167.
- [8] 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.
- [9] 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 Press, 2001: 285-295.