文章编号：2095-2163(2020)01-0193-04 中图分类号：TP301 文献标志码：A

基于属性特征的个性化旅游推荐算法研究

丁 恒，黄全舟
（西安石油大学 计算机学院，西安 710065）

摘 要：随着旅游产业的兴起，旅游信息呈爆炸式增长，信息过载问题日益突出。为使用户能够高效、准确地得到所需信息，本文针对传统协同过滤算法仅采用单一的总体评分，从而导致相似度计算不准确的问题，提出了基于属性特征的推荐算法。该算法考虑了项目各属性特征的相似性，改进了传统方法相似度的计算方式，分别从多个维度进行相似度的衡量。实验结果表明，该算法在个性化旅游推荐中得到了很好的应用，相对于传统协同过滤算法有着更高的推荐精度，能够提升推荐的质量。

关键词：协同过滤；属性特征；个性化旅游推荐

Research on personalized tourism recommendation algorithm based on attribute features

DING Heng, HUANG Quanzhou
(School of Computer Science, Xi′an Shiyou University, Xi′an 710065, China)

【Abstract】 With the rise of tourism industry, tourism information is increasing explosively, and the problem of information overload is increasingly prominent. In order to enable users to get the required information efficiently and accurately, this paper proposes a recommendation algorithm based on attribute features to solve the problem that traditional collaborative filtering algorithm only uses a single overall score, therefore leads to inaccurate similarity calculation. The algorithm takes into account the similarity of the attributes of items, improves the traditional method of similarity calculation, and measures the similarity from multiple dimensions. The experimental results show that this algorithm has been well applied in personalized tourism recommendation, and has higher recommendation accuracy compared with the traditional collaborative filtering algorithm, which can improve the recommendation quality.

【Key words】 collaborative filtering; attribute features; personalized tourism recommendation

0 引 言

当今正处在信息时代，每天都有各式各样的信息不断涌来，信息过载问题日益严重。目前，个性化推荐技术[1]是解决这一问题的有力工具。个性化推荐技术是建立在大量的用户行为数据之上的，有了用户行为数据的强力依托，其身影遍布于互联网各类网站中，并已在电影、电子商务、图书、音乐等领域取得了显著的成效。推荐算法种类繁多，其中协同过滤推荐算法在个性化推荐中得到了更为广泛的应用[2]。

近年来，随着旅游产业规模日渐庞大，大量旅游信息相继产生，研究学界即已开始将个性化推荐技术应用于旅游行业。如何从海量数据中挖掘出高质量的旅游信息提供给用户？协同过滤算法将是一个不错的选择。本文在传统协同过滤算法的基础上，考虑了项目属性对相似度计算的影响，提出了一种基于属性特征的协同过滤算法，并将其应用于个性化旅游推荐中，以提升旅游推荐的质量。

1 协同过滤算法概述

协同过滤算法通过其它用户的偏好，找出与目标用户兴趣相似的用户所喜好的物品，然后进行推荐。协同过滤算法有基于记忆（内存）的协同过滤和基于模型的协同过滤两大类。

1.1 协同过滤算法分类

研究可知，基于记忆的协同过滤算法可分为：基于用户的协同过滤算法和基于项目的协同过滤算法。本节也将对这 2 种协同过滤算法进行概述分析。

（1）基于用户的协同过滤算法。作为最基本的推荐算法，基于用户的协同过滤算法最先被应用于推荐系统中[3]。基于用户的协同过滤算法由用户的兴趣产生推荐，其基本思想是依据用户对物品的偏好找出与其兴趣相似的邻居用户，再将邻居用户偏爱的、且目标用户不曾涉及到的物品推荐给目标用户。

（2）基于项目的协同过滤算法。该算法目前在

作者简介：丁 恒（1995-），男，硕士研究生，主要研究方向：软件工程、智能算法；黄全舟（1964-），男，副教授，主要研究方向：软件工程、人工智能。

收稿日期：2019-10-11

互联网行业的应用很普及,Amazon、YouTube、Hulu 的推荐算法都是由此演化而来。算法是依据项目之间的共性来进行协同过滤,其基本思想是分析用户的历史偏好,将用户以往所喜欢物品的相似物品推荐给用户。

1.2 协同过滤算法存在的问题

(1)精确性问题。拥有高精确性的推荐算法能为用户提供更加精准的推荐,提升用户满意度。可靠的推荐结果是一个推荐系统赖以生存的关键。显然,如果一个推荐系统不能产生优质的推荐,就会失去大量的用户。

(2)数据稀疏性问题。相似度的计算依赖于用户-项目评分矩阵,然而用户只是对少量的项目进行了评分,相对比例仅为 1% ~ 2%[4]。这就造成评分矩阵过于稀疏,用户寻找相似邻居成为难题,大大降低了推荐的质量。

(3)冷启动问题。随着新用户、新项目的不断加入,但在此之前并没有任何有关新加入用户或是新加入项目的记录,从而导致系统无法进行推荐。

2 传统协同过滤算法的步骤

2.1 建立评分矩阵

协同过滤算法所涉及的数据为用户对项目的历史评分数据,可以将这些数据用 $m \times n$ 的评分矩阵 $\boldsymbol{R}$ 来表示,其中,m 表示用户数,n 表示项目数,u 表示用户,i 表示项目,r_{ij} 表示用户 u_i 对项目 i_j 的评分值,评分值一般用整数 1 ~ 5 来表示(0 表示用户未对该项目进行评价),评分越高则用户越喜欢。研究中给出了一个用户 - 项目评分矩阵的样例见表 1。

表 1 用户-项目评分矩阵

Tab. 1 User-item rating matrix

User	Item					
	i_1	i_2	…	i_j	…	i_n
u_1	1	3	…	0	…	4
…	…	…	…	…	…	…
u_i	3	4	…	r_{ij}	…	0
…	…	…	…	…	…	…
u_m	5	2	…	3	…	r_{mn}

2.2 选取最近邻

最近邻的选取是协同过滤算法中至关重要的一部分,并直接影响着预测的准确度,下面,将基于用户的协同过滤算法作为例子,对最近邻集合的产生过程进行论述。首先,通过有关公式计算出不同用户与目标用户 u 的相似性,由此找出与 u 最为相似的 k 个最近邻所组成的最近邻集合 $K_u = \{u_1, u_2, \ldots, u_k\}$。常见的相似度计算方法有:余弦相似性、欧几里得距离、Pearson 相关相似性。本文仅选取同等条件下性能相对优越的 Pearson 相关相似性进行说明[5],Pearson 相关相似性的数学定义即如式(1)所示:

$$sim(u,v) = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r_u})^2 \sum_{i \in I_{u,v}} (r_{v,i} - \overline{r_v})^2}}. \tag{1}$$

其中,$I_{u,v}$ 表示用户 u、v 共同评过分的项目集;$r_{u,i}$ 和 $r_{v,i}$ 分别表示用户 u、v 对项目 i 的评分值;$\overline{r_u}$ 和 $\overline{r_v}$ 为用户的评分均值。

2.3 预测评分并生成推荐

在得到 k 最近邻集合后,可根据集合中相似用户的数据对目标用户未评分的项目进行评分预测,并将预测评分从高到低进行排序,从而生成 top-N 推荐给目标用户进行选择。具体计算如表达式(2)所示:

$$P_{u,i} = \overline{r_u} + \frac{\sum_{x \in k_u} sim(x,u)(r_{x,i} - \overline{r_x})}{\sum_{x \in k_u} sim(x,u)}. \tag{2}$$

其中,$sim(x,u)$ 是通过相似度计算公式所得出的两用户之间的相似性,k_u 表示用户 u 的最近邻集合。

基于项目的协同过滤算法预测和推荐过程与此基本类似,这里,仅详细探讨阐述了基于用户的协同过滤算法研究。

3 基于属性特征的协同过滤算法

3.1 算法分析

传统的协同过滤算法基于项目整体评分进行相似度的计算,并没有衡量项目各属性对推荐结果的影响,这样的推荐往往不够准确。结合属性特征从多个维度进行分析计算能提升整个算法的精度,使推荐的结果更能符合用户需求。同时,结合属性特征能够很好地解释用户为什么喜欢这些项目,体现不同用户所重视的不同方面,从而进行个性化的推荐。

下面将结合旅游这一领域,通过 2 个具体的例子来进行说明,内容表述详见如下。

假设有 3 个用户(u_1, u_2, u_3)和 4 个景点(i_1, i_2, i_3, i_4),其中有些景点评分已知,由此预测用户 u_1 对景点 i_4 的评分。传统协同过滤算法用户 - 景点评分见表 2。通过表 2 中的数据进行相似性计算发现 u_1 和 u_2 是相似用户,从而得出用户 u_1 对景点 i_4 的预测

评分为 5 分。

在表 2 的基础上，研究给出对各个景点的总体评分（总体评分为各属性评分和的平均值）以及景点分别在美丽、人文、休闲、刺激、特色、浪漫六个属性上的评分，评分结果见表 3。其中，括号内为各属性的评分值。

表 2　传统协同过滤算法用户-景点评分表

Tab. 2　Traditional collaborative filtering algorithm user - site rating table

User	Item			
	i_1	i_2	i_3	i_4
u_1	5	4	3	?
u_2	5	4	3	5
u_3	2	4	1	3

表 3　多属性用户-景点评分表

Tab. 3　Multi-attribute user-site rating table

User	Item			
	i_1	i_2	i_3	i_4
u_1	3(3,2,2,4,5,2)	4(5,3,3,4,5,4)	2(2,1,3,4,1,1)	3(3,3,3,4,2,3)
u_2	4(2,5,5,4,3,5)	5(5,5,5,5,5,5)	3(5,4,2,2,2,3)	4(4,5,3,5,4,3)
u_3	3(1,4,5,1,2,5)	4(4,5,5,2,3,5)	2(4,2,1,1,2,2)	?

由表 3 可知，根据以往单个评分的计算来看，用户u_1 和用户 u_3 最为相似。但在引入多属性的评分准则后，则需要从多维度进行考虑，从而找出相似用户。这样不仅体现了用户不同的偏好，而且所得出的推荐结果会更加令人满意。显然，用户u_1 和用户u_3 在各属性的评分值上存在较大偏差，用户u_2 和用户u_3 各属性值更为相似，因此 u_2 和u_3 为最相似用户。总体评分的预测也应该依据 u_2 来决定。

由上述两个例子可以看出，虽然总体的评分在一定程度上表达了用户的喜爱程度，但多属性的评分更能体现用户对不同方面的偏好，很好地解释了用户喜爱的理由，使得推荐结果更加精确。因此，本文将采用基于属性特征的方式来对传统协同过滤算法进行改进，以求能够提升推荐的质量，得到更多用户的满意评价。

3.2　改进后的相似性度量方法

在合理评分的前提下，景点的总体评分与各属性评分之间存在一定的关联性。总体评分的高低是用户对各属性特征满意程度的体现，如果用户对各个方面都不是很满意，那么对于整体印象就会大打折扣，评分就会较低。只有在各方面都满意的前提下，整体才会得到高分。因此，研究通过景点的多属性特征评分来代替传统单一的总体评分计算各用户之间的相似度，最终使得最近邻集合更加准确，提高推荐算法的精度。

用户对景点的在各个属性的打分可以看做一个 k 维向量，例如，$\boldsymbol{R}(u,i)=(r_1,r_2,r_3,\ldots,r_k)$，$\boldsymbol{R}(v,i)=(r'_1,r'_2,r'_3,\ldots,r'_k)$，研究使用多维向量间的欧式距离公式计算对于同一景点评过分的两用户间的距离，其数学公式可表示为：

$$d_{\boldsymbol{R}(u,i),\boldsymbol{R}(v,i)}=\sqrt{\sum_{i=0}^{k}|r_i-r'_i|^2}\ , \tag{3}$$

假设用户 u 和用户 v 共同评分过的景点数目表示为 $I(u,v)$，则可用如下公式计算两用户间的总体距离，其数学公式可表示为：

$$d_{u,v}=\frac{1}{|I(u,v)|}\sum_{i\in I(u,v)}d_{\boldsymbol{R}(u,i),\boldsymbol{R}(v,i)}\ , \tag{4}$$

距离和相似度之间存在反比关系，因此在使用距离来代表用户间的相似性时，拟用以下公式进行转换，其数学公式可表示为：

$$sim(u,v)=\frac{1}{1+d_{u,v}}\ . \tag{5}$$

上述方法基于属性特征，用多属性评分的方式来代替传统单一的总体评分，从而改变了传统协同过滤算法在相似度方面的计算方式。在下一节，将通过旅评网上的评分数据进行实验分析，来评估这种改进方式是否能对传统协同过滤算法在计算精度上有所提高，提升推荐的质量。

3.3　算法描述

基于属性特征的协同过滤算法的流程描述如下。

输入：多属性用户 - 景点评分矩阵$\boldsymbol{R}$，最近邻个数k

输出：用户 u 所对应的 Top-N 推荐列表

（1）利用构建好的多属性用户-景点评分矩阵$\boldsymbol{R}$，依据公式（3）~（5）计算出用户间的相似度 $sim(u,v)$，并取最为相似的 k 个用户组成最近邻集合 $K_u=\{u_1,u_2,\ldots,u_k\}$。

（2）根据最近邻 k_u 集合中相似用户的数据，利用公式（2）计算出用户 u 未评分景点 i 的预测评分值 $p_{u,i}$。

（3）将预测评分值进行降序排列后，产生 Top-N 推荐列表给用户进行选择。

4 实验结果与分析

4.1 数据来源

实验中的数据取自“旅评网”。该网站用户量充足,可提供足够多的景点评分数据。对于景点的评定,用户可分别从“美丽”、“人文”、“休闲”、“刺激”、“特色”、“浪漫”六个属性特征进行评分,评分值区间为 1-5 分。采用网络爬虫技术采集了 5 216 名注册用户对 3 260 个景点的 23 563 条评分数据,将数据分为训练集和测试集两部分,其中 80%为训练集,其余 20%为测试集。用户评分的数据片段见表 4。

对表 4 中数据进行预处理后,构建多属性特征用户-景点评分矩阵。

表 4 用户评分数据片段

Tab. 4 User score data fragment

用户	旅游景点	美丽评分	人文评分	休闲评分	刺激评分	特色评分	浪漫评分	综合评分
Aurora03	华山	4	4	5	4	4	5	5
wilsy	秦兵马俑	3	5	2	5	5	3	4
那年初夏	丽江古城	5	5	5	5	5	4	5
Aurora03	华清池	4	5	4	3	4	4	3
无尽慈悲	袁家界	5	3	4	5	5	5	5
水影	长白山	4	3	4	3	3	4	3
……	……	……	……	……	……	……	……	……

4.2 评价指标

本文采用平均绝对误差 *MAE* 作为评判推荐结果优劣的标准[6]。*MAE* 值越小,表示该推荐算法的推荐质量就越高,相应的计算公式为:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N}. \quad (6)$$

其中,p_i 是用户真实的评分值;q_i 为该算法所预测的评分值;N 为预测评分的总条目数。

4.3 结果及分析

为了检验本文所提出算法的实际推荐效果,将其与传统协同过滤算法在同一数据集上进行了比较,结果如图 1 所示。

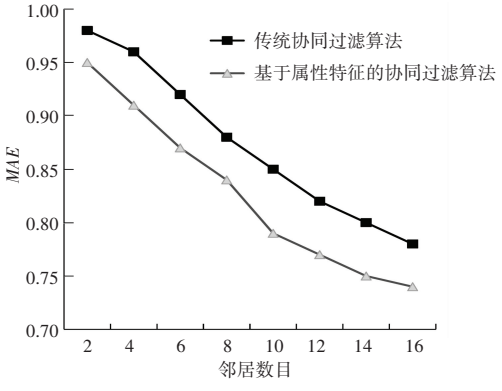

图 1 推荐算法精度比较图

Fig. 1 Accuracy comparison chart of recommendation algorithm

由图 1 可知,本文算法在相同邻居数目的条件下有着更小的 *MAE* 值,推荐性能显著优于传统方法。

5 结束语

本文针对传统协同过滤算法采用单一评分,导致相似度计算存在偏差,影响整个算法的精确性。提出了一种基于属性特征的协同过滤算法并应用于个性化旅游推荐中,以景点多属性评分代替单一评分来计算用户间的相似性。实验结果表明,本文算法能够解决推荐精确度的问题,使推荐的质量得到了一定程度的提升。

参考文献

[1] 冷亚军,陆青,梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能,2014,27(8):720-734.

[2] 毛勇. 基于协同过滤的推荐算法研究[J]. 计算机时代,2018(7):28-31.

[3] 项亮. 推荐系统实践[M]. 北京:人民邮电出版社,2012.

[4] 张忠平,郭献丽. 一种优化的基于项目评分预测的协同过滤推荐算法[J]. 计算机应用研究,2008,25(9):2658-2660,2683.

[5] BOBADILLA J, ORTEGA F, HERNANDO A. A collaborative filtering similarity measure based on singularities[J]. Information Processing and Management,2012,48(2):204-217.

[6] HERLOKER J L, KONSTAN J A, TERVEEN L G, et al. Evaluating collaborative filtering recommender system[J]. ACM Transactionon Information Systems, 2004, 22 (1) :5-53.