Computer Science and Application 计算机科学与应用, 2018, 8(11), 1689-1695
Published Online November 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.811186

Hans汉斯

User Feature Collaborative Filtering Algorithm Integrating Item Factors

Wenjuan Cheng, Yunhai Liu

Hefei University of Technology, Hefei Anhui
Email: qwertylevel3@126.com

Received: Oct. 28th, 2018; accepted: Nov. 9th, 2018; published: Nov. 16th, 2018

Abstract

In order to solve the problem of inaccuracy of project classification based on user partial feature collaborative filtering algorithm, a kind of user partial feature collaborative filtering algorithm is proposed, which combines the approximate item (AICF). The algorithm combines project label data and user score data to calculate the comprehensive similarity of the project to obtain the approximate items of the recommended items. Then, the nearest neighbor is calculated in the approximate project. Finally, the recommended user is recommended by the nearest neighbor to the designated user. Experiments show that by combining the project label data and improving the selection of neighbor items, the recommendation effect of the final collaborative filtering algorithm can be improved.

Keywords

Collaborative Filtering, Item Label, Similarity

融合项目因素的用户部分特征协同过滤算法

程文娟，刘云海

合肥工业大学，安徽 合肥
Email: qwertylevel3@126.com

收稿日期：2018年10月28日；录用日期：2018年11月9日；发布日期：2018年11月16日

摘 要

为解决基于用户部分特征协同过滤算法中出现的项目分类不准确问题，本文提出了一种融合了近似项目的用户部分特征协同过滤算法(AICF)。该算法通过融合项目标签数据和用户评分数据来计算项目综合相

文章引用：程文娟，刘云海．融合项目因素的用户部分特征协同过滤算法[J]．计算机科学与应用，2018，8(11)：1689-1695. DOI: 10.12677/csa.2018.811186

似度，以获取待推荐项目的近似项目，然后在近似项目中计算待推荐用户最近邻，最后根据近邻用户对指定用户推荐。实验表明，通过结合项目标签数据改进近邻项目的选择，可以提高最终协同过滤算法的推荐效果。

关键词

协同过滤推荐，项目标签，相似性

1. 引言

随着人类进入大数据时代，互联网上的信息呈指数爆炸式增长，如何从茫茫多数据中获取自己感兴趣的数据成为了人类当前所面临的巨大问题。于是便产生了能够主动帮助用户提供有效信息的推荐系统。而协同过滤技术作为最成功的推荐技术之一受到了越来越多的研究者的关注。目前，几乎所有的大型商务系统中都不同程度的使用了各种形式的推荐系统[1]。

通常，协同过滤技术主要包括基于用户的协同过滤推荐，基于项目的协同过滤推荐和基于矩阵分解的协同过滤推荐[2]，其中基于用户的协同过滤算法是目前运用最为广泛的算法。该算法首先需要找到和当前待推荐用户有相似兴趣的其他用户，然后根据这些用户的行为将他们感兴趣的内容推荐给该待推荐用户[3]。但是该算法也面临着数据稀疏性，冷启动问题和系统延展性等问题。

为了解决以上问题，邓爱林等[1]提出了一种基于项目评分预测的协同过滤算法 Sarwar 等[4]提出通过矩阵奇异值分解来减少项目空间的维数，进而提高推荐效果。黄裕洋等[5]提出了一种综合用户和项目因素的协同过滤算法，该算法在产生推荐时同时考虑了用户和项目的近邻结合，并能够自适应的调节目标用户和目标项目的最近邻集合。程高伟等[6]提出结合项目标签和用户评分的协同过滤算法，该算法在传统协同过滤算法中引入了项目标签数据，提高了最终的推荐准确率。

其中李永超等[7]认为虽然用户可能在某些项目类别中口味相似，但是在其他项目类别中口味可能大相径庭，所以需要基于用户部分特征进行推荐，故提出了一种基于用户部分特征的协同过滤算法。但是某些项目可能很难准确的分为某一个固定类别，或者某些项目可能处于该项目类别边缘，如果直接使用该聚类结果中的项目可能导致评分偏差。另一方面，在项目类别分类中，项目标签数据并没有完全发挥出作用。故上述基于用户部分特征协同过滤基础上，本文提出了一种融合了项目因素的用户部分特征协同过滤算法，该算法在基于用户部分特征的协同过滤基础上，通过融合了项目标签数据从而提高项目分类的准确性，进而提高用户近邻选择的准确性，同时可以避免在用户近邻选择中出现的因为只考虑全局相似性出现的误差，以此提高最终的推荐结果准确性。

2. 融合项目因素的用户部分特征协同过滤算法

融合项目因素的用户部分特征协同过滤算法可分为以下四个阶段

1) 评分矩阵预处理
2) 近似项目选择
3) 近邻用户生成

4) 推荐生成

作为算法的输入数据，用户－项目评分矩阵通常可以表示为一个 m*n 矩阵 R，其中 m 为用户数量，n 为项目数量，矩阵元素 rij 表示第 i 个用户对第 j 个项目的评分值。

2.1. 评分矩阵预处理

相对于数目巨大的总项目数来说，往往用户只能对少量项目进行评分，这就导致评分矩阵极度稀疏，所以在进行项目分类之前首先要对评分矩阵预处理。

传统处理评分矩阵的方法包括矩阵缺失值预测填充，矩阵分解等方法。在这里，项目评分数量可能会影响到最终的推荐结果，举例来说：一部好莱坞大片有 10,000 个观众投票，一部小成本的文艺片只有 100 个观众投票。这两者的投票数量上的不同可能导致最终推荐上的偏差。故在这里采用了[6]提出的未评分项目预测方法，该方法通过项目流行系数来解决热门项目评分偏差问题：

项目流行系数 w：

$$w = 1\Big/\left(\log\left(1+S(i)\right)\right) \tag{1}$$

这里 $S(i)$表示项目 i 已经被评分的总次数

$$\tilde{r}_{ui} = \overline{R} + b_u + b_i * w \tag{2}$$

这里 $\tilde{r}_{ui}$ 表示预测填充值。$\overline{R}$ 表示所有用户评分均值，b_u 表示用户 u 和用户评分均值的偏差，b_i 表示项目 i 和用户评分均值的评分偏差。

2.2. 相似度计算

用户之间的相似度计算通常包含余弦相似性，相关相似性以及修正过的余弦相似度[8]，在这里采用修正过的余弦相似度算法。该方法通过减去用户对项目的平均评分来解决传统余弦相似度中未考虑不同用户评分尺度的问题[1]：

$$sim(u,v) = \frac{\sum_{i\in I_{uv}}\left(r_{ui}-\overline{R_u}\right)*\left(r_{vi}-\overline{R_v}\right)}{\sqrt{\sum_{i\in I_{uv}}\left(r_{ui}-\overline{R_u}\right)^2}\sqrt{\sum_{i\in I_{uv}}\left(r_{vi}-\overline{R_v}\right)^2}} \tag{3}$$

这里 u，v 表示用户 u，v。r_{ui} 表示用户 u 对项目 i 的评分，$\overline{R_u}$ 表示用户 u 的平均评分，I_u，I_v 表示用户 u，v 已经评价过的项目集合，I_{uv} 表示用户 u 和用户 v 的共同评分项目集合。

相似的，项目之间的相似度计算也可以采用类似的公式：

$$sim(i,j) = \frac{\sum_{u\in U_i\cap U_j}\left(r_{u,i}-\overline{r_i}\right)*\left(r_{u,j}-\overline{r_j}\right)}{\sqrt{\sum_{u\in U_i\cap U_j}\left(r_{u,i}-\overline{r_i}\right)^2}\sqrt{\sum_{u\in U_i\cap U_j}\left(r_{u,j}-\overline{r_j}\right)^2}} \tag{4}$$

其中，U_i，U_j 表示项目 i 和项目 j 中存在评分值的用户集合。r_i 表示项目 i 的平均被评分值。$r_{u,i}$，$r_{u,j}$ 表示用户 u 对项目 i 和 j 的评分值。

通常计算两个布尔变量之间的相似度使用 Jaccard 系数：

$$J(A,B) = \frac{|A\cap B|}{|A|+|B|-|A\cap B|} \tag{5}$$

这里采用 Jaccard 系数计算项目之间在标签上的相似度。

2.3. 近似项目生成

基于用户的协同过滤算法基于以下假设：如果用户之间对一些项目的评分比较相似，则他们对其它项目的评分也将会比较相似[5]。但是用户口味可能随着项目种类的不同而大相径庭[7]。故需要先对项目进行分类，然后计算近邻用户过程中仅根据在相同类别项目中计算用户相似度。但是传统聚类算法中部分项目可能难以分为具体某一个类别或者可能该项目处于分类边缘，则此时基于该项目的推荐可能会有偏差。另外，仅仅使用用户评分矩阵并不能对项目作出准确客观的评价。故而在对项目分类过程中，需要综合考虑项目评分相似度和项目标签信息，最终加权计算项目综合相似度。这样保证了只有项目评分较多且标签相似的项目才具有较高的相似度，从而提高了项目分类的准确性。

首先用公式(4)计算各个项目和待推荐项目的评分相似度 $sim_p(i,j)$，然后利用公式(5)计算各个项目和待推荐项目之间的标签相似度 $sim_t(i,j)$，最终计算项目的综合相似度：

$$sim(i,j)=\alpha sim_p(i,j)+(1-\alpha)sim_t(i,j) \tag{6}$$

这里 α 用来调整评分相似度和标签相似度之间的影响权重。

综合相似度计算完毕后即可选择和待推荐项目相似度最大的 n 个项目，作为近似项目集合。

2.4. 近邻用户生成

利用公式(3)：

$$sim(u,v)=\frac{\sum_{i\in I_{uv}}\left(r_{ui}-\overline{R_u}\right)*\left(r_{vi}-\overline{R_v}\right)}{\sqrt{\sum_{i\in I_{uv}}\left(r_{ui}-\overline{R_u}\right)^2}\sqrt{\sum_{i\in I_{uv}}\left(r_{vi}-\overline{R_v}\right)^2}} \tag{7}$$

计算用户和指定用户之间的相似度，但此时项目集合仅考虑 2.3 中计算生成的近似项目集合。计算完毕后选取相似度最大的 k 个项目，作为近邻用户集合。

2.5. 推荐结果生成

利用 2.4 节计算生成的近邻用户集合来计算指定用户 u 对待推荐项目 i 的评分：

$$\hat{r}_{ui}=\overline{R}_u+\frac{\sum_{v\in N_u}sim(u,v)*\left(r_{vi}-\overline{R}_v\right)}{\sum_{v\in N_u}\left|sim(u,v)\right|} \tag{8}$$

这里 $\overline{R}_u$ 表示用户 u 的评价项目评分，N_u 为用户 u 的最近邻用户集合。这样便最终得到了 u 关于项目 i 的评分。

3. 实验结果和分析

3.1. 数据集

实验采用的数据集是目前衡量推荐算法质量时比较常用的 MovieLens 100K 数据集，由美国明尼苏达大学 GroupLens 研究小组创建并维护。该数据中包括 100,004 个评分数据，其中包括 671 个用户和包括 9125 个电影的 1296 个标签数据。数据集中评分范围为 1~5，数值越大表示用户对评分电影兴趣越大。本次实验按照 5 折交叉实验，取 5 次实验平均值作为最终结果。

3.2. 评价标准

学术研究中通常使用平均绝对误差 MAE (mean absolute error)来评价推荐系统的推荐质量：

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}|pi-ri|}{N} \tag{9}$$

这里 N 表示测试集的数据个数，pi 为预测值，ri 为实际评分值。其最终 MAE 值越小表示预测结果越准确。

3.3. 实验结果分析

3.3.1. 参数调整

首先讨论公式(6)中的 α 取值。α 和 $1-\alpha$ 表示计算项目综合相似度中项目的评分相似度和项目标签相似度所占的比重。其取值会影响到近似项目计算的准确性，进而影响到最终的推荐效果。在这里，对 α 取不同值，观察其对最终 MAE 值的影响。

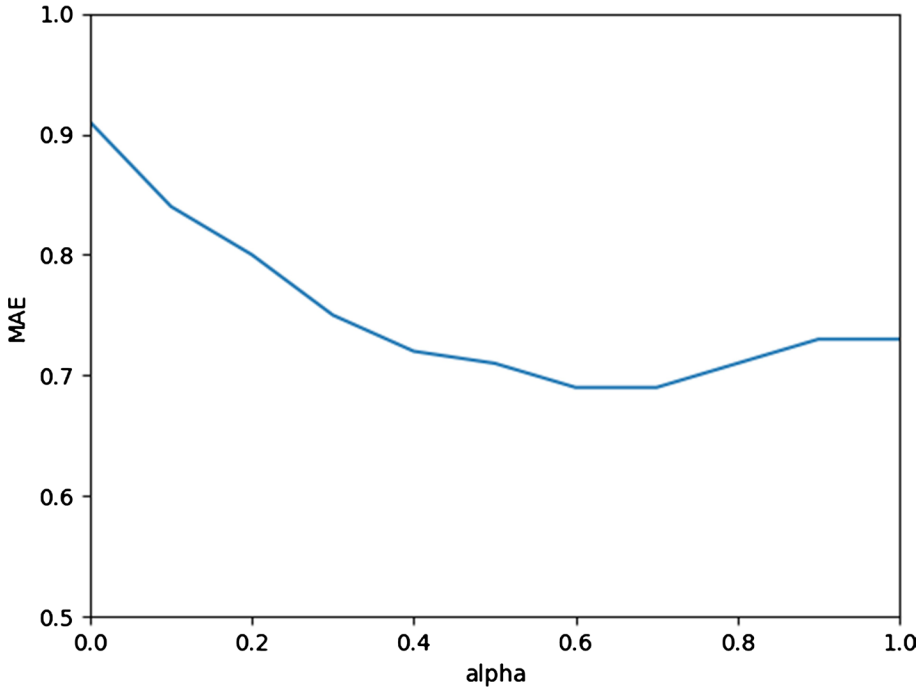

Figure 1. α result diagram
图 1. α 取值结果图

由图 1 可以看出，当 α 取 0.5~0.8 之间时，最终评分准确度最高。

3.3.2. 实验结果比较

这里使用本文提出的算法和传统的基于用户聚类的协同过滤算法(UBCF)，综合用户和项目因素的协同过滤算法(HCFR)，基于用户部分特征的协同过滤算法(UPCF)的平均绝对误差 MAE 进行对比观察实验：

由图 2 可以看出，该算法和基于用户部分特征的协同过滤算法有最高的 MAE 值，因为其考虑了用户的部分相似性而非用户的全局相似性。于此同时，本文提出的融合项目因素的用户部分特征协同过滤算法(AICF)由于在计算过程中融合了项目标签数据来计算近似项目，提高了近似项目选择的准确性，故

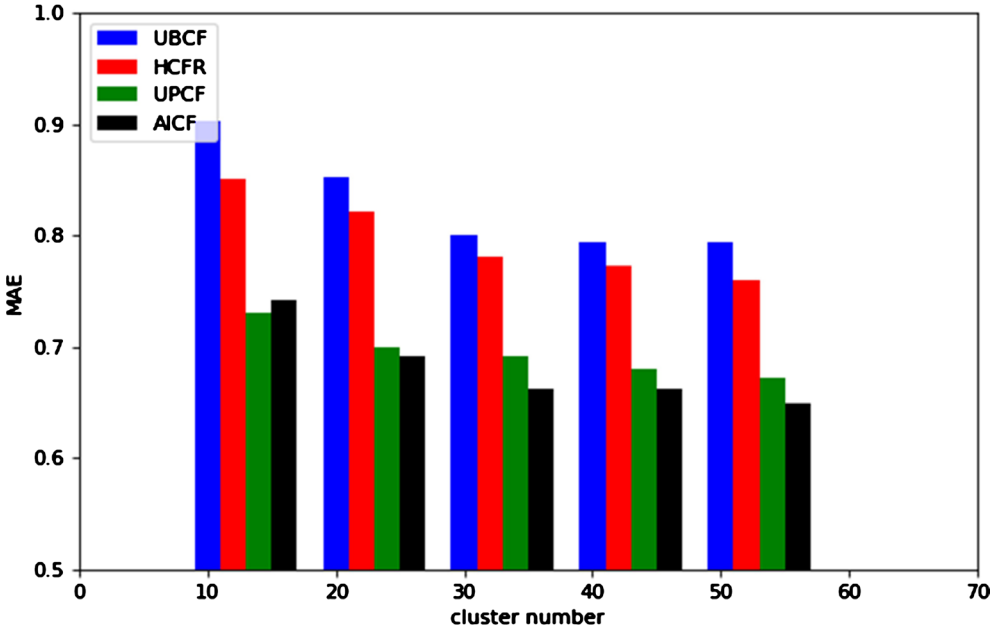

Figure 2. Experimental result diagram
图 2. 实验结果图

最终平均绝对误差要低于基于用户部分特征的协同过滤算法。

计算性能上，项目近似计算可以通过 hadoop 或 cuda 并行加速[9] [10] [11] [12] [13]，也可事先离线计算完毕。用户近邻计算因为是在近似项目集合中，故复杂度远小于传统的基于用户的协同过滤算法。

4. 结语

本文在基于用户部分特征的协同过滤基础上对其作出了改进，通过综合标签数据和用户评分数据的方法来提高近邻项目选择的准确率。最终实验表明，通过融合项目评分数据和项目标签数据来计算近似项目集合，可以有效避免简单项目聚类所导致的近似项目选择不准确问题，故最终可以有效提高最终用户近邻选择的准确率，进而提高最终推荐准确率。

但是由于针对每个待推荐项目都要计算其相似项目，其计算效率仍然不是十分理想。现在也出现了很多利用 GPU 并行加速的方法[9]。此外，公式(6)中 α 值的选取是事先人为指定，不够灵活。而[14]中也提到了一种参数自适应的方法。如何利用计算并行化和 GPU 加速提升算法效率，进一步简化算法，提升算法灵活性，将是下一阶段研究的方向。

基金项目

安徽省质量工程项目(2016ckjh141)。

参考文献

[1] 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.

[2] Bobadilla, J., Ortega, F., Hernando, A., *et al.* (2013) Recommender Systems Survey. *Knowledge-Based Systems*, **46**, 109-132. https://doi.org/10.1016/j.knosys.2013.03.012

[3] Deshpande, M. and Karypis, G. (2004) Item-Based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems*, **22**, 143-177. https://doi.org/10.1145/963770.963776

[4] Sarwar, B.M., Karypis, G., Konstan, J.A., *et al.* (2000) Application of Dimensionality Reduction in Recommender

System—A Case Study. ACM Webkdd Workshop. https://doi.org/10.21236/ADA439541

[5] 黄裕洋, 金远平. 一种综合用户和项目因素的协同过滤推荐算法[J]. 东南大学学报(自然科学版), 2010, 40(5): 917-921.

[6] 程高伟, 丁亦喆, 吴振强. 结合用户评分和项目标签的协同过滤算法[J]. 计算机技术与发展, 2015(3): 71-75.

[7] 李永超, 罗军. 基于用户部分特征的协同过滤算法[J]. 计算机系统应用, 2017, 26(3): 204-208.

[8] Breese, J.S., Heckerman, D. and Kadie, C. (1998) Empirical Analysis of Predictive Algorithms for Collaborative Filtering. 14*th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 43-52.

[9] 许建, 林泳, 秦勇, 等. 基于 GPU 的并行协同过滤算法[J]. 计算机应用研究, 2013, 30(9): 2656-2659.

[10] 闫永刚, 马廷淮, 王建. KNN 分类算法的 MapReduce 并行化实现[J]. 南京航空航天大学学报, 2013, 45(4): 550-555.

[11] 韦泽鲲, 夏靖波, 付凯, 等. 并行 MapReduce 模型下的一种改进型 KNN 分类算法[J]. 空军工程大学学报·自然科学版, 2017, 18(1): 92-98.

[12] 陈薇. 基于 MapReduce 的机器学习并行化研究与实现[J]. 产业与科技论坛, 2017, 16(9): 69-70.

[13] 涂敬伟, 皮建勇. 基于 MapReduce 和分布式缓存的 KNN 分类算法研究[J]. 微型机与应用, 2015, 34(2): 18-21.

[14] 黄创光, 印鉴, 汪静, 等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报, 2010, 33(8): 1369-1377.