Advances in Applied Mathematics 应用数学进展, 2017, 6(3), 267-274
Published Online May 2017 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2017.63032

Hans汉斯

An Improved and More Accurate Hybrid Recommendation Algorithm

Quanmin Wang, Shi Gu, Zhenguo Li, Kaiyang Wang, Yanfeng Sun

Beijing University of Technology, Beijing
Email: 18800168633@163.com

Received: May 6th, 2017; accepted: May 21st, 2017; published: May 27th, 2017

Abstract

Recommender system can filter some useless information and can predict whether the users love given resources. Content-based recommendation and collaborative filtering recommendation algorithm is the main personalized recommendation method. However, with the continuous increase of user projects, there are sparse, cold start and other issues in the user-project scoring matrix. In response to this problem, we propose a unique cascade hybrid recommendation method that uses rating data, demographic data, and feature data to calculate the similarity between projects. Experiments show that our method is superior to the traditional recommendation system algorithm.

Keywords

Personalization, Content-Based, Collaborative Filtering, Demographic

一种改进的更准确的混合推荐算法

王全民，谷　实，李振国，王开阳，孙艳峰

北京工业大学，北京
Email: 18800168633@163.com

收稿日期：2017年5月6日；录用日期：2017年5月21日；发布日期：2017年5月27日

摘　要

推荐系统可以过滤一些无用信息，可以预测用户是否喜欢给定的资源。基于内容的推荐和协同过滤推荐算法是目前主要的个性化推荐方法。但是随着用户项目的不断增加，用户-项目评分矩阵存在着稀疏性、冷启动等问题。针对此问题，我们提出了一个独特的层叠混合推荐方法，使用评级数据，人口统计数据和特征数据来计算项目之间的相似度。实验表明我们的方法优于传统的推荐系统算法。

文章引用: 王全民, 谷实, 李振国, 王开阳, 孙艳峰. 一种改进的更准确的混合推荐算法[J]. 应用数学进展, 2017, 6(3): 267-274. https://doi.org/10.12677/aam.2017.63032

关键词

个性化，基于内容，协同过滤，推荐算法

1. 引言

近年来，数字信息、电子资源和在线服务的数量呈指数级增长。信息超载出现了一个潜在的问题，即如何给用户过滤和有效地传递相关信息。这是一个可以过滤用户不需要的信息和可以预测用户需要的物品的系统。这样的系统被称为推荐系统。

假设 $M=\{m_1,m_2,\cdots,m_x\}$ 是所有用户的集合，$N=\{n_1,n_2,\cdots,n_y\}$ 是所有可能的被推荐的项目，$r_{m_i n_j}$ 是用户 m_i 对项目 n_j 的评分。

$$u: M\times N\to R \tag{1}$$

R 是一个完全有序的集合。对于每一个用户 $m_i\in M$，推荐系统的目的是选择最符合用户的兴趣的项目 $n'_j\in N$。我们可以指定如下：

$$n'_{j_{m_i}}=\arg\max_{n_j\in N} u\left(m_i,n_j\right):\forall m_i\in M \tag{2}$$

协同过滤系统可以分为两类：基于内存的协同过滤和基于模型的协同过滤。基于内存的协同过滤，其基本思想是用统计的方法得出所有用户对物品或者信息的偏好，然后发现与当前用户口味和偏好相似的邻居用户群[1]。基于模型的方法就是基于样本的用户喜好信息，训练一个推荐模型，然后根据实时的用户喜好的信息进行预测，计算推荐。基于模型的方法比基于用户的协同过滤方法更有扩展性。

推荐系统有两个潜在的问题。一是可扩展性，即一个推荐系统多快可以产生推荐，第二是改善给一个用户的推荐准度。纯协同过滤推荐系统能够比纯基于内容和基于人口统计学推荐系统产生更好的推荐效果，但是，由于冷启动问题，导致推荐效果的偏差。

本文提出一个混合方案，能进行更准确的预测和推荐，并用来解决冷启动问题。我们提出的方案是基于一个级联的混合推荐技术，其基于项目的评分，特征和人口信息构建项目模型。评估算法的数据集是 MovieLens 和 FilmTrust。

2. 相关研究

2.1. 基于项目的协同过滤推荐系统：项目评分信息

基于项目的协同过滤推荐系统用离线平台建立一个计算项目相似度的模型[2]。主要有以下三步：

1) 检索由活跃用户评价过的所有项目。

2) 使用检索的项目的集合计算目标项目的相似性。选择 k 个最相似的项目的集合，也称为具有它们的相似性的目标项目的邻居。

3) 通过计算对 k 个最相似的项目的活跃用户评价的加权和来进行对目标项目的预测。

2.2. 基于内容的推荐系统：项目特征信息

基于内容的推荐系统是基于项目的文本信息推荐项目。在这些系统中，感兴趣的项目由其相关联的特征定义，例如新闻过滤系统使用文本的词作为特征。

我们从 IMDB 下载关于电影的信息，并应用 TF-IDF 方法从关于每部电影的信息中提取特征[3]。我们构建了 IMDB 中的电影的关键字，标签，导演，男演员/女演员，以及用户评论的向量。此外，我们利用 WordNet 使用 Java WordNet Interface 克服特征之间的同义词问题，同时找到(文本)特征之间的相似性。

2.3. 基于人口统计学推荐系统：项目人口统计信息

人口统计学推荐系统是基于用户或项目的个人属性对其进行分类，并基于人口统计分类进行推荐。在我们的工作中，我们使用关于电影的类型信息作为其人口统计信息，并构造一个矢量。

3. 改进的算法

令 m_a， n_t，R，D，F 分别为活动用户，目标项目，用户项目评级矩阵，项目 n_j 的人口统计矢量和项目 n_j 的特征向量。让 R_{ISim}， D_{ISim} 和 F_{ISim} 表示项目之间的评级，人口统计信息和项目间的特征相似性。此外，令 R_{DSim}， D_{DSim} 和 F_{DSim} 分别表示在计算所有项目之后的特征相关性之后找到的候选项目之间的评级相似性，计算所有项目之后的特征相关性之后发现的候选项目之间的人口统计相似性，以及计算评分之后发现的候选项目之间的特征相似性。

提出的算法可以总结如下：

步骤 1：使用评级数据，人口统计数据和特征数据来计算项目之间的相似度，并存储该信息[4]。两个项目之间的修正的余弦相似性用于测量已经评过分的项目的相似度。两个项目之间的向量相似性用于测量使用人口统计和特征向量的相似性。

步骤 2：提高的相似性 $Boosted_{Sim}$ 由函数 $f_{\max}$ 定义，其将 R_{ISim}， D_{ISim} 和 F_{ISim}， R_{DSim}， D_{DSim} 和 F_{DSim} 组合在训练集中的项目集合上。此函数使用公式(5)进行预测。它可以规定如下：

$$f_{\max} = \arg\max_{f \in F} u\left(m_i, n_j\right) : \forall m_i \in M^T, \forall n_j \in N^T \tag{3}$$

等式(3)告诉我们选择使训练集中的项目集(N^T)上的所有用户(M^T)的效用最大化(即减少 MAE)的函数。表 1 给出了在训练集上检查的函数的不同组合以及观察到的它们各自最低的 MAE。这表明在施加特征相关性后发现的候选邻居项目中对评级和人口统计相关性进行了级联混合设置后给出了最小误差。

Table 1. A SAMPLE OF FUNCTION(F)WITH $k = 20$
表 1. $k = 20$ 的函数样本

函数号	函数(f)	MAE(ML)	MAE(FT)
1	R_{ISim}	0.793	1.443
2	F_{ISim}	0.788	1.437
…	…	…	…
32	$F_{ISim} + R_{DSim} + D_{DSim}$	0.736	1.378
…	…	…	…
83	$R_{ISim} + F_{ISim} + D_{ISim}$ $R_{DSim} + F_{DSim} + D_{DSim}$	0.835	1.452

令 $C_{n_t}=\{c_1,c_2,\cdots,c_k\}$ 是在应用特征相似性之后找到的 k 个候选邻居的集合[5]。我们通过在训练集中的项目集合上的 F_{ISim}， R_{DSim}， D_{DSim} 的线性组合来定义提高的相似性 $Boosted_{Sim}$，如下：

$$Boosted_{Sim}(n_t,c_i)=\alpha\times F_{ISim}+\beta\times R_{DSim}+\gamma\times D_{DSim} \tag{4}$$

公式(4)中， α， β 和 γ 参数表示相互影响的三个相似点。我们假设 $\alpha+\beta+\gamma=1$。

步骤 3：通过使用以下公式来预测目标项目 n_t 上的活动用户 m_a 的评分 P_{m_a,n_t}：

$$P_{m_a,n_t}=\frac{\sum_{i=1}^{k}\left(Boosted_{Sim}(n_t,c_i)\times r_{m_a,c_i}\right)}{\sum_{i=1}^{k}\left(\left|Boosted_{Sim}(n_t,c_i)\right|\right)} \tag{5}$$

4. 实验

4.1. 数据集

我们使用 MovieLens (ML)和 FilmTrust (FT)数据集用于评估我们的算法。MovieLens 数据集包含 943 个用户，1682 个电影和 100000 个评分记录。规模为 1 (差)至 5 (优)。MovieLens 数据集被用在很多研究项目中。这个数据集的稀疏性约 93.7% $\left(1-\frac{100000}{943\times1682}\approx0.937\right)$。

我们通过 FilmTrust 创建了第二个数据集。检索的数据集包含 1592 位用户，1930 电影和 28645 评级，规模为 1 (差)到 10 (优)。此数据集的稀疏度约为 99.06% $\left(1-\frac{28645}{1592\times1930}\approx0.9906\right)$。

4.2. 评价指标

我们在本文中的具体任务是预测已经被实际用户评分的项目的分数，并且检查该预测如何有助于用户选择高质量项目。考虑到这一点，我们使用平均绝对误差(MAE)。

MAE 测量推荐系统的预测评级和用户分配的真实评级之间的平均绝对偏差。其计算如下：

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}\left|r_{p_i}-r_{a_i}\right|}{N}$$

其中 r_{p_i} 和 r_{a_i} 分别是等级的预测值和实际值，N 是已经评分的项目的总数。对于 MovieLens 数据集，如果用户对其评分为 4 分或更高，则认为该项目很好，否则为差[6]。类似地，对于 FilmTrust 数据集，如果用户对其评分为 7 分或更高，则认为项目良好，否则为差。

此外，我们使用覆盖度来衡量推荐系统可以推荐多少项目。我们随机选择每个用户的 20%评级作为测试集，并使用剩余的 80%作为训练集。我们将训练集进一步细分为测试集和用于测量参数灵敏度的训练集。为了学习参数，我们通过随机选择在 80%的训练集上进行 5 重交叉验证，每次随机选择不同的测试和训练集合，并取结果的平均值。

我们将我们的算法与几种不同的算法进行比较：使用皮尔逊相似性的基于用户的协同过滤，基于项目的协同过滤使用修正的余弦相似度，朴素贝叶斯分类方法使用项目特征信息，用于生成推荐的朴素混合方法，用于进行概率推荐的个性诊断算法[7]。此外，我们调整了所有算法的参数。

4.3. 正选择邻居大小的最优值(k)

我们将活动用户的邻居数目从 0 改到 100，并计算 F_{ISim}， R_{DSim}， D_{DSim} 的相应 MAE。结果如图 1 所示：

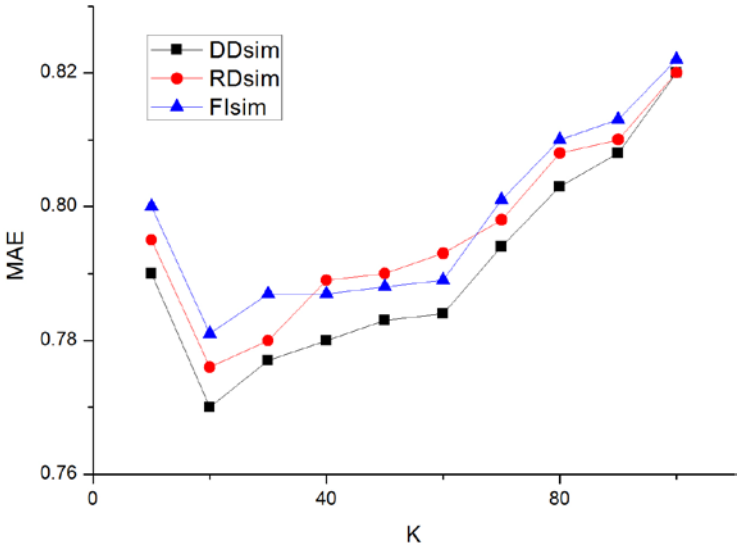

Figure 1. Determining the optimal value of neighbourhood size
图 1. 确定邻域大小的最优值 k

图 1 表示对于 MovieLens 数据集的 MAE 对于 $k = 20$ 是最小的。所以对于进一步的实验，我们选择邻域大小为 20。

4.4. 学习参数的最优值(α, β, γ)

通过产生参数值的所有可能组合产生 36 个参数集，范围从 0.1 到 1.0，差值为 0.1。表 2 给出了所学习的参数集的样本。参数设置 $\alpha = 0.5, \beta = 0.3, \gamma = 0.2, \alpha = 0.7, \beta = 0.2, \gamma = 0.1$ 分别在 MovieLens 和 FilmTrust 数据集下给出最低的 MAE。值得注意的是，组合相似性在很大程度上取决于特征相似性，即 α。此外，对于 MovieLens 和 FilmTrust 数据集，参数的值是不同的，这是由于这两个数据集具有不同的密度，评分分布和评级量表。

4.5. 改进的算法与其他算法的比较

1) MAE 方面的性能评估：图 2 显示我们的算法明显优于其他算法。对于 FilmTrust 数据集我们观察到类似的结果。我们可以从项目集合的结果中得出结论，应用 F_{ISim}， R_{DSim}， D_{DSim} 后对推荐结果具有互补的作用。

2) MAE，ROC 灵敏度，覆盖率与其他算法的在线成本的比较：表 3 显示了在线成本(在最坏情况下)每个算法具有的最低 MAE 和覆盖率。这里，P 是针对训练示例的特征的数量(即针对电影的特征)。值得注意的是，对于 FilmTrust 数据集，与 MovieLens 数据集相比，对于所有算法，ROC 灵敏度更高。我们认为这是由于评分分布的原因。此外在 FilmTrust 数据集下，算法的覆盖率低得多，这是由于其非常稀疏(99%)的原因[8]。该表描述了 $\text{Boosted}_{\text{DemoFeature}}$ 是可扩展和实用的，因为其在线成本小于或等于其他算法的成本[9]。

3) 新项目和新用户冷启动问题下的性能评估：当将新项目添加到系统时，则不可能从用户获得该项目的评分数据，因此协同过滤推荐系统不会推荐该项目。这个问题被称为新项目冷启动问题。为了在这种情况下测试我们的算法，我们选择 1000 个来自测试集的用户/项目对的随机样本[10]。在对目标项目进行预测时，训练集中已经对目标项目进行评分的用户的数量被保持为 1,2 和 5。在表 4 中，对应的 MAE 由 MAE1，MAE2，MAE5 表示。表 4 表明，所提出的方案在新项目冷启动中表现良好。

Table 2. A sample of parameter set with $k = 20$
表 2. $k = 20$ 的参数集样本

参数集号	α	β	γ	MAE (ML)	MAE (FT)
1	0.1	0.1	0.8	0.738	1.382
…	…	…	…	…	…
29	0.5	0.3	0.2	0.732	1.379
…	…	…	…	…	…
35	0.7	0.2	0.1	0.739	1.374
36	0.8	0.1	0.1	0.742	1.379

Table 3. A comparison of the proposed algorithm
表 3. 各种算法成本，精度，覆盖率比较

Algorithm	Online Cost	MAE (ML)	MAE (FT)	ROC (ML)	ROC (FT)	Coverage (ML)	Coverage (FT)
Userbased	NM^2	0.792	1.442	0.402	0.643	99.42	96.61
Itembased	N^2	0.791	1.441	0.384	0.623	99.22	92.31
$Boosted_{RDF}$	N^2	0.725	1.363	0.563	0.755	100	99.19
NaiveBayes	M (NP)	0.833	1.472	0.623	0.835	100	99.99
Naivehybrid	$NM^2 + M$	0.822	1.462	0.525	0.725	100	99.99
Personality diagnosis	NM	0.785	1.433	0.521	0.735	99.14	94.23

Table 4. Performance evaluation under new item cold-start problem
表 4. 冷启动下的性能评估

算法	MAE1		MAE2		MAE5	
	(ML)	(FT)	(ML)	(FT)	(ML)	(FT)
User-based CF	1.64	2.66	1.23	2.24	0.95	1.94
Item-based CF	1.36	2.55	1.19	2.15	0.91	1.58
$Boosted_{RDF}$	0.98	1.61	0.84	1.58	0.82	1.45

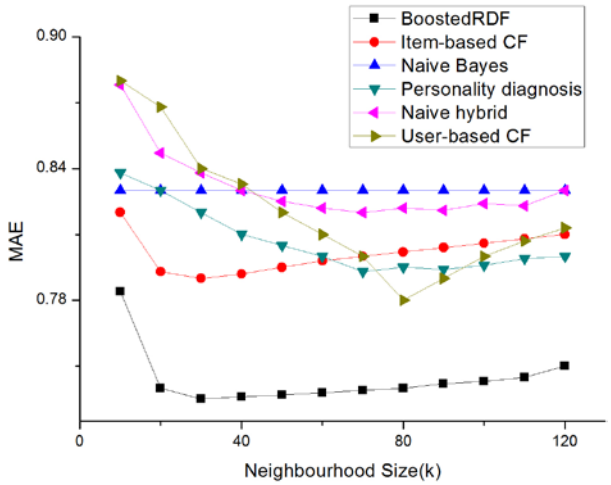

Figure 2. MAE of the proposed algorithm with others, against various neighbourhood sizes
图 2. 不同邻域下改进算法与其他算法的 MAE 对比

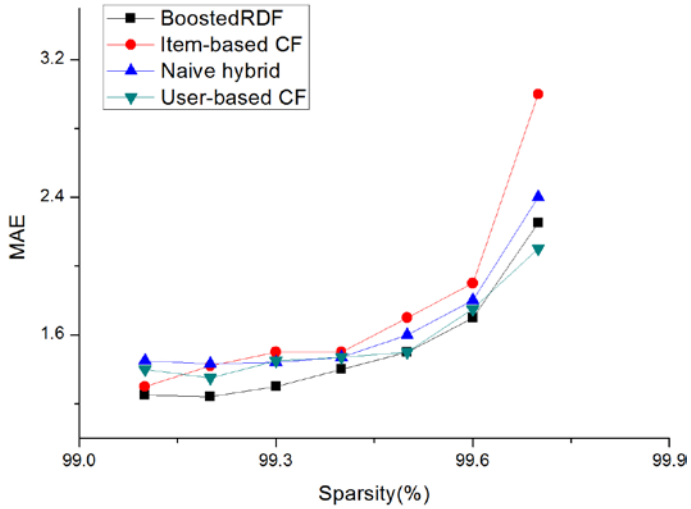

Figure 3. Performance of algorithms under different sparsity levels
图 3. 不同稀疏水平下的算法性能

对于新用户冷启动问题，其中用户的信息不完整，我们使用线性回归模型来找到一个项目的活动用户的评分的近似值[11]。我们使用的是此评分，而不是公式(5)中的活跃用户的实际评分预测生成的评分。

$$r'_{m_a,n_i}=\begin{cases} r_{m_a,n_i}: \text{活跃用户评分超过} J \text{部电影} \\ r_{reg}: \text{活跃用户小于等于} J \text{部电影} \end{cases} \tag{6}$$

在(6)中，J 的选择来自训练集，发现 MovieLens 为 10，FilmTrust 数据集为 5。评分 r_{reg} 通过线性回归模型：$R_s=\theta_1 R_t+\theta_2$，其中 R_s，R_t 是目标项目的向量和相似项目的向量。参数 θ_1 和 θ_2 可以通过两个相似矢量找到。这个模型用于克服基于项目的协同过滤中项目之间的误差。我们使用这种模型只有当我们有不完整的用户参数时才有意义。

4) 不同稀疏性下的性能评价：为了检查稀疏性的影响，我们通过丢弃一些随机选择的条目来增加训练集的稀疏度。但是我们保持每个稀疏训练集的相同的测试集。我们检查了提出的算法的性能与以纯用户为基础的协同过滤，基于项目的协同过滤和一个朴素的混合推荐算法。图 3 表示在所提出的不同算法的情况下，性能不会快速降低。这是因为项目的特征仍然可以用于找到类似的项目。此外，同义词检测算法了丰富项目特征，同时也可以找到项目之间的相似性。

5. 结束语

在本文中，我们提出了一个独特的层叠混合推荐方法，使用评级数据，人口统计数据和特征数据相结合的方法来计算项目之间的相似度。实验表明我们的方法优于传统的推荐算法。

参考文献 (References)

[1] Adomavicius, G. (2005) Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, **17**, 734-749. https://doi.org/10.1109/TKDE.2005.99

[2] Breese, J.S., Heckerman, D. and Kadie, C. (1998) Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Morgan Kaufmann, Burlington, Massachusetts, 43-52.

[3] Sarwar, B., Karypis, G., Konstan, J. and Reidl, J. Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the* 10*th International Conference on World Wide Web*, Hong Kong, 1-5 May 2001, 285-295. https://doi.org/10.1145/371920.372071

[4] Vozalis, M. and Margaritis, K. (2007) Using SVD and Demographic Data for the Enhancement of Generalized Collaborative Filtering. *Information Sciences*, **177**, 3017-3037. https://doi.org/10.1016/j.ins.2007.02.036

[5] Burke, R. (2002) Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, **12**, 331-370. https://doi.org/10.1023/A:1021240730564

[6] Pazzani, M.J. (1999) A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, **13**, 393-408. https://doi.org/10.1023/A:1006544522159

[7] Pennock, D., Horvitz, E., Lawrence, S. and Giles, C. (2000) Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based Approach. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, 2000, 473-480.

[8] Vozalis, M. and Margaritis, K. (2006) On the Enhancement of Collaborative Filtering by Demographic Data. *Web Intelligence and Agent Systems*, **4**, 117-138.

[9] Jonathan, L.G.T., Herlocker, L., Konstan, J.A. and Riedl, J.T. (2004) Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* (*TOIS*) *Archive*, **22**, 734-749.

[10] Lang, K. (1995) Newsweeder: Learning to Filter Netnews. *Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe City, California, 9-12 July 1995, 331-339. https://doi.org/10.1016/b978-1-55860-377-6.50048-7

[11] Melville, P., Mooney, R.J. and Nagarajan, R. (2002) Content-Boosted Collaborative Filtering for Improved Recommendations. *Eighteenth National Conference on Artificial Intelligence*, Edmonton, Canada 28 July-1 August 2002, 187-192.