文章编号：1007-1423(2016)07-0016-04 DOI：10.3969/j.issn.1007-1423.2016.07.004

协同过滤推荐算法比较研究

周泓宇[1]，梁刚[1]，杨进[2]

（1. 四川大学计算机学院，成都 610065；2. 乐山师范学院计算机科学学院，乐山 614000）

摘要：

推荐系统被广泛用于电子商务、视频网站、新闻小说推荐等各个领域，旨在向用户提供其可能感兴趣的信息。协同过滤是推荐系统的主流技术，分为基于内存的协同过滤、基于模型的协同过滤以及组合协同过滤。以其中基于用户（项目）、基于矩阵分解以及基于线性回归集成策略的协同过滤算法为例进行说明比较，然后通过 MovieLens 的数据集进行实验对比。

关键词：

推荐系统；协同过滤；线性回归；矩阵分解

基金项目：

四川省科技厅项目（No.2014JY0036）、四川省教育厅创新团队基金（No.13TD0014）

0 引言

随着信息技术的发展，网络数据规模急剧扩大，大数据满足用户对信息需求的同时，带来了信息过载的问题——用户难以从海量数据中快速找到有用的信息[1]。推荐系统从用户的角度出发，根据用户的需求偏好、行为记录等，挖掘出用户可能感兴趣的信息，并主动推荐给用户。

协同过滤是推荐系统中采用的最为广泛最为重要的技术之一[2]，其基本思想是具有相似行为的用户对项目的需求也是相似的[3]。协同过滤算法只关注用户的历史行为，不受项目具体属性的限制；并且能够与社会网络相结合，具有较好的推荐精度。其主要类型分为基于内存的协同过滤、基于模型的协同过滤以及组合协同过滤。本文首先介绍三种类型的协同过滤算法，以其中基于用户（项目）的协同过滤算法、基于矩阵分解的协同过滤算法以及基于线性回归的集成策略为例进行详细阐述，然后给出对比实验结果和分析，最后是全文总结。

1 基于用户（项目）的协同过滤算法

基于内存的协同过滤算法包括基于用户的方法和基于项目的方法，大致分为三个步骤：①基于用户-项目评分矩阵，计算用户（项目）之间的相似性；②通过相似度的逆序，选取最相似的前 K 个用户（项目）作为邻居；③根据邻居的评分，对目标用户（项目）未评分的项进行预测。下面以基于用户的协同过滤算法为例进行详细说明。

定义一个给定的用户集 U 和项目集 S，用户对项目的评分表示为一个 $m\times n$ 的矩阵 R，如表 1 所示。$R(i,j)$ 表示用户 i 对项目 j 的评分，代表用户对项目的偏好。如 MovieLens 数据集中用 1~5 分表示用户的喜爱程度；若 $R(i,j)=0$ 则表示用户 i 未对项目 j 打分。

表 1 用户-项目 m×n 阶评分矩阵R

	S_1	…	S_j	…	S_n
U_1	$R_{1,1}$	…	$R_{1,j}$	…	$R_{1,n}$
…	…	…	…	…	…
U_i	$R_{i,1}$	…	$R_{i,j}$	…	$R_{i,n}$
…	…	…	…	…	…
U_m	$R_{m,1}$	…	$R_{m,j}$	…	$R_{m,n}$

首先基于用户-项目评分矩阵计算用户之间的相似度，常用的相似度算法包括余弦相似度算法、修正的余弦相似度算法和 Pearson 相关相似度算法。本文采用余弦相似度算法，把用户的评分看作一个 n 维的评分向量，第 k 维的值表示对项目 k 的评分，设用户 u 与用户 v 的评分向量分别表示为向量 u 和 v，则用户 u 和用户 v 之间的相似度为：

$$\mathrm{sim}(u,v)=\cos(\vec{u},\vec{v})=\frac{\vec{u}\cdot\vec{v}}{\|\vec{u}\|\cdot\|\vec{v}\|} \tag{1}$$

选取与用户 u 相似度最大的 k 个用户作为邻居 $G(u)$，依据这 k 个邻居对目标项目 j 的评分，加权预测用户 u 对项目 j 的评分 $P_{u,j}$，如公式(2)所示。其中，$\overline{R_i}$ 表示用户 i 评分的均值，$R_{i,j}$ 表示用户 i 对项目 j 的评分。

$$P_{u,j}=\overline{R_u}+\frac{\sum_{i\in G(u)}^{n}\mathrm{sim}(u,j)(R_{i,j}-\overline{R_i})}{\sum_{i\in G(u)}^{n}\mathrm{sim}(u,j)} \tag{2}$$

基于项目的方法与基于用户的方法相似，通过计算两两项目的相似性得到目标项目的邻居，并通过项目邻居的评分预测用户对目标项目的评分。两种基于内存的协同过滤算法适用于不同的推荐环境，例如在电子商务中，项目的个数远小于用户的个数，且项目内容相比用户变动更少，因此基于项目的推荐方法有更优的时间复杂度和实时性；而在微博、新闻等方面的推荐则与之相反，使用基于用户的方法更好一些。

2 基于矩阵分解的协同过滤算法

基于模型的协同过滤算法通过对数据集的训练完成对系统复杂模式的识别，并基于学习模型对协同过滤任务做出智能预测，包括基于概率的算法、朴素贝叶斯算法、聚类算法和基于矩阵分解的算法等，其中基于矩阵分解的算法常用于对基于评分的推荐环境[4]。基于矩阵分解的推荐算法是将原本高维度的评分矩阵 $R_{M\times N}$ 分解成两个低维度矩阵 $P_{M\times D}$ 与 $Q_{D\times N}$ 的乘积，可将 $P_{M\times D}$、$Q_{D\times N}$ 分别看作用户和项目的隐含特征矩阵，其中 D 表示隐含特征个数[5]。通过多次迭代训练，使得 $P_{M\times D}Q_{D\times N}$ 不断地逼近评分矩阵 $R_{M\times N}$，最后得到最终的 P、Q 矩阵（$P\times Q\approx R$），以此来对未评分的项进行预测。下面以基础的矩阵分解推荐算法为例进行说明。

为使得 $P\times Q\approx R$，需求 $P\times Q$ 到 R 的最近距离，即使整个模型的损失最小，如式(3)所示，其中 K 为所有已评分项。

$$\min\sum_{(i,j)\in K}(R_{ij}-\sum_{d=1}^{D}P_{id}Q_{dj})^2 \tag{3}$$

通常采用梯度下降法来求最优值，定义 $E_{ij}=R_{ij}-\sum_{d=1}^{D}P_{id}Q_{dj}$，每次迭代对矩阵 P、Q 的更新如下：

$$P_{id}^{t+1}=P_{id}^{t}+2\alpha(E_{ij}Q_{dj}^{t}) \tag{4}$$

$$Q_{dj}^{t+1}=Q_{dj}^{t}+2\alpha(E_{ij}P_{id}^{t}) \tag{5}$$

其中，t 为迭代次数，α 为迭代步长。通过多次迭代得到最终的 P、Q 矩阵，具体算法如下：

```
输入：R
初始化：P、Q、D、α、steps
for 1 to steps: //迭代
    Err = (R - P * Q) * sign(R) //将 R 中未评分项剔除
    P = P + 2 * α* (Err * Q.T) //更新矩阵 P
    Q = Q + 2 * α* ( P.T * Err) //更新矩阵 Q
    Loss = sum(pow(Err, 2)) //损失值
    if Loss < limit_loss: //损失值小于某个阈值时退出
        break
return P * Q  //返回评分近似矩阵
```

基于矩阵分解的推荐算法关注整个评分矩阵的损失程度，更注重全局性，且空间复杂度较低。

3 基于线性回归的集成策略

基于线性回归的集成策略是将多个弱学习器通过某种线性组合，获得比单个弱学习器效果更好的强学习器的过程，线性系数由回归分析确定。将上述基于用户的方法、基于项目的方法和基于矩阵分解的方法看作三个弱学习器，分别表示为 $h_u(X)$、$h_s(X)$ 和 $h_v(X)$，强学习器 $H_w(X)$ 如式(6)所示。

$$H_w(X)=w_0+w_1h_u(X)+w_2h_s(X)+w_3h_v(X) \tag{6}$$

其中，w_0,w_1,w_2,w_3 为线性系数。便于标记，令 $h_0(X)=1$，$h(X)=[h_0(X),h_u(X),h_s(X),h_v(X)]$，$W=[w_0,w_1,w_2,w_3]^T$，则 $H_w(X)=h(X)\times W$。定义实际评分列向量为 $R(X)$，整个训练集上的误差平方和如式(7)所示。

$$J(W)=\sum_{x\in X}\sum(H_w(x)-R(x))^2 \tag{7}$$

为了使误差平方和 $J(W)$ 最小，本文采用最小二乘

估计法，即：

$$\hat{W}=(h(X)^{T}h(X))^{-1}h(X)^{T}R(X) \tag{8}$$

4 实验结果与分析

4.1 实验数据与评价标准

本实验采用 MovieLens100K 数据集，其中包含了 943 位用户对 1682 个项目的 100K 个评分，并将该数据集的 80%作为训练集，剩下 20%为测试集。采用平均绝对误差 MAE(Mean Absolute Error)作为指标，衡量推荐算法的优劣。设预测的用户评分集合为$\{p_1,p_2,\cdots,p_C\}$，对应的实际用户评分集合为$\{r_1,r_2,\cdots,r_C\}$，则平均绝对误差 MAE 定义如式(9)所示。

$$MAE=\frac{\sum_{i=1}^{C}|p_i-r_i|}{C} \tag{9}$$

4.2 实验步骤

为了说明邻居数 K 对基于用户的协同过滤算法 UBCF 和基于项目的协同过滤算法 IBCF 的影响，选取 K 值从 5 到 35 进行测试，如图 1 所示。从图 1 中可看出，基于用户的方法和基于项目的方法分别在 K 取 30 和 K 取 15 时达到最优效果，且前者的误差普遍大于后者。

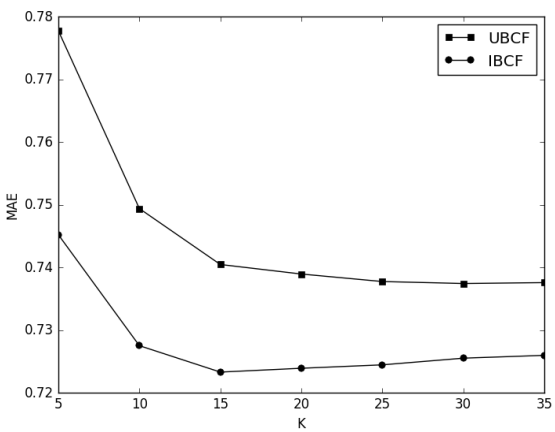

图 1 UBCF 和 IBCF 的 MAE 指标

为了说明隐含特征个数 D 对基于矩阵分解的协同过滤算法 MFCF 的影响，分别选取 10 个，20 个，30 个，50 个进行测试，如图 2 所示。从图 2 中可看出，在当前数据集下，MAE 值随着 D 的增加而增加，D 取 10 时，效果最佳。

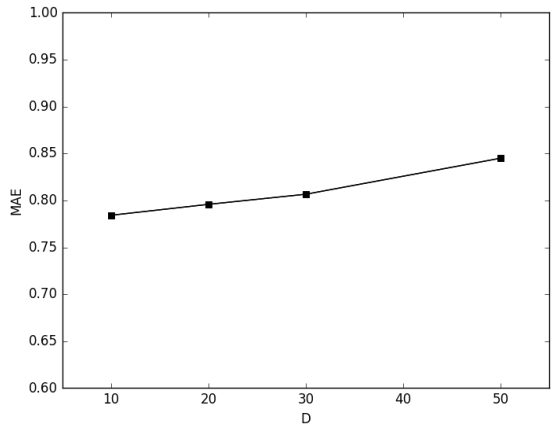

图 2 MFCF 的 MAE 指标

为了直观地比较基于线性回归的集成算法 LICF 与单个学习器算法的优劣，选取每个学习器的最优效果，即分别选择 UBCF(K=30)、IBCF(K=15)和 MBCF(D=10)作为弱学习器，并与集成后的强学习器作比较，如图 3 所示。从图中可看出，LICF 算法的效果明显优于每个弱学习器。

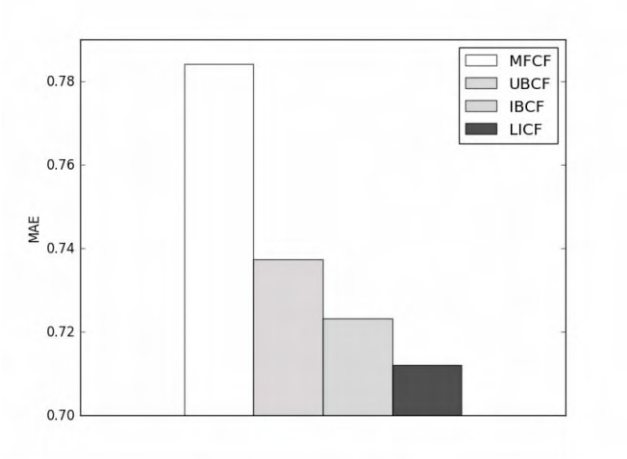

图 3 四种算法 MAE 指标的比较情况

5 结语

本文就三类协同过滤算法，针对性地对其中基于用户(项目)、基于矩阵分解和基于线性回归集成的方法进行了阐述和比较，然后利用 MovieLens 的电影评分数据对几种算法的推荐效果进行了验证。实验结果表明，基于线性回归的集成策略比其他单一的协同过滤算法效果好一些。组合推荐的优势明显，然而单个推荐算法的选择和组合策略的方式都是多样化的，如何找到最优的方案还有待进一步解决。

参考文献：
[1]柯良文，王靖. 基于用户特征迁移的协同过滤推荐[J]. 计算机工程，2015，41(1)：37-43.
[2]王国霞，刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用，2012，48(7)：66-76.
[3]Candillier L, Meyer F, Boulle M. Comparing State-of-the-Art Collaborative Filtering Systems[J]. Machine Learning and Data Mining in Pattern Recognition，2007，11(4)：548-562.
[4]Vucetic S, Obradovic Z. Collaborative Filtering Using a Regression-Based Approach[J]. Knowledge and Information Systems，2005，7(1)：1-22.
[5]王鹏. 基于矩阵分解的推荐系统算法研究[D]. 北京：北京交通大学.

作者简介：
周泓宇(1990-)，男，硕士，研究方向为机器学习
梁刚(1976-)，男，博士，讲师，研究方向为机器学习、智能计算、网络安全
杨进(1980-)，男，博士，教授，研究方向为机器学习
收稿日期：2016-01-15　　修稿日期：2016-02-20

Comparable Research on Collaborative Filtering Recommendation Algorithms

ZHOU Hong-yu[1], LIANG Gang[1], YANG Jin[2]

(1.College of Computer Science,, Sichuan University, Chengdu 610065;
2. College of Computer Science,, Leshan Normal University, Leshan 614000)

Abstract：
Recommender system designed to provide users with information that may be of interest is widely used in E-Commerce, video websites, news and novels recommendation, etc. Collaborative filtering is the main technology of recommender system, is divided into three classes: collaborative filtering based on memory, collaborative filtering based on the model and hybrid recommendation. Compares the algorithms of user(item) based, matrix factor based and linear regression in integration strategy as an example and gets the result through experimental comparison with the dataset of MovieLens system.
Keywords：
Recommender System; Collaborative Filtering; Linear Regression; Matrix Factorization