Computer Science and Application 计算机科学与应用, 2020, 10(4), 649-658
Published Online April 2020 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2020.104067

Hans汉斯

Collaborative Filtering Recommendation Algorithm Based on Matrix Decomposition and Meanshift Clustering

Haowei Deng, Shuxin Yang, Jiaqi Pei, Renyao Lin

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou Jiangxi
Email: *767617274@qq.com

Received: Mar. 22nd, 2020; accepted: Apr. 7th, 2020; published: Apr. 14th, 2020

Abstract

Scalability, sparseness of data and cold start of users are the main problems faced by traditional collaborative filtering recommendation algorithms. A collaborative filtering recommendation algorithm based on matrix decomposition and Meanshift clustering was proposed. Firstly, the original matrix was decomposed by singular value decomposition (SVD) method, and the original data would be better reduced. Then Meanshift clustering applied to all items, and finally combined the improved item-based collaborative filtering algorithm in the clustered categories to reduce the search range of neighbors. This method not only improves the recommendation speed, but also solves the user's cold start problem and data sparse problem properly. Compared with the traditional item-based collaborative filtering algorithm, the MAE value of this method on MovieLens 1M data set is reduced by 4.52%.

Keywords

Scalability, Matrix Decomposition, Meanshift Clustering, Collaborative Filtering, User Cold Start Problem, Data Sparsity Problem

基于矩阵分解和Meanshift聚类的协同过滤推荐算法

邓浩伟*，杨书新，裴嘉琪，林仁耀

江西理工大学信息工程学院，江西 赣州
Email: *767617274@qq.com

*通讯作者。

文章引用: 邓浩伟, 杨书新, 裴嘉琪, 林仁耀. 基于矩阵分解和 Meanshift 聚类的协同过滤推荐算法[J]. 计算机科学与应用, 2020, 10(4): 649-658. DOI: 10.12677/csa.2020.104067

收稿日期：2020年3月22日；录用日期：2020年4月7日；发布日期：2020年4月14日

摘　要

可扩展性、数据的稀疏性及用户的冷启动问题是传统的协同过滤推荐算法所面临的主要问题。由此提出一种基于矩阵分解和Meanshift聚类的协同过滤推荐算法：首先将原始矩阵使用奇异值分解(SVD)方法进行矩阵分解，较好地对原始数据进行降维，然后使用Meanshift (均值漂移)聚类对所有的物品进行聚类，最后在聚类后的类别中结合改进的基于物品的协同过滤算法，进而减少邻居商品的搜索范围。此方法不仅提高了推荐速度，还良好地解决了用户冷启动问题及数据稀疏问题，在MovieLens 1M数据集上相比于传统的基于物品的协同过滤算法MAE值最多下降了4.52%。

关键词

可扩展性，矩阵分解，Meanshift聚类，协同过滤，用户冷启动问题，数据稀疏问题

1. 引言

人们现在处在一个科技飞速进步的“大数据”时代，移动互联网技术的日新月异，不仅为人们的生活提供了极大的便利，同时也带来了海量的数据信息资源。然而面对如此庞大的信息资源，如何有效地利用这些资源自然成了近些年来研究者们研究的热点问题之一。而作为普通用户面对大量的数据时常常不知如何更好更快地选择自己所需要的信息，比如在音乐平台选择自己喜欢的类型的歌，在电子商务平台中选择想要购买的商品等。推荐系统应运而生，它在信息过滤、信息细化以及向用户提供个性化服务中发挥了显著作用，提供了一种崭新的信息服务模式[1]。推荐系统中目前较为主流的推荐算法是协同过滤推荐[2] (Collaborative Filtering, CF)。

面对用户的数量往往远远大于物品数量的现象，用户的冷启动问题[3]成为了一个比较严重的问题。杨秀梅[4]等在新闻推荐系统中提出基于用户上下文信息的方法，也改善了用户的冷启动问题，提升了用户的满意度。高玉凯[5]等提出了一种基于协同矩阵分解的用户冷启动推荐算法，来缓解推荐算法在用户冷启动上面临的情况。杨圩生[6]等使用基于信任环的用户冷启动推荐，不仅有效解决了用户冷启动问题，还提高了推荐的准确率。但是现在很多用户更倾向于操作简单的直接评分而非字字需要思考输入的评论，这就导致了大多数情况下我们并没有过多的相关数据可以进行分析。

数据的逐渐增多引起的可扩展性问题[7]也比较严重，因此国内外很多研究人员将聚类这一方法结合到协同过滤推荐算法中，来改善各种推荐算法的性能。Birtolo [8]等提出了一种基于模糊 C 均值的物品协同过滤推荐算法，实验证明有较好的推荐性。Sarwar [9]等提出了基于用户的 k-means 聚类协同过滤推荐算法，良好地改善了传统基于用户的协同过滤推荐算法的性能。邓爱林[10]等使用基于项目聚类的协同过滤算法，有效提高了推荐系统的实时响应速度。林建辉[11]等采用了基于 SVD 与模糊聚类的协同过滤推荐算法提高了推荐的质量。王伟[12]等通过 SVD 与 K-means 聚类结合的协同过滤算法来提升推荐效果。可是对于 k-means 聚类等一系列需要提前设定簇数 K 的聚类方法而言，如果簇的数量选择不当有时将有

可能严重影响聚类的效果。

同时在推荐系统中，所用到的数据往往是稀疏的，处理起来十分困难，众多研究人员研究表明数据填充和降维是缓解以上问题的有效方法。孙金刚[13]等基于项目的属性结合云填充技术解决了数据稀疏带来的相似性度量问题。高风荣[14]等通过划分稀疏矩阵，缩小推荐搜索范围的方法提升了传统协同过滤推荐算法的性能。

为了更好地解决以上用户冷启动、数据稀疏、可扩展性严重等问题，同时综合考虑到现实生活中的实际情况，我们提出一种基于矩阵分解和 Meanshift 聚类(均值漂移聚类)的协同过滤推荐算法。奇异值是矩阵的一个良好特征[15]。通过使用奇异值分解(SVD)进行矩阵分解可以在较好地保留原有数据特征的情况下，对数据进行一定程度上的降维，以达到节省时间及降低空间复杂度的目的。Meanshift [16]聚类是一种基于密度的聚类，Meanshift 聚类无需提前指定簇的数量，所以我们认为采用 Meanshift 聚类在面对未知的数据时是一种更合理的处理方法。同时算法采用了一种改进的欧几里得相似性度量方法进行物品间的相似性度量，最后在此基础上进行评分预测推荐。这使得整个推荐算法不仅可以对推荐结果有更好的解释性，而且更符合现实生活种的实际情况。

2. 相关工作

2.1. 问题定义

首先通过收集用户对物品的评分及评分时间 T_t，得到用户－物品评分矩阵 $\boldsymbol{R}_{m\times n}$，其中用户集合 $U=\{u_1,u_2,\cdots,u_m\}$ 表示共有 m 个用户，物品集合 $I=\{i_1,i_2,\cdots,i_n\}$ 表示共有 n 个物品，r_{ij} 表示的则是用户 u_i 对物品 i_j 的评分。用户－物品评分矩阵 $\boldsymbol{R}_{m\times n}$ 的具体表达形式如下所示：

$$\boldsymbol{R}_{m\times n}=\begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mn} \end{bmatrix}$$

2.2. 传统协同过滤算法

传统的协同过滤算法主要分为三大类，分别是于内容的协同过滤(Content-Based Collaborative-Filtering)、基于邻域的协同过滤(Neighborhood based Collaborative-Filtering)和基于模型的协同过(Model-based Collaborative Filtering)。其中基于邻域的协同过滤又分为基于用户的协同过滤(User-based Collaborative Filtering)和基于物品的协同过滤(Item-based Collaborative Filtering)。基于邻域的协同过滤的基本思想是通过在大量用户或物品中找到相近相似的用户或物品进行推荐。基于邻域的协同过滤推荐算法可将其过程分为四步：

(1) 构造用户－物品评分矩阵。

通过多种途径获取到每一个用户对每一样物品的评分，进而构造出用户－物品评分矩阵。

(2) 计算用户或物品的相似性，得到最近邻集合。

$sim(a,b)$ 表示的是物品 a 和物品 b 的相似度，其中 a，b 表示的是两种不同的物品，$R_{u,a}$ 和 $R_{u,b}$ 分别代表的是用户对于物品 a，b 的评分，$\overline{R}_a$ 代表的是物品 a 得到所有用户的评分的平均值，$\overline{R}_u$ 代表的是物品 b 得到的所有用户评分的平均值，$\overline{R}_u$ 代表的是用户 u 给予的所有评分的平均值。目前用于相似性度量的方法主要有四种，分别是：

欧几里得相似度：

$$sim(a,b)=\frac{1}{1+\sqrt{\sum(a_i-b_i)^2}} \tag{1}$$

其中 a_i 表示物品 a 在第 i 维的值。

余弦相似度(Cosine Similarity)：

$$sim(a,b)=\cos(\vec{a},\vec{b})=\frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\cdot\|\vec{b}\|} \tag{2}$$

修正余弦相似度(Adjusted Cosine Similarity)：

$$sim(a,b)=\frac{\sum_{u\in U}(R_{u,a}-\bar{R}_u)(R_{u,b}-\bar{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,a}-\bar{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,b}-\bar{R}_u)^2}} \tag{3}$$

皮尔森系数(Pearson correlation)：

$$sim(a,b)=\frac{\sum_{u\in U}(R_{u,a}-\bar{R}_a)(R_{u,b}-\bar{R}_b)}{\sqrt{\sum_{u\in U}(R_{u,a}-\bar{R}_a)^2}\sqrt{\sum_{u\in U}(R_{u,b}-\bar{R}_b)^2}} \tag{4}$$

假设若是使用基于物品的协同过滤算法，令目标物品为 i，$sim(i,i_q)$ 表示的是物品 i 与集合中物品 i_q 的相似性表达形式如公式(5)所示。计算完物品之间的相似性后可由若干个相似性较高的物品构成则需要搜寻的目标物品 i 的最近邻集合 $I=\{i_1,i_2,\cdots,i_q\}$,其中 $i\cap I=\varnothing$。

$$sim(i,i_q)(1\le q\le n) \tag{5}$$

(3) 预测评分。

根据目标用户或目标物品的最近邻集合计算出目标用户对目标物品的预测评分。对于用户 u 对物品 i 的评分预测，若使用基于物品的协同过滤算法进行评分预测，则其评分预测公式为如公式(6)所示。

$$\hat{r}_{u,i}=\bar{r}_i+\frac{\sum_{i_q\in I}sim(i,i_q)\times(\hat{r}_{u,i_q}-\bar{r}_{i_q})}{\sum_{i_q\in I}(|sim(i,i_q)|)} \tag{6}$$

其中 $\bar{r}_i$ 表示的是物品 i 收到的所有用户的平均评分，$\hat{r}_{u,i_q}$ 是用户 u 对物品 i_q 的评分。

若使用基于用户的协同过滤算法进行评分预测，则其评分预测公式为如公式(7)所示。

$$\hat{r}_{u,i}=\bar{r}_u+\frac{\sum_{u\in U'}sim(u,u_q)\times(\hat{r}_{u_q,i}-\bar{r}_{u_q})}{\sum_{u\in U'}(|sim(u,u_q)|)} \tag{7}$$

其中 $\bar{r}_u$ 和 $\bar{r}_{u_q}$ 分别表示的是用户 u 和用户 u_q 对所有物品的平均评分，U' 是用户 u 的最近邻集合。

(4) 进行 top-q 推荐。

在预测完评分的基础上选取 top-q 进行推荐。

3. 基于矩阵分解和 Meanshift 聚类的协同过滤推荐算法

3.1. 奇异值分解

在众多矩阵分解方法中，SVD 是常用的矩阵分解的方法之一，SVD 和特征值分解所必须要求的满秩的方阵不同，奇异值分解可以应用于任何实矩阵，因此其往往用于推荐算法中。因此我们可通过原始的用户－物品评分矩阵 $\boldsymbol{R}_{m\times n}$，利用 SVD 将其分解成三个矩阵，分别是左向量矩阵 $\boldsymbol{U}_{m\times m}$、对角矩阵 $\boldsymbol{K}_{m\times n}$、

右向量矩阵 $\boldsymbol{I}_{n\times n}^{\mathrm{T}}$。原始的用户－物品评分矩阵 $\boldsymbol{R}_{m\times n}$ 使用 SVD 进行矩阵分解的具体表达如公式（8）所示。

$$\boldsymbol{R}_{m\times n}=\boldsymbol{U}_{m\times m}\boldsymbol{K}_{m\times n}\boldsymbol{I}_{n\times n}^{\mathrm{T}} \tag{8}$$

使用 SVD 进行矩阵分解之后我们便可以获得具有重要信息的对角矩阵 $\boldsymbol{K}_{m\times n}$。接下来通过设置所要保留的特征的维度 d，其满足的条件是 $d\ll m$ 且 $d\ll n$。由此我们通过计算便可以获得物品－评分矩阵 $\boldsymbol{I}_{d\times n}$，计算的具体公式如公式(9)所示。

$$\boldsymbol{I}_{d\times n}=\boldsymbol{K}_{d\times d}\boldsymbol{U}_{m\times d}^{\mathrm{T}}\boldsymbol{R}_{m\times n}\boldsymbol{I}_{n\times n}^{\mathrm{T}} \tag{9}$$

通过公式(8)和公式(9)，我们不仅可以将特征的维度降至为 d，同时还可以获得更好的可以反应物品与评分之间关系的物品－评分的矩阵 $\boldsymbol{I}_{d\times n}$，这有利于后面我们使用改进的基于物品的协同过滤算法来更好的计算相似度并完成推荐任务。

3.2. Meanshift 聚类

面对海量的数据，随着用户或者项目的增多，推荐系统的响应速度会越来越慢，所以先采用聚类的方法后再进行推荐其可以减少邻居的搜索范围，进而提升响应速度。在聚类算法中，往往簇的数量是不好确定的，而且簇的大小也不均。但是 Meanshift 聚类具有不需要事先设定具体的簇数 K，只需要设定带宽 B 的特点，同时其具有很好的实时计算性。所以我们决定使用 Meanshift 聚类算法对数据进行聚类。

假设给出一个 k 维的空间，其中样本集合为 $X=\{x_1,x_2,\cdots,x_z\}$，x_z 为其中的第 z 个样本点。区域 $S_H(x)$ 是满足以下表达式(10)关系的 a 点的集合，其可以视为一个半径为 H，圆心为 x 的高维度球形区域。

$$S_H(x)=\left(a\,|\,(a-x)(a-x)^{\mathrm{T}}\le H^2\right) \tag{10}$$

μ 表示在这 z 个样本点中，有 μ 个样本点在区域 $S_H(x)$ 中。由此可以得出均值漂移向量的的基础定义公式为：

$$M_H(x)=\frac{1}{m}\sum_{x_z\in S_H}\left(x_z\frac{\delta y}{\delta x}-x\right) \tag{11}$$

为了使得区域中距离中心点越近的点拥有越大的权值，取得更好的聚类效果，我们使用高斯核函数，其在 Meanshift 聚类中的具体计算公式为：

$$N(x)=\frac{1}{\sqrt{2\pi}B}\mathrm{e}^{-\frac{x^2}{2B^2}} \tag{12}$$

结合公式(11)和公式(12)，我们可以得出最终引入了和函数的均值漂移向量的计算公式为：

$$M_H(x)=\frac{\sum_{z=1}^{p}x_zG\left(\left\|\frac{x-x_z}{H}\right\|^2\right)}{\sum_{z=1}^{p}G\left(\left\|\frac{x-x_z}{H}\right\|^2\right)}-x \tag{13}$$

其中 p 代表的是带宽 B 范围内点的数量，$G(x)$ 为当前高斯核函数公式(12)的导数的负值。

若以一组二维空间中的数据点为例，均值漂移聚类的步骤可以分为以下五步：

(1) 随机选取一点 x 为中心点，然后以 r 为半径画一个圆形，作为滑动窗口。圆形滑动窗口迭代地向更高密度区域去移动，直至收敛。

(2) 在迭代过程中，x 点通过每次迭代移向圆形区域内的均值点处移向更高密度的区域(即包含数据点

点数更多的区域)。

(3) 圆形滑动窗口一直不断移动，直到窗口中的数据点点数不再增加。

(4) 当有多个圆形滑动窗口出现重叠时，删除包含数据点点数较少的窗口。

(5) 根据数据点所处的滑动窗口进行聚类。

虽然聚类所消耗的时间较长，但是我们可以进行离线的聚类，并且保存聚类的结果，进而在接下来的相似性度量及推荐时可以保证以较快的速度完成相应的任务。

3.3. 改进的物品相似性度量

在传统的协同过滤算法中相似性度量都没有考虑到时间这一影响因素，只是将所有的物品的评价均等的看待，以此来衡量物品的相似性。而在现实生活中，针对一个物品的评价，近期的评分往往比往期的评分更重要，因为其更能反映出物品当前的状态，和用户对该物品的看法。所以我们针对物品之间的相似性度量，将时间因子考虑了进去。随着时间的推移，我们认为相似性表现为一种呈指数形式的衰减，衰减速度与当前推荐时的时间与之前物品被评分的时间的差值成正比，其表达式如公式(14)所示。

$$\frac{sim_{T_n}(a,b)}{sim_{T_a}(a,b)} = \mathrm{e}^{-w_t T_n} \tag{14}$$

最终我们采用的是一种改进的欧几里得相似性度量方法。其中 T_n 代表的是当前推荐时的时间，T_t 代表的是之前物品被评分的时间，w_t 代表的是指数衰减常数。结合公式(1)和公式(14)后，我们可以得到改进的欧几里得相似性度量方法的表达式如下所示：

$$sim_{T_n}(a,b) = \frac{1}{1+\sqrt{\sum(a_i-b_i)^2}}\mathrm{e}^{-w_t T_n} \tag{15}$$

通过离线的聚类之后我们只需在类别目标物品所在的类别中度量其与其他物品的相似性即可，然后选取特定数量的物品形成目标物品的最近邻集合，这有效缓解了算法的扩展性问题。同时，考虑到时间的影响不仅使得本算法可以更快速的完成推荐，还使得其对于推荐结果拥有了更好的解释性。

3.4. 物品评分预测

基于物品的协同过滤算法的基本思想是：同一用户因为其自身的特点，往往会造成其对于不同物品的评分具有一定的相似性，因此当我们需要预估某个用户对于某样物品的评分时，我们可以利用该目标用户对该目标物品的若干样相似物品的评分来预测目标用户对于目标物品的评分。

所以我们最后采用公式(6)进行评分预测。

3.5. 物品评分预测

算法过程如下：

输入：用户－物品评分矩阵 $\boldsymbol{R}_{m\times n}$，均值漂移聚类带宽 B,所要保留的特征的数目 d，推荐物品的数量 q，目标用户 u_i，目标用户 u_i 有过评价的物品 i_j，指数衰减常数 w_t。

输出：目标用户 u_i 对于物品 i_j 的预测评分，top-q 推荐物品。

步骤 1：对原始的用户－物品评分矩阵通过公式(8) (9)进行奇异值分解处理，获得物品－评分矩阵。

步骤 2：离线的对物品－评分矩阵通过使用公式(13)迭代的完成均值漂移聚类，获得各物品之间聚类后的结果，并进行保存。

步骤 3：根据公式(15)改进的欧几里得相似性度量方法计算物品 i_j 与其所属类别中的物品的相似度，

获得得物品 i_j 的最近邻集合。

步骤 4：在最近邻集合的基础上，利用公式(6)基于物品的评分预测方法对物品 i_j 进行评分预测。

步骤 5：对前 q 样物品进行 top-q 推荐。

4. 实验及结果分析

4.1. 实验环境

实验环境的配置如表 1 所示。

Table 1. Lab environment
表 1. 实验环境

硬件/软件	版本/配置
OS	Windows10, 64 位
Python	3.7
处理器	Intel(R) Core(TM)i7-7500 CPU @ 2.7 GHz 2.90 GHz
RAM	8 GB

4.2. 实验数据集

本实验采用的是 MovieLens 1M 数据集(http://fifles.grouplens.org/datasets/movielens/)，MovieLens 是著名的推荐算法数据集之一，其中含许多用户对多部电影的评分评级、用户属性和电影标签等信息。MovieLens 1M 数据集共包括 6040 位用户和 3883 部电影以及 1,000,209 条评分，用户对电影的评分的范围为 1~5，共分为 5 个不同的评级。综上所述，经计算得到 MovieLens 1M 的数据密度为：

$$\frac{1000209}{6040\times 3883}\times 100\% \approx 4.26\%$$

我们可以看出，MovieLens 1M 的数据密度比较小。

4.3. 评价指标

推荐算法中用于评价预测评分的评价指标比较著名的是平均绝对误差(MAE)。如果在测试集 T 中，令 r_{ij} 是用户 u_i 对物品 u_i 的真实评分，$\hat{r}'_{i,j}$ 是用户 u_i 对物品 i_j 的预测评分，$|T|$ 为测试集中样本的数目。MAE 采用绝对值来计算预测评分的误差，MAE 定义如公式(16)所示：

$$\mathrm{MAE}=\frac{\sum_{u,i\in T}\left|r_{i,j}-\hat{r}'_{i,j}\right|}{|T|} \tag{16}$$

在实验过程中 *MAE* 的值越小表明预测评分更接近真实评分。说明预测的精度更高。

除以上评价预测评分外，推荐算法中关于推荐结果往往可以通过召回率(Recall)来进行度量，召回率是指在推荐算法的推荐集合与测试样本的评分记录集合重合的部分与测试样本的评分记录集合的比值。假设对于某一位用户来说，M_r 为测试样本的评分记录集合，M_t 为使用推荐算法产生的推荐集合，则召回率的计算方法如公式(17)所示：

$$\mathrm{Recall}=\frac{\left|M_r\cap M_t\right|}{\left|M_r\right|} \tag{17}$$

在实验过程中，召回率的值越大则表明算法的推荐效果越好。

4.4. 实验设置

在本次实验中为了获得更好的实验效果，我们将整个数据集分为两部分，其中 70%的数据集作为训练集，30%的数据集作为测试集。通过前期的预实验我们在本次实验中将均值漂移聚类的带宽设定为 2.2，w_t 设置为 $\frac{1}{45}\ln 4$，保留的特征设定为原始特征的 35%，即舍弃掉 65%的原始特征。对于推荐数量我们依次设定为 5，10，15，20，25，30 进行实验，每个算法的实验次数共计六次。最后通过以上实验设置进行实验并使用上面给出的 MAE 和 Recall 评价指标进行实验评价。

同时为了展示本文所提出的新算法的算法性能，将本文所提出的基于矩阵分解和 Meanshift 聚类的协同过滤推荐算法(MMCCF)与传统的基于物品的协同过滤算法[17] IBCF (Item-based Collaborative Filtering)和基于 SVD 的物品协同过滤算法 SIBCF (SVD-item-based Collaborative Filtering)进行实验对比。

4.5. 实验结果分析

通过对比实验中，不同推荐数量引起的 MAE 变化我们可以由图 1 很明显的看出：随着推荐数量的增多，当我们使用 SIBCF 进行推荐实验时，MAE 的值会先变小再变大，在推荐数量增多的时候呈上升的趋势。与 SIBCF 不同的是，IBCF 和本文所提出的 MMCCF 在推荐数量增多时，MAE 值始终越来越小，呈下降的趋势，说明预测的精度越来越高。不过对比 IBCF 和本文所提出的 MMCCF，我们通过观察图 1 可以发现，在推荐数量较少时，通过 MAE 对算法进行评价，MMCCF 的预测效果要优于 IBCF。

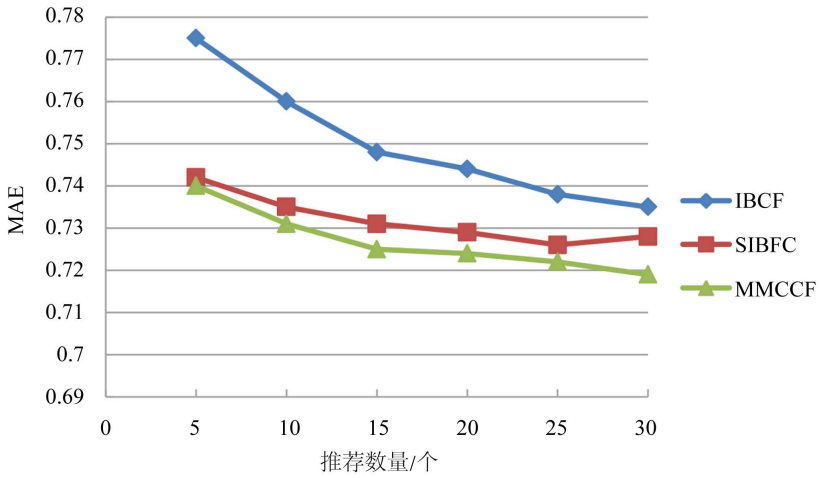

Figure 1. The relationship between the recommended number of different algorithms and MAE
图 1. 不同算法的推荐数量与 MAE 的关系

为了更直观的看出三种算法的性能对比，在不同的推荐数量下，本文提出的 MMCCF 对比 IBCF 和 SIBFC 改善的 MAE 值的百分比如表 2 所示。我们可以看出对于 IBCF，使用 MMCCF 最多使得 MAE 下降了 4.52%，对于 SIBCF，使用 MMCCF 最多使得 MAE 下降了 1.24%。

同时从实验中得出的图 2 中我们可以发现，在使用召回率对三种算法进行推荐结果评价时候，三种算法随着推荐数量的增加召回率都处在一种波动状态，但整体趋势是上升的这说明三种算法在一定程度上都具有比较良好的推荐效果，可以较好的完成推荐。通过图 2 我们可以清晰看出本文提出的 MMCCF 算法对比于 IBCF 和 SIBFC 两种算法的推荐效果在实验结果中更好。

Table 2. MMCCF improvement results on MAE values under different recommended quantities
表 2. MMCCF 在不同推荐数量下关于 MAE 值的改善结果

推荐数量/个	IBCF	SIBCF
5	4.52%	0.27%
10	3.82%	0.54%
15	3.07%	0.82%
20	2.69%	0.69%
25	2.17%	0.55%
30	2.18%	1.24%

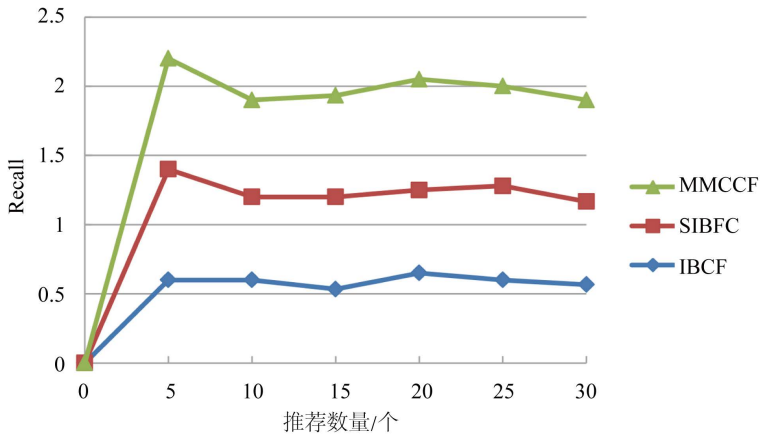

Figure 2. The relationship between the recommended number of different algorithms and Recall
图 2. 不同算法的推荐数量与 Recall 的关系

通过多次实验及实验对比后我们可以发现，本文所提出的新算法 MMCCF 通过以上评价指标进行评价，其性能在一定程度上都优于 IBCF 和 SIBFC。这证明了本文所提出的基于矩阵分解和 Meanshift 聚类的协同过滤推荐算法具有一定的优势，可以较好的完成预测评分与物品推荐，并且提升推荐的质量。

5. 结束语

传统的协同过滤算法，往往不能有效地处理数据稀疏问题和用户冷启动问题，同时推荐时间也较长，可扩展性问题严重。我们所提出的这种基于矩阵分解和 Meanshift 聚类的协同过滤推荐算法，通过矩阵分解中 SVD 方法重新构造出一个低维度的物品－评分矩阵，然后结合带有高斯核函数的均值漂移聚类对物品进行聚类，进而缩小物品搜索的范围，聚类后针对目标物品只需在其同一类中使用改进的欧几里得相似性度量方法进行相似度度量，最后进行完评分预测后即可更好更快地完成推荐任务，同时本文所提出的算法也在一定程度上有效缓解了用户冷启动和数据稀疏的问题。

基金项目

本项目受国家自然科学基金(No. 61662028)、国家级大学生创新创业训练计划项目基金(No. 201810407018)和江西省教育厅科学技术研究项目基金(No. GJJ170518)资助。

参考文献

[1] 翁小兰, 王志坚. 协同过滤推荐算法研究进展[J]. 计算机工程与应用, 2018, 54(1): 5-31.

[2] 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.

[3] 乔雨, 李玲娟. 推荐系统冷启动问题解决策略研究[J]. 计算机技术与发展, 2018, 28(2): 83-87.

[4] 杨秀梅, 孙咏, 王美吉, 等. 新闻推荐系统中用户冷启动问题的研究[J]. 小型微型计算机系统, 2016, 37(3): 479-482.

[5] 高玉凯, 王新华, 郭磊, 等. 一种基于协同矩阵分解的用户冷启动推荐算法[J]. 计算机研究与发展, 2017, 54(8): 1813-1823.

[6] 杨圩生, 罗爱民, 张萌萌. 基于信任环的用户冷启动推荐[J]. 计算机科学, 2013, 40(11): 363-366.

[7] 李聪. 电子商务协同过滤可扩展性研究综述[J]. 现代图书情报技术, 2010(11): 37-44.

[8] Birtolo, C. and Ronca, D. (2013) Advances in Clustering Collaborative Filtering by Means of Fuzzy C-Means and Trust. *Expert Systems with Applications*, **40**, 6997-7009. https://doi.org/10.1016/j.eswa.2013.06.022

[9] Sarwar, B.M., Karypis, G., Konstan, J., *et al*. (2002) Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. *Proceedings of the International Conference on Computer and Information Technology*, Hong Kong, 158-167.

[10] 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.

[11] 林建辉, 严宣辉, 黄波. 基于 SVD 与模糊聚类的协同过滤推荐算法[J]. 计算机系统应用, 2016, 25(11): 156-163.

[12] 王伟, 杨宁, 李丽华, 等. 基于 SVD 的 K-means 聚类协同过滤算法[J]. 微计算机信息, 2012, 28(8): 139-141.

[13] 金刚, 艾丽蓉. 基于项目属性和云填充的协同过滤推荐算法[J]. 计算机应用, 2012, 32(3): 658-660.

[14] 高风荣, 杜小勇, 王珊. 一种基于稀疏矩阵划分的个性化推荐算法[J]. 微电子学与计算机, 2004, 21(2): 58-62.

[15] 罗小桂, 河雁. 矩阵奇异值分解在计算技术中的应用[J]. 计算机与现代与现代化, 2006(6):67-68.

[16] Comaniciu, D. and Meer, P. (2002) Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 603-619. https://doi.org/10.1109/34.1000236

[17] Sarwar, B., Karypis, G., Konstan, J., *et al*. (2001) Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the* 10*th International Conference on World Wide Web*, Shanghai, 285-295. https://doi.org/10.1145/371920.372071