

基于矩阵分解与用户近邻模型的协同过滤推荐算法

杨 阳*, 向 阳, 熊 磊

(同济大学 电子与信息工程学院, 上海 201804)

(* 通信作者电子邮箱 yyai326@gmail.com)

摘 要: 针对个性化推荐系统中协同过滤算法面对的矩阵稀疏和新使用者问题, 提出基于矩阵分解与用户近邻模型的推荐算法。通过对用户档案信息构建近邻模型以保证新使用者预测的准确性; 同时考虑到数据量大和矩阵稀疏会引起时间和空间复杂度过高等问题, 引入奇异值矩阵分解的方式, 从而减小矩阵稀疏和数据量大的影响, 提高推荐系统的准确性。实验结果表明, 该算法能有效解决大数据量的矩阵稀疏问题以及新使用者问题。

关键词: 协同过滤; 矩阵分解; 用户近邻模型; 电子商务; 推荐算法

中图分类号: TP181 **文献标志码:** A

Collaborative filtering and recommendation algorithm based on matrix factorization and user nearest neighbor model

YANG Yang*, XIANG Yang, XIONG Lei

(College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

Abstract: Concerning the difficulty of data sparsity and new user problems in many collaborative recommendation algorithms, a new collaborative recommendation algorithm based on matrix factorization and user nearest neighbor was proposed. To guarantee the prediction accuracy of the new users, the user nearest neighbor model based on user data and profile information was used. Meanwhile, large data sets and the problem of matrix sparsity would significantly increase the time and space complexity. Therefore, matrix factorization was introduced to alleviate the effect of data problems and improve the prediction accuracy. The experimental results show that the new algorithm can improve the recommendation accuracy effectively, and solve the problems of data sparsity and new user.

Key words: collaborative filtering; matrix factorization; user nearest neighbor model; E-commerce; recommendation algorithm

0 引言

近年来,随着互联网的全面普及与电子商务的逐步完善,推荐系统已经成为了一个越来越受到关注的研究领域。目前,几乎所有的电子商务、在线视频音乐网站都使用了推荐系统,亚马逊推荐系统引擎的开发者 Linden 等^[1]指出,在 2002 年超过 20% 的亚马逊销量是由其个性化推荐系统所产生的。也有许多非商业目的的推荐系统,如个性化的网页、社交网络会员、维基百科推荐等。

推荐系统分为两种不同的方法:1) 基于内容过滤。通过分析物品的属性,比如手机的内存大小、屏幕材质等和用户的性别、偏好、年龄等内容来给用户推荐与之匹配的商品。2) 协同过滤(Collaborative Filtering, CF)。根据用户偏好的相似性来推荐资源,把和当前用户相似的其他用户的意见提供给当前用户。目前,有许多协同过滤的实现方法,如基于 K 邻居模型的方法^[2]、奇异值矩阵分解^[3]和基于图论的模型^[4]等。文献[5-6]中列出了不同协同过滤算法的比较。当系统拥有充足的用户偏好信息时,协同过滤技术有着良好的推荐质量,但随着信息量的不断增加,协同过滤也面临着稀疏性、冷启动、新使用者等问题。例如,当一个新用户刚刚注册之后,并没有足够的历史信息来找出这个用户的兴趣偏好,这

时协同过滤算法的预测效果往往就不那么准确。然而当一个用户拥有足够多的历史信息之后,通过协同过滤对于此用户的预测结果会有明显的提高。随着互联网行业竞争日益加剧,为吸引用户留给用户的第一印象就显得十分重要,因此如果能够有效解决新使用者问题,在新用户刚注册网站时就给出准确的预测与推荐,将可能留住更多的用户群体。有了大量用户数据,用户与物品之间的评分矩阵往往就会越发地稀疏,因此从稀疏矩阵中获取有用的历史信息来取得进一步精确的推荐也十分关键。

本文围绕解决上述问题展开研究,并在已有研究的基础上,基于矩阵分解和用户近邻模型,提出了一种既能够解决系统新使用者,又能够通过应用协同过滤来达到准确推荐的推荐算法。它首先通过判断预测用户兴趣的历史信息是否完备,然后分别应用用户近邻模型和矩阵分解的方式来达到准确预测用户喜好的目的。实验表明,该算法能有效地解决评分矩阵稀疏问题和新使用者问题,提高了推荐精度。

1 问题定义及相关工作

首先,本文分别使用 u, v 表示用户, i, j 表示物品, r_{ui} 表示用户 u 对物品 i 的已有评价分数, \hat{r}_{ui} 表示用户 u 对物品 i 的预测评价分数,分数越高表示该用户越喜欢该物品。用户评分的

收稿日期: 2011-07-14; 修回日期: 2011-09-18。

基金项目: 国家自然科学基金资助项目(61103069, 71171148, 70771077); 上海市信息化发展专项资金项目(200901015)。

作者简介: 杨阳(1987-), 男, 福建福州人, 硕士研究生, 主要研究方向: 数据仓库、数据挖掘、协同过滤; 向阳(1962-), 男, 重庆人, 教授, 博士生导师, 博士, 主要研究方向: 决策支持系统、人工智能; 熊磊(1989-), 男, 江西丰城人, 硕士研究生, 主要研究方向: 数据仓库、数据挖掘、协同过滤。

所有记录可以看成是一个横坐标为用户、纵坐标为物品的评分矩阵 R , 其中包括 m 个用户的集合 $U = \{User_1, User_2, \dots, User_m\}$ 和 n 个物品的集合 $I = \{Item_1, Item_2, \dots, Item_n\}$, 如表1所示。表1中已知值是某一用户已经对某一物品评过多少分, 缺失值表示该某一用户未对某一物品进行过评分。推荐算法所要做的工作是基于已有的评分值和相关信息, 对矩阵的缺失值进行预测。

表1 用户—物品评分表

用户	物品			
	$Item_1$	$Item_2$	\dots	$Item_n$
$User_1$	r_{11}	r_{12}	\dots	r_{1n}
$User_2$	r_{21}	r_{22}	\dots	r_{2n}
\vdots	\vdots	\vdots	\vdots	\vdots
$User_m$	r_{m1}	r_{m2}	\dots	r_{mn}

1.1 基准偏移量

在用户评分矩阵中, 存在着个人基准偏好这一因素, 就是说有一些用户的打分记录普遍比其他用户高, 有些物品得到的分数普遍比其他物品高。由此文献[7]提出使用基础偏移量 b_{ui} 来表示用户或物品偏好:

$$b_{ui} = u + b_u + b_i \quad (1)$$

其中: b_u 和 b_i 分别表示对用户 u 和物品 i 的偏好, u 表示整个评分矩阵中用户对物品打分的平均分。例如: 要预测用户 A 对电影《搏击俱乐部》打分, 平均分为 3.5 (分数范围 1~5), 电影《搏击俱乐部》好于其他一般的电影 0.8 分, 另一方面用户 A 是一个要求高的用户, 他一般偏向于给电影打低 0.3 分, 所以 $b_{ui} = 3.5 + 0.8 - 0.3 = 4$ 。

1.2 矩阵分解

矩阵分解是协同过滤算法中一种十分有效的方法, 它应用不同的数学或机器学习方法来从用户—物品的打分矩阵中分解出潜在特征来解释并预测打分。其中包括概率潜在语义分析 (probabilistic Latent Semantic Analysis, pLSA)^[8], 神经网络^{[4]796} 和潜在狄利克雷分布 (Latent Dirichlet Allocation, LDA)^[9], 但都存在对稀疏矩阵的预测不准和可扩展性低的问题。本文中将应用矩阵奇异值分解 (Singular Value Decomposition, SVD) 方法在用户—物品评分矩阵上。近来, 矩阵奇异值分解方法受到越来越多的关注, 它不仅具有可扩展性, 还有良好的预测准确程度。它用 $r_{ui} = p_u \times q_i^T$ 来表示预测分数, 其中 p_u 向量表示用户 u 的潜在 f 维特征向量, 向量 q_i 表示物品 i 的潜在 f 维特征向量。

SVD 方法常常应用在信息获取 (Information Retrieval, IR) 领域^[9]。但是传统的 SVD 方法应用在 CF 领域内, 会遇到大量值缺失、矩阵不完整的问题。很多解决方法^[11-12] 都是通过估计、插入这些缺失值来使矩阵密度达到要求。然而随着矩阵规模的不断增加, 这些缺失值插入的方法的效率将会越来越低。因此文献[13-14]等提到直接使用用户—物品评分已有记录来进行 SVD。本文中使用了一种改进的增量奇异值矩阵分解的方式, 通过随机梯度下降法来对用户—物品评分矩阵进行分解, 能够有效地解决矩阵稀疏的问题。

1.3 用户近邻模型

近邻模型是协同过滤算法中最常用的模型, 它的中心思想是通过寻找 k 个近邻来模拟主体的行为。在协同过滤算法中, 按照用户和物品可以分为基于用户的近邻模型和基于物品的近邻模型。在用户近邻模型中主要有两个问题: 1) k 个邻居的选取问题; 2) 每个邻居与主体之间的相似程度。为解决邻居选取的问题, 先要选择一种方式来表示用户之间的距离关系。通常两个用户之间的邻近程度可以有如下几种方式表

示^[6]:

1) 余弦相似性: 设用户 u_1 和用户 u_2 同时对 n 个物品进行打分, 用两个 n 维的向量 n_1 和 n_2 表示, 通过计算

$$\text{sim}(u_1, u_2) = \cos(n_1, n_2) = \frac{n_1 \times n_2}{|n_1| \times |n_2|} \quad (2)$$

得到两个用户的余弦相似性。

2) 皮尔逊相关系数: 同样是用两个 n 维向量表示用户 u_1 和 u_2 的打分记录, 通过计算

$$p(u_1, u_2) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (3)$$

$$\text{sim}(u_1, u_2) = \frac{n}{n + \lambda} \times p(u_1, u_2) \quad (4)$$

得到两个用户的相关系数, 其中: $\text{sim}(u_1, u_2)$ 表示用户 u_1 与用户 u_2 的相似程度; λ 为参数, $\frac{n}{n + \lambda}$ 是为了保证两个用户同时对越多的物品评分, 则他们计算出来的相似度偏高, 相反两个用户同时评分的物品越少, 相似度偏低。根据不同的实际情况, λ 的值可做修改。本文 λ 取 30。

另外文献[15-16]中也提到了许多改进的相似度计算方法。但是以上这些方法所用到的相似度计算都是在用户已有的打分记录的基础上进行的, 在遇到一个新用户时, 会出现没有邻居可以匹配的情况。文献[17]通过对用户档案的相似度建模, 在进行协同过滤前预先填充用户—物品的评分矩阵。本文中将通过用户档案来计算用户之间的皮尔逊相关系数选取当前用户的 k 个邻居, 再通过机器学习的方式调整 k 个邻居与当前用户的近邻关系, 最后达到精确推荐的目的。在这一过程中, 不论是新老用户都可以找到与之相关的 k 个邻居, 有效克服了协同过滤算法中新使用者的问题。

2 用户档案的近邻模型

一般来说, 推荐系统除了包含用户—项目的评分信息之外, 还有记录注册用户信息和记录物品信息的数据集合。文献[18]中提到 80% 的用户愿意向 Web 站点提供自己的性别、教育背景和兴趣等不太敏感信息, 而且目前越来越多的网站都提供谷歌、雅虎、腾讯等账号登录服务, 可以提供更多更完善的用户档案信息。这些事实驱动本文考虑利用已有的用户档案信息来对传统的协同过滤算法进行改进, 所以本章将介绍利用用户档案资料来选择近邻以及插入权重法来计算近邻与主体之间的相似度, 公式^[7]如下所示:

$$\hat{r}_{ui} = b_{ui} + \sum_{v \in R_k(u)} (r_{vi} - b_{vi}) \times w_{uv} \quad (5)$$

其中: $R_k(u)$ 表示用户 u 的 k 个邻居的集合; b_{ui} 是上文中所提到的基准偏移量; w_{uv} 是插入的权重, 表示在基准偏移量之上的偏移量, 而 $(r_{vi} - b_{vi})$ 表示该偏移量的系数。例如, 当 u 和 v 相关程度 w_{uv} 比较高, $(r_{vi} - b_{vi})$ 也比较高时, 表示实际值比预测偏移量高出很多, 就将增加 $(r_{vi} - b_{vi}) \times w_{uv}$ 较多的偏移量到基准偏移量中; 而且当 $(r_{vi} - b_{vi})$ 接近 0 时, 表示基准偏移量已经达到一定的精确程度, 将不会添加过多的偏移量到基准偏移量中。在式(5)中, 需要考虑以下两个主要问题:

1) $R_k(u)$ 的选取。在预测用户 u 对物品 i 的评分时, 查找已经对物品 i 打分的所有用户集合 $R(u)$; 然后需要计算用户 u 与集合中每个用户 v 的相似度, 来选取最相似的 k 个近邻, 相似度通过用户档案资料来计算。假设每个用户有 m 维的属性信息, 其中包括性别、年龄、地区、收入等。首先, 将每维属性信息预先处理成数字型表示法, 例如性别: $\{0, 1\}$; 年龄: 每 5 岁

为一个增量,0 ~ 5 为 0,6 ~ 10 为 1 等。通过数据挖掘中的相关数据处理方法后,可以得到用户 u 的一个 m 维档案信息资料(v_1, v_2, \dots, v_m)。运用上一章中提到的皮尔逊相关系数的方法(式(4)),得到所有的相关系数后,通过选取前 k 个最相关的用户集合 $R_k(u)$ 作为用户 u 的 k 个近邻。

2) w_{uv} 的计算。一般来说许多方法将 $\text{sim}(u, v)$ 直接作为 w_{uv} , 当成用户 u 与 v 之间的评分关系, 由于本文中使用的用户档案之间的相似度计算, 将它直接作为预测用户 u 的评分依据时, 会遗漏 u 和 v 之间对于物品评分的历史记录相似信息, 所以本文方法中加入判断: 如果用户 u 存在历史评分记录时, 就将使用随机梯度下降来训练 w_{uv} 来表示用户 u 与 v 之间的评分关系; 如果用户 u 作为一个新用户时, 用户 u 与 v 之间并没有相关的历史评分记录, 这时就将使用 $\text{sim}(u, v)$ 直接作为 w_{uv} 来计算 u 与 v 之间的评分关系。这样做既能考虑到用户的档案信息和历史评分记录, 又能较好地解决新使用者的问题。

要计算 w_{uv} , 通过已有的评分记录来训练求得误差最小的 w_{uv} 。首先定义误差公式:

$$\sum \left(r_{ui} - u - b_u - b_i - \sum_{v \in R_k(u)} (r_{vi} - b_{vi}) \times w_{uv} \right)^2 + \lambda_1 (b_u^2 + b_i^2 + \sum_{v \in R_k(u)} w_{uv}^2) \quad (6)$$

其中 $\lambda_1 (b_u^2 + b_i^2 + \sum_{v \in R_k(u)} w_{uv}^2)$ 为惩罚因子, 是为了避免数据量不足和训练过拟合的现象。要最小化式(6), 分别对 b_u, b_i, w_{uv} 求偏导, 通过随机梯度下降法, 针对每一个评分数据都做如下更新:

$$b_u \leftarrow b_u + Y(e_{ui} - \lambda_1 b_u)$$

$$b_i \leftarrow b_i + Y(e_{ui} - \lambda_1 b_i)$$

$$\forall v \in R_k(u)$$

$$w_{uv} \leftarrow w_{uv} + Y(e_{ui}(r_{vi} - b_{vi}) - \lambda_1 w_{uv})$$

其中: $e_{ui} = r_{ui} - \hat{r}_{ui}$ 为误差, Y 为梯度下降的步长, λ_1 为惩罚因子。在每一轮迭代后, 计算验证集合的平均误差值来判断是否终止迭代; 或在经过设定的轮次迭代后终止迭代过程。

通过上述方法可以根据用户 u 的档案信息找到与用户 u 最邻近的 k 个用户, 并利用用户 u 和 k 个近邻权重 w_{uv} 预测用户的评分情况。当用户 u 没有历史记录时, 就将用户 u 与邻居的档案信息相关程度当作权重来预测用户的评分情况。这样既能有效利用用户档案和评分历史记录来预测用户评分, 又能解决新使用者的问题, 有效提高预测精度。

3 基于矩阵分解与用户近邻模型的推荐算法

矩阵分解模型^[7]的中心思想是将用户—物品评分矩阵分解成若干规模较小的矩阵乘积, 本文中增量奇异值矩阵分解的方式将用户—物品评分矩阵 R 分解为 P, Q 两个矩阵的乘积, $R = P \times Q$ 。这样做有两点好处: 1) 能够有效减少空间复杂度; 2) 能够提取出隐藏的 k 维属性, 为预测矩阵缺失值(用户对物品评分的预测)提供依据。

要将用户评分矩阵 R 分解成矩阵 P 和 Q 的乘积, 本文所使用的方法是求最小化下式:

$$\sum (r_{ui} - p_u \times q_i^T)^2 + \lambda_1 (|p_u|^2 + |q_i|^2)$$

其中: r_{ui} 为 R 矩阵中的第 u 行、第 i 列的已知打分值; p_u 为用户 P 矩阵中的第 u 行; q_i 为物品 Q 矩阵中的第 i 行; $\lambda_1 (|p_u|^2 + |q_i|^2)$ 是为避免相对数据量比较小的物品和数据过拟合现象, 需要加惩罚因子来约束递归调整过程。

为了最小化上式, 通过应用随机梯度下降法, 得到:

$$p_{uk} \leftarrow p_{uk} + Y(e_{ui} q_{ik} - \lambda_1 p_{uk})$$

$$q_{ik} \leftarrow q_{ik} + Y(e_{ui} p_{uk} - \lambda_1 q_{ik})$$

其中 k 表示 f 维向量中第 k 个值。训练过程中, 首先初始化 p_u 和 q_i , 随机对 p_u, q_i 中的 f 维的向量赋值。对于一个已知的评分记录 r_{ui} , 计算 $\hat{r}_{ui} = p_u \times q_i^T$, 得到误差 $e_{ui} = \hat{r}_{ui} - r_{ui}$, 然后对每一维都做更新。在完成 20 ~ 30 轮迭代更新之后, 得到 P, Q 两个矩阵, 就是所需的分解矩阵。

结合上一章介绍的用户近邻模型, 为了进一步提高预测准确性, 本文将矩阵分解方法也加入预测公式中, 因此得到如下预测公式:

$$\hat{r}_{ui} = b_{ui} + \sum_{v \in R_k(u)} (r_{vi} - b_{vi}) \times w_{uv} + p_u \times q_i^T$$

为了最小化

$$\sum \left(r_{ui} - u - b_u - b_i - \sum_{v \in R_k(u)} (r_{vi} - b_{vi}) \times w_{uv} - p_u \times q_i^T \right)^2 + \lambda_1 (b_u^2 + b_i^2 + \sum_{v \in R_k(u)} w_{uv}^2 + |p_u|^2 + |q_i|^2)$$

从而计算 $b_u, b_i, p_u, q_i, w_{uv}$ 使用了如下的更新过程:

LearnFactorizedNeighborhoodModel(Known ratings: r_{ui} , rank: f , neighborhood: k)

Const #Iterations = 30, $Y = 0.003$, $\lambda_1 = 0.01$

Initialize $p_u, q_i = \text{Random}(-0.01, 0.01)$

for count = 1, ..., #Iterations do

for $i = 1, 2, \dots, m$ do

if u is a new user

$$\hat{r}_{ui} \leftarrow u + b_u + b_i + \sum_{v \in R_k(u)} (r_{vi} - b_{vi}) \times \text{sim}(u, v)$$

return

else // u have historical rating set $S(u)$

for all $i \in S(u)$ do

$$r_{ui} \leftarrow u + b_u + b_i + \sum_{v \in R_k(u)} (r_{vi} - b_{vi}) \times w_{uv} + p_u \times q_i^T$$

$$e_{ui} \leftarrow r_{ui} - \hat{r}_{ui}$$

$$b_u \leftarrow b_u + Y(e_{ui} - \lambda_1 b_u)$$

$$b_i \leftarrow b_i + Y(e_{ui} - \lambda_1 b_i)$$

for all $v \in R_k(u)$ do

$$w_{uv} \leftarrow w_{uv} + Y(e_{ui}(r_{vi} - b_{vi}) - \lambda_1 w_{uv})$$

for $j = 0, 1, \dots, f$ do

$$p_{uk} \leftarrow p_{uk} + Y(e_{ui} q_{ik} - \lambda_1 p_{uk})$$

$$q_{ik} \leftarrow q_{ik} + Y(e_{ui} p_{uk} - \lambda_1 q_{ik})$$

在样本集中应用上述训练过程, 得到每一个用户的 b_u 和向量 p_u , 每一个物品的 b_i 和向量 q_i , 以及用户 u 的 k 个邻居权重 w_{uv} 。这样通过应用增加奇异值矩阵分解的方式, 有效降低了时间和空间的复杂度, 再结合用户档案信息所选取出的 k 个邻居的近邻模型, 使得用户之间的预测相关性进一步提高, 从而有效地提高了预测的准确性。从实验中可知, 在 $f = 30$, $k = 10$ 时, 在普通 PC 上进行一次迭代更新的时间约为 18 min, 预测的平均绝对误差值为 0.932; 在 $f = 500$, $k = 10$ 时, 进行一次迭代更新的时间约为 43 min, 预测的平均绝对误差值为 0.865; 在 $f = 30$, $k = 30$ 时, 进行一次迭代更新的时间约为 22 min, 预测的平均绝对误差值为 0.927。所以当 f 和 k 越来越大时, 预测的准确性会越来越好, 但是时间和空间的复杂度也会越来越高。

4 实验分析

本章通过实验来检验本文提出算法的推荐质量, 并讨论关于维度和邻居数量的调整对推荐结果的影响; 同时结合用户档案与矩阵分解的推荐算法与其他协同过滤推荐算法的比较。

4.1 数据集

实验所使用的数据集是由 Minnesota 大学的 GroupLens 项目研究组所提供的 GroupLens (<http://www.grouplens.org>) 数据集,其中有 10 万条电影评分记录的数据集,记录了 943 个用户对 1682 部电影的评分,另外还有用户和电影的基本档案信息。每个用户至少对 20 部电影进行了评分,评分范围 1~5,越高表示越满意。

整个实验数据集需要进一步划分为训练集和测试集,为此将整个数据集的 80% 作为训练集,20% 作为测试集,并在训练集中去除其中 200 名用户的历史打分记录,然后加入测试集合,作为新用户预测。

4.2 度量标准

本文采用平均绝对误差 (Mean Absolute Error, MAE) 作为度量标准,也是目前最常用的一种推荐质量度量方法。通过计算预测的用户评分与实际的用户评分之间的误差,来表示预测的准确性,MAE 值越小,推荐质量就越好。

设总共有 n 条评分记录, r_{ui} 为实际评分, \hat{r}_{ui} 为预测评分。

$$MAE = \left(\sum_{i=1}^n |r_{ui} - \hat{r}_{ui}| \right) / n$$

4.3 实验结果及分析

由图 1 可知,随着矩阵分解维度 f 和近邻数量 k 的增加,预测精确度逐步增加,同时也增加了时间和空间的消耗。在 $f=500, k=30$ 时,在普通 PC 上进行一次迭代更新所需的时间大概为 47 min; 在 $f=50, k=30$ 时,运行一次迭代更新所需的时间为 20 min。

为检验本文算法 (Proposed) 的有效性,以传统的协同过滤推荐算法作为对照,在传统的协同过滤推荐算法中,分别以余弦相似性 (Cosin-Based) 和相关系数 (Corre-Based) 作为邻居的权重信息,并计算其 MAE, 实验结果如图 2 所示。

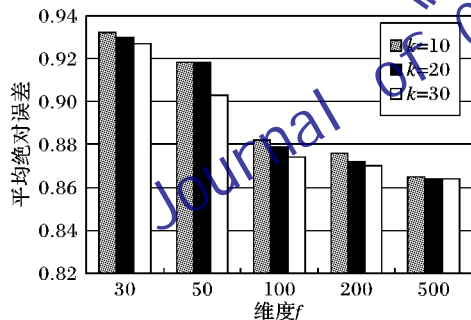


图 1 矩阵分解维度 f 和近邻数量 k 的关系

在图 2 中,选取潜在特征维数 $f=50$ 与不同的近邻数量计算出来的结果。在各种实验条件下,本文提出的基于矩阵分解与用户近邻模型的推荐算法均具有最小的 MAE, 由此可知,与传统的协同过滤算法相比,本文提出的推荐算法可以显著地提高推荐系统的推荐质量。

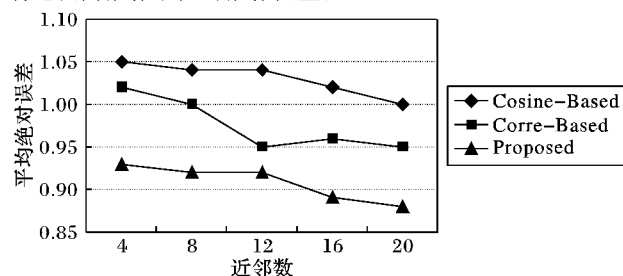


图 2 推荐算法精确度比较

5 结语

协同过滤推荐算法是推荐系统应用中的一个热门研究课

题。本文针对传统协同过滤算法面对的稀疏矩阵和新使用者问题提出了基于矩阵分解与用户近邻模型的推荐算法。该算法充分利用了已有的用户档案信息和矩阵分解降维的方法,减少了空间开销,提高了预测的准确性。但是,如何发现并解决在训练过程中矩阵赋初值,以及模型过拟合的问题仍值得继续研究。

参考文献:

- [1] LINDEN G, SMITH B, YORK J. Amazon.com recommendations: Item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [2] ALI K, WIJNAND V S. TiVo: Making show recommendations using a distributed collaborative filtering architecture [C]// Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2004: 394-401.
- [3] GOLDBERG K Y, ROEDER T, GUPTA D, et al. Eigentaste: A constant time collaborative filtering algorithm [J]. Information Retrieval, 2001, 4(2): 133-151.
- [4] SALAKHUTDINOV R, MNIH A, HINTON G. Restricted Boltzmann machines for collaborative filtering [C]// Proceedings of the 24th International Conference on Machine Learning. New York: ACM, 2007: 791-798.
- [5] MARLIN B. Collaborative filtering: a machine learning perspective [D]. Toronto: University of Toronto, 2004.
- [6] SU X, KHOSHGOFTAAR T M. A survey of collaborative filtering techniques [J]. Advances in Artificial Intelligence, 2009, 2009(4): 421-445.
- [7] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C]// Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2008: 426-434.
- [8] HOFMANN T. Latent semantic models for collaborative filtering [J]. ACM Transactions on Information Systems, 2004, 22(1): 89-115.
- [9] BLEI D, NG A, JORDAN M. Latent dirichlet allocation [J]. Journal of Machine Learning Research, 2003, 3: 993-1022.
- [10] DEERWESTER, DUMAIS S, FURNAS G W, et al. Indexing by latent semantic analysis [J]. Journal of the Society for Information Science, 1990, 4(1): 391-407.
- [11] KIM D, YUM B-J. Collaborative filtering based on iterative principal component analysis [J]. Expert Systems with Applications, 2005, 28(4): 823-830.
- [12] HERLOCKER J L, KONSTAN J A, BORCHERS, et al. An algorithmic framework for performing collaborative filtering [C]// Proceedings of the 22nd ACM SIGIR Conference on Information Retrieval. New York: ACM, 1999: 230-237.
- [13] BELL R M, KOREN Y, VOLINSKY C. Modeling relationships at multiple scales to improve accuracy of large recommender systems [C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2007: 95-104.
- [14] FUNK S. Netflix update: try this at home [EB/OL]. [2011-06-31]. <http://sifter.org/simon/journal/20061211.html>.
- [15] 陈健, 印鉴. 基于影响集的协作过滤推荐算法 [J]. 软件学报, 2007, 18(7): 1685-1694.
- [16] 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.
- [17] 吴一帆, 王浩然. 结合用户背景信息的协同过滤推荐算法 [J]. 计算机应用, 2008, 28(11): 2973-2974.
- [18] 曾春, 邢春晓, 周立柱. 个性化服务技术综述 [J]. 软件学报, 2002, 13(10): 1952-1961.