第46卷第9期 中国科学技术大学学报 Vol. 46, No. 9
2016年9月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Sep. 2016

文章编号:0253-2778(2016)9-0736-07

基于最近邻居聚类的协同过滤推荐算法

魏慧娟, 戴牡红, 宁勇余

(湖南大学信息科学与工程学院, 湖南长沙 410082)

摘要:随着推荐系统用户数量和服务项目增多, 可扩展性问题成为推荐算法应用的瓶颈. 目前, 大部分推荐算法以及基于这些算法的改进主要集中在推荐质量上, 随着系统规模扩大, 暴露出实时推荐效率降低和运行耗时的缺点. 针对这些问题, 提出了一种基于最近邻聚类的协同过滤推荐算法. 首先, 该算法采用二分k-means算法把评分相似的用户划分到相同的类中, 以此建立用户聚类模型. 然后, 从聚类模型中挑选出目标用户的最近邻居类作为检索空间. 最后, 从检索空间中搜索目标用户的最近邻居, 由最近邻居的信息产生最终的推荐列表. 实验结果表明, 该算法在保持较高的推荐质量的同时可以显著提高推荐系统的效率, 比传统的协同过滤算法可扩展性强.

关键词:推荐系统;系统过滤;划分聚类;扩展性

中图分类号:TP311.132 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2016.09.004

引用格式:魏慧娟, 戴牡红, 宁勇余. 基于最近邻居聚类的协同过滤推荐算法[J]. 中国科学技术大学学报, 2016, 46(9):736-742.

WEI Huijuan, DAI Muhong, NING Yongyu. Collaborative filtering recommendation algorithm based on nearest neighbor clustering[J]. Journal of University of Science and Technology of China, 2016, 46(9):736-742.

Collaborative filtering recommendation algorithm based on nearest neighbor clustering

WEI Huijuan, DAI Muhong, NING Yongyu

(*College of Information Science and Engineering, Hunan University, Changsha* 410082, *China*)

Abstract: With the increasing number of users and items in recommender systems, designing a scalable algorithm becomes a big challenge for recommendation systems. However, many recommendation algorithms and the improved algorithms proposed thus far have focused on improving recommendation quality, resulting in shortcomings such as lower recommendation efficiency and running time consumption as the system increases in scale. To address the problem of scalability, a collaborative filtering recommendation algorithm based on nearest neighbor clustering was proposed. Firstly, the k-means algorithm was utilized to place similar scores into the same cluster, which was used to build the user clustering model. Then, it picked out the active users' nearest neighbor clusters from the clustering model and treats them as a retrieval space. Finally, the nearest neighbors of an active user are found according to the retrieval space, and the recommendation to the active user was given. Experimental results show that

收稿日期:2016-03-01;**修回日期**:2016-09-17

基金项目:湖南省自然科学基金(2015JJ2027)资助.

作者简介:魏慧娟, 女, 1990年生, 硕士生. 研究方向:数据挖掘. E-mail: 1194883962@qq.com

通讯作者:戴牡红, 博士/教授. E-mail: 695940548@qq.com

the algorithm proposed in this paper not only significantly improves the response speed of the recommendation system online but also maintains a high accuracy.

Key words: recommendation system; collaborative filtering; partition-based clustering; scalability

0　引言

随着互联网技术飞速发展，互联网上的信息资源数量呈指数增长，用户很难从这些信息中找出自己真正需要的信息. 推荐系统作为一种有效的工具应运而生，它根据用户习惯和兴趣向用户推荐值得浏览的信息，在很大程度上解决了用户的信息需求问题[1].

协同过滤技术是目前推荐系统中应用最为广泛和成功的技术之一[2]，其利用相似用户的偏好预测目标用户对未浏览过的信息的兴趣度. 协同过滤技术已广泛应用在电子商务、社交网络和学术信息查阅等 Web2.0 服务当中. 如 Amazon、Net-flix、eBay 等网站都采用协同过滤算法进行商品推荐[3]. 传统的协同过滤存在诸如扩展性、稀疏性和冷启动等问题. 随着推荐系统规模扩大，其包含的用户数量和服务项目不断增多，协同过滤的扩展性问题变得尤为突出，严重影响推荐生成的速度和质量[4].

针对推荐系统的扩展性问题，国内外研究者进行了很多相关研究，提出了多种方法对传统的推荐算法进行改进[5]. Newton 等[6]在 2004 年提出基于 PRM 的推荐技术，通过建立用户和项目的关系模型进行推荐；一旦建立好模型，其推荐效率很高，但是推荐质量还需提高. 文献[7-8]提出了以基于 SVD 模型的协同过滤算法，使用奇异值分解技术对特征向量进行降维，加快了最近邻居的搜寻速度，但降维会导致信息的丢失. 文献[9]利用 Boltzmann 机对协同过滤推荐系统进行建模，使得该模型能更好地处理大数据集，其性能优于 SVD 模型. 除了奇异值分解，聚类分析等数据挖掘的方法也经常被用来提高推荐算法的效率. 例如，文献[10]中对喜欢相似项目的用户根据项目类别聚类，具体提出了根据用户年龄、性别和职业等属性聚类的方法. 可是这些推荐都需要提前收集项目的内容描述以及用户的个人属性等大量额外信息. 文献[11]通过聚类选取模范用户，以模范用户进行推荐，但是模范所代表的兴趣过于狭隘，不能为目标提供全面的推荐. 文献[12]综合了对用户和项目两种数据聚类，提高了推荐质量. 文献[13]提出了一种快速 k-medoids 聚类的混合推荐算法，比 k-medoids 和 k-mean 聚类效果更好，提高了推荐算法的推荐效率，但是推荐质量不高. 除此之外，Hadoop 云平台下的并行化计算也被用来解决大数据集推荐效率低的问题[14].

根据上述研究中的一些不足，本文对协同过滤推荐算法进行了改进，提出了一种可高度扩展的基于最近邻聚类的协同过滤推荐算法，该算法首先引入划分聚类技术把评分相似的用户划分到相同的类中. 然后，根据目标用户同各聚类中心的相似性找到“最近邻居类”，把“最近邻居类”作为目标用户的最近邻居的检索空间. 最后，根据精简后的最近邻居的兴趣进行推荐. 本文通过实验证明，该算法具有很强的扩展性，实时推荐速率较传统的协同过滤算法快，并且保持了较高的推荐准确性.

1　协同过滤算法

图 1 显示了传统的协同过滤系统的典型架构. 由左侧表格所示的用户评分矩阵作为输入，第 i 行与第 j 列的值表示用户 u_i 对商品 i_j 评分，如果未评分则评分设为 0. 由右侧两种情况作为输出，输出结果分为两种：预测目标用户对指定商品的兴趣度；推荐给目标用户可能兴趣度最高的 top-N 商品列表.

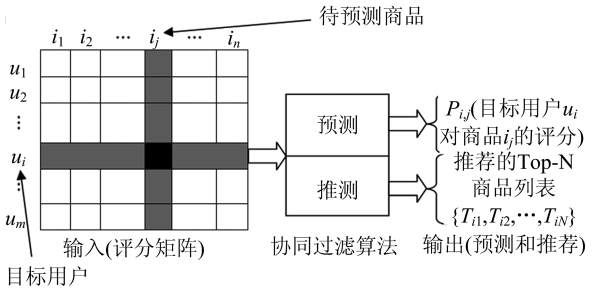

图 1　用户评分数据及协同过滤推荐过程

Fig. 1　The user rating data and CF process

传统的协同过滤推荐可以分为基于内存(memory-based)推荐和基于模型(model-based)推荐这两种. 基于内存推荐方法利用 K 邻居(KNN)算法来寻找最近邻居并以此产生推荐，主要包括基于用户推荐(user-based)和基于项目(item-based)推荐这两种推荐. 基于内存的协同过滤算法实现简单，需要的训练数据容易获取，并且成本低. 基于内存的推荐算法在用户和商品数量很大的时候计算速

度会大大下降,不适用于大型推荐系统.基于模型的推荐算法分为离线和在线两个阶段行进,离线阶段利用训练数据集学习一个预测模型,再将模型应用到在线系统进行推荐.主要包括因子模型、贝叶斯分类模型和图模型等.基于模型的协同过滤算法在模型已经建立的基础上实行推荐,能够省去很多在线计算过程,迅速做出推荐,但是基于模型的推荐算法在模型建立过程中非常耗时.

根据上述两种方法的优缺点分析总结,本文提出了一种基于最近邻聚类的协同过滤推荐算法.该算法对基于模型和基于内存这两种协同过滤推荐算法进行混合,继承两种算法的优点以提高推荐效率.

2 基于最近邻居类的协同过滤算法

2.1 聚类建模

算法的第一步需要对用户进行聚类,把相似兴趣的用户划分在同一类别中.在传统的推荐算法中,找到最近邻居需要计算目标用户同所有用户之间的距离,计算的时间复杂度高.本文采用划分聚类算法将用户划分成有意义的组(簇),使用簇原型(代表簇中其他用户的质心用户)来刻画簇特征,有效地寻找最近邻居.使用簇原型可以减少寻找最近邻居所需要计算的次数.直观地说,如果目标用户同某个簇原型相距很远,则对应簇中的对象与目标用户互为近邻居的概率很小,可以忽略.因此,为了找出一个目标用户的最近邻,只需要计算到邻近簇中对象的距离,极大地缩小了计算范围.

本文采用二分 k-means 算法对用户进行聚类划分,该聚类算法是基本 k-means 算法的直接扩充.基本 k-means 算法的聚类结果受初始质心选择影响过大,如果质心的个数选择不当,即使重复多次运行,每次使用一组不同的随机初始质心也无法克服,只能得到局部最优.二分 k-means 算法克服了基本 k-means 的这些缺点,其具体算法描述如下:

算法 2.1 基于二分 k-means 的用户聚类算法

输入:聚类数目 K 和用户评分向量集 U

输出:K 个类

1: 初始化簇表 T,使之包含由所有点组成的簇.
2: repeat
3: 从簇表中取出一个簇.
4: {对选定的簇进行多次二分"试验".}
5: for i=1 to 试验次数 do
6: 使用基本 k-means,二分选定的簇.
7: end for
8: 从二分试验中选择具有最小总 SSE 的两簇.
9: 将这两个簇添加到簇表中.
10: until 簇表中包含 k 个簇.

在算法 2.1 中,初始化定义一个簇表 T 用于预存放簇.首先,将所有评分向量集 U 作为一个簇,放入簇表 T 中.然后,从簇表 T 中取出一个簇,用基本 k-means 算法进行第一次聚类迭代,找到具有最小 SSE(误差的平方和)的 2 个簇,把这 2 个簇放回簇表 T 中.计算簇表 T 中所有簇的误差和 SSE,如此循环从簇表 T 中取出簇进行二分聚类,再放回簇表 T 中,经过 k-1 次迭代得到 k 个簇.每个簇代表具有相似兴趣的一类用户.SSE 定义如下:

$$\mathrm{SSE}=\sum(u_i-C)^2 \tag{1}$$

式中,C 表示簇中心(类的质心用户)向量,u_i 表示 C 所对应簇中的用户评分向量.

在二分 k-means 算法中,"局部"地使用了 k-means 算法,最终的簇集不代表使 SSE 局部最小的聚类.相对于 k-means 算法,避免了随机产生质心而得到局部最优化结果.

2.2 推荐算法

下面详细介绍算法的流程,首先,假设系统数据集的组成如下:用户集 $\{u_1,u_2,\cdots,u_n\}$,商品集 $\{I_1,I_2,\cdots,I_m\}$ 和用户对商品的评分数据.整个算法分为离线建模和在线推荐两个阶段.计算流程如下:

2.2.1 离线建模部分

(Ⅰ)根据系统数据(包括数据集的大小等因素)确定预计聚类所得的用户簇的数目 k.

(Ⅱ)采用二分 k-means 算法对用户进行聚类,得到 k 个用户簇,每个簇拥有相应的 k 个聚类中心.

(Ⅲ)把 k 个聚类中心表示成 k 个质心用户.由这 k 个质心用户建立新的质心用户矩阵.即每个质心用户代表其所在用户簇的平均兴趣度.这里假设 k 个聚类中心用 $\{C_1,C_2,\cdots,C_k\}$ 表示.其中,C_i 表示由 m 个商品组成的 m 维向量:$C_i=(\widetilde{R}_{c_i,I_1},\widetilde{R}_{c_i,I_2},\cdots,\widetilde{R}_{c_i,I_m})$.其中 $\widetilde{R}_{c_i,I_j}$ 表示 C_i 所对应的用户簇中的用户对商品 I_j 的平均评分.

2.2.2 在线推荐部分

假设预测目标为用户 u_t 对商品 I_t 的评分 $\widehat{R}_{c_t,I_t}$,下面对算法进行详细描述:

(Ⅰ)根据离线模型所建立的质心用户矩阵,采用修正过的 Pearson 相关系数计算目标用户 u_t 与 k 个质心用户之间的相似性:

$$w_{u_t,c_i}=\frac{\sum_{I?\tau}(R_{u_t,I}-\bar{R}_{u_t})(\widetilde{R}_{c_i,I}-\bar{R}_{c_i})}{\sqrt{\sum_{I_\tau}(R_{u_t,I}-\bar{R}_{u_t})^2\sum_{I_\tau}(\widetilde{R}_{c_i,I}-\bar{R}_{c_i})^2}} \tag{2}$$

式中，τ 表示目标用户和质心用户共同评过分的商品集.

(Ⅱ)通过上述计算得到与目标用户 u_t 最相似的 l 个质心用户，用这 l 个类中的用户构成新的用户评分矩阵 $\boldsymbol{U}$.

(Ⅲ)基于新的用户评分矩阵 $\boldsymbol{U}$ 采用协同过滤算法预测用户 u_t 对商品 I_t 的评分 $\hat{R}_{c_t,I_t}$：

$$\hat{R}_{c_t,I_t}=\bar{R}_{u_t}+\frac{\sum_{v\in U}(R_{v_i}-\bar{R}_v)\,w_{u_t,v}}{\sum_{v\in U}w_{u_t,v}} \tag{3}$$

由此预测出用户 u_t 对商品 I_t 的评分 $\hat{R}_{c_t,I_t}$.

3　时间复杂度分析

根据上一节算法描述，该算法的时间复杂度由两部分构成：离线建模部分和在线推荐部分. 接下来从时间复杂度的角度分析该算法的推荐速率.

离线建模部分：传统的 k-means 算法时间复杂度为 $O(n)$. 本文采用二分 k-means 对 n 个用户和 m 个商品聚类，需重复经过 k-1 次划分产生 k 个簇，时间复杂度为 $O((k\text{-}1)mn)$. 由于聚类数 k-1 往往远小于用户和商品数目，离线建模部分的时间复杂度可近似为 $O(mn)$.

在线推荐部分：目标用户需同 k 个聚类中心进行 k 次相似度计算，时间复杂度为 $O(k)$，每次相似度计算时间复杂度为 $O(m)$，因此在线推荐部分的时间复杂度为 $O(km)$. 由于聚类数目 k 往往远小于商品数目 m，在线推荐部分的时间复杂度简化为 $O(m)$.

本文选择了几种最常用的协同过滤算法与基于最近邻聚类的协同过滤算法进行了时间复杂度比较，如表 1 所示. 从理论上可以得出，基于最近邻聚类的协同过滤算法在线推荐速度是最快的. 之后，将继续通过实验来说明其推荐效率.

表 1　几种常用系统过滤算法时间复杂度比较

Tab. 1　Comparison of time-complexities of the selected CF algorithms

协同过滤算法	离线部分	在线部分
User-based CF	\\	$O(mn)$
Item-based CF	\\	$O(mn)$
pLSA-based	$O(mn)$	$O(m)$
SVD	$O(n^2m+m^2n)$	$O(m)$
基于最近邻聚类的协同过滤推荐算法	$O(mn)$	$O(m)$

4　相关的实验分析

下面给出基于最近邻聚类的协同过滤推荐算法的实验数据、度量指标和实验结果分析.

4.1　实验数据

数据集来源自 GroupLens 创建的 MovieLens 推荐网站(http://grouplens.org). 该网站为一个电影推荐网站，到目前为止，MovieLens 拥有超过 160 000 的注册用户，10 000 多部电影，用户的评分次数达 13 000 000 次之多.

实验中，针对大数据集的特点，本文提取最近的 3 000 000 次评分数据进行实验，据统计3 000 000次评分源自 20 000 多用户对约 9 000 部电影的打分. 具体数据如表 2 和图 2 所示.

表 2　实验数据集

Tab. 2　Experimental dataset

数据集属性	数值属性
用户(个)	22 041
电影(部)	8 956
评分(次)	3 011 826
平均评分	3.44
稀疏度(%)	98.5%

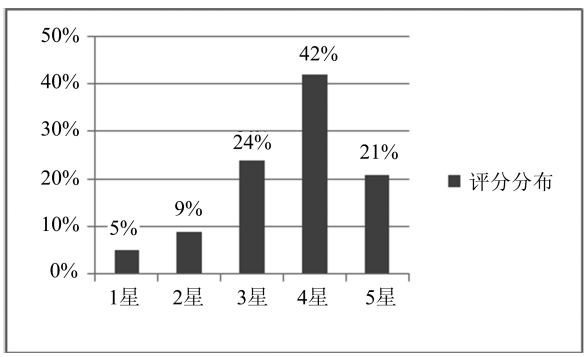

图 2　评分分布

Fig. 2　Rating distribution

4.2　度量指标

本文采用吞吐量(throughput)来验证推荐算法

的实时性以及采用平均绝对误差作为准确性的度量标准.

(Ⅰ)吞吐量的计算公式如下:

$$F = \frac{R}{T} \tag{4}$$

式中,F 为吞吐量(次/秒),R 为系统推荐次数,T 表示在线推荐所用的时间. 吞吐量越大,推荐速率越快,则算法具有更强的可扩展性.

(Ⅱ)推荐系统中评分预测的预测准确率一般通过均方根误差(RMSE)和平均绝对误差(MAE)计算. 本文采用平均绝对误差 MAE(normalized mean absolute error)作为预测打分准确率的评价指标,它通过计算预测评分和真实评分的差异度量预测准确率. MAE 的值越小,表示算法的推荐精度越高. 假设由预测得出的用户评分集为$\{R_{u_i}, R_{u_i}, \cdots, R_{u_i}\}$,实际用户评分集为$\{\hat{R}_{u_i}, \hat{R}_{u_i}, \cdots, \hat{R}_{u_i}\}$,则平均绝对误差定义为:

$$\mathrm{MAE} = \frac{\sum_{u,i \in T} | R_{u_i} - \hat{R}_{u_i} |}{T} \tag{5}$$

式中,T 表示测试集数量.

4.3 实验结果分析

本文采用五折交叉验证法进行实验. 实验中把用户数据随机分割成 5 个子样本,一个单独的子样本被保留作为验证模型的测试集,其他 4 个样本用来训练本文提出的基于最近邻聚类的协同过滤推荐算法. 交叉验证重复 5 次,保证每个子样本都能够作为测试集验证一次. 最后,平均 5 次的结果得到一个单一估测.

4.3.1 实时性评估

图 3 显示了基于最近邻聚类的协同过滤推荐算法、传统的协同过滤算法和传统的 k-means 聚类推荐算法三种推荐算法在不同聚类数目下推荐速率的比较结果. 图中横坐标表示用户聚类的数目,为了更加直观地测试推荐算法的可扩展性,设定用户的聚类数目以 10 为单位间隔从 10 个增加到 100 个;图中纵坐标表示吞吐量,以万次每秒为单位. 如图 3 所示,基于最近邻聚类的协同过滤推荐算法比传统协同过滤算法和 k-means 聚类推荐算法的推荐速率有显著提高,并且随着聚类数目的增加速率加快,表现出更强的扩展性. 这是因为聚类数目越多,每个类中的用户数量越小,查询到的指定数量的最近邻居类中的用户也就越加精简,速度变快. 对应的传统协同过滤算法吞吐量与聚类数目无关,固定不变.

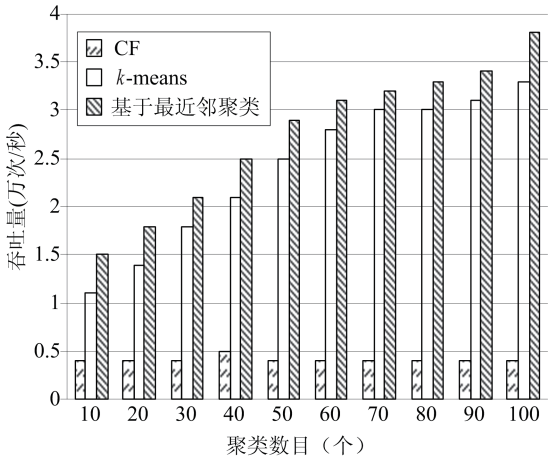

图 3 不同聚类数目下吞吐量比较

Fig. 3 The comparison of throughput of different clustering numbers

图 4 显示了 3 种推荐算法在不同数据集的大小下推荐速率的比较结果. 随着数据集增大,传统的协同过滤推荐算法吞吐量下降,反映出系统的推荐速率降低. 当数据集增大到一定程度时,吞吐量极小,表明传统的协同过滤推荐算法可扩展性很差;传统的 k-means 聚类推荐算法也受到数据集变化的影响,随着数据集的增大,吞吐量呈下降趋势,系统的推荐速率降低,而基于最近邻聚类的推荐算法几乎不受数据集大小的影响,一直维持在一个稳定的水平,表明本文提出的算法具有很强的可扩展性.

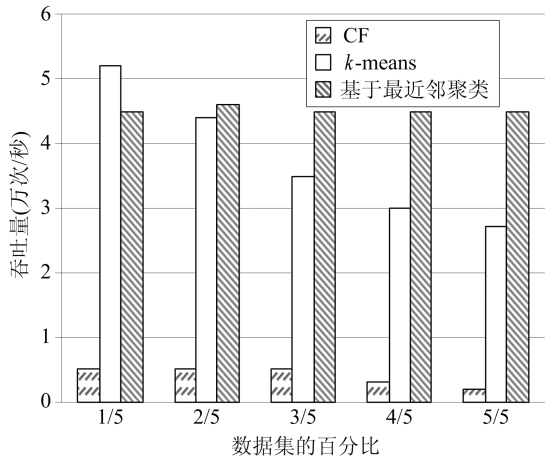

图 4 不同大小数据集下吞吐量比较

Fig. 4 The comparison of throughput of different datasets

4.3.2 准确性评估

为了验证基于最近邻聚类的协同过滤推荐算法的准确率,分别以聚类数目和最近邻居数为自变量进行实验,对基于最近邻聚类的协同过滤推荐算法、

传统的协同过滤算法，传统的 k-means 聚类推荐算法以及基于 SVD 的推荐算法 4 种算法进行比较. 特别地，对基于 SVD 的推荐算法而言，聚类数表示其特征矩阵维数.

图 5 显示了在用户聚类数目变化的情况下四种推荐算法的 MAE 比较. 假设选取与目标用户最相似的前 5 个类，最近邻居数目为 50 进行实验. 图中横坐标表示用户聚类的数目，设定用户的聚类数目以 10 为单位间隔从 10 个增加到 110 个. 纵坐标为平均绝对误差 MAE 值. 随着聚类数目的增加虽然传统的聚类推荐算法表现出较高的推荐速率，但是准确率也随着下降，在 4 种算法中预测准确率最低. 基于最近邻居聚类推荐算法的准确率随着聚类数目的增加而升高，这是因为聚类的粒度越密集，邻居类的划分越细，系统做出预测的参考项就越多. 当聚类数目达到 70 左右时，基于最近邻聚类算法的准确率逐渐平缓，MAE 的值接近 0.7，表现出最高的预测准确率.

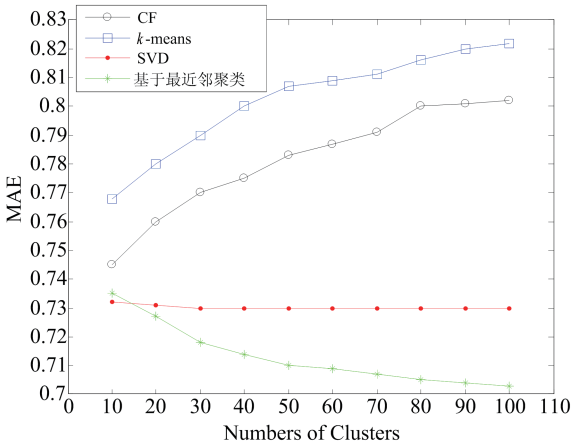

图 5　用户聚类数目变化时 MAE 比较

Fig. 5　The MAE with varying clusters

图 6 显示了在最近邻居数目变化的情况下 4 种推荐算法的 MAE 比较. 假设选取与目标用户最相似的前 5 个类，用户聚类数目为 70 进行实验. 图中横坐标表示最近邻居的数目，设定用户的最近邻居数目以 20 为单位间隔从 20 个增加到 200 个为止，经过反复试验发现最近邻居数目超过 100 个之后所得结果趋于平缓；纵坐标为平均绝对误差 MAE 值. 随着目标用户的最近邻居个数增加，4 种算法的 MAE 变小，预测准确度提高. 传统的协同过滤推荐算法预测准确率比 k-means 聚类算法的准确率高，基于 SVD 的推荐算法 MAE 值保持在一个稳定水平，基于最近邻居聚类的推荐算法表现出最高的预测准确率.

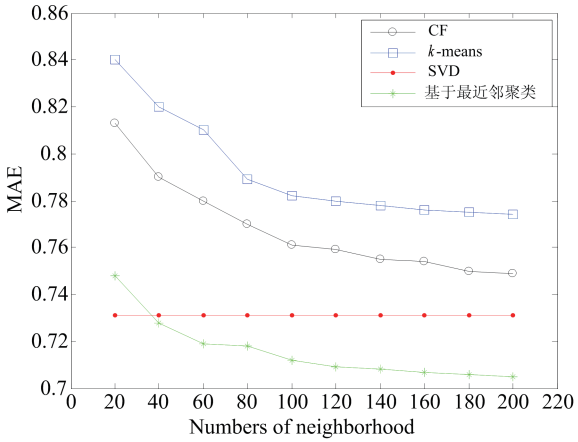

图 6　最近邻居数目变化时 MAE 比较

Fig. 6　The MAE with varying nearest-neighbors

5　结论

协同过滤作为推荐系统中广泛使用的较成功的推荐技术，被许多大型商务网站所使用，但却面临可扩展性差的问题，随着用户和商品数量的增加，这种缺点越加明显. 因此，本文提出了一种基于最近邻聚类的协同过滤推荐算法，该算法利用二分 k-means 算法在线下根据评分相似性对用户进行聚类建模，根据该聚类模型选择目标用户的最近邻居类作为检索空间，从检索空间中搜索目标用户的最近邻居. 这种策略能够将爱好相似的用户划分到同一空间中，基于具有共同爱好的用户进行推荐，大大缩小了最近邻居的搜索范围. 该算法灵活多变且具有很强的扩展性，适用于用户和商品数量较多的大型推荐系统. 随着用户和商品数量的增加，用户的聚类结果将受到影响. 下一步的工作中将在数据流上进行聚类，即对当前数据进行聚类的同时，随着新数据的不断流入，动态地调整和更新聚类结果以达到真实反应数据流的聚类形态. 另外，该算法的准确率会受到数据稀疏性的影响，在以后的工作中，也将致力于用户和项目间关系联合聚类，改善数据稀疏性带来的影响，以求达到更高的推荐效率.

参考文献(References)

[1] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions [J]. IEEE Transaction on Knowledge and Data Engineering, 2005, 17(6): 734-749.

[2] 刘青文. 基于协同过滤的推荐算法研究[D]. 中国科学

技术大学，2013.
[3] MENG X W，HU X，WANG L C，et al. Mobile recommender systems and their applications [J]. Journal of Software，2013，24(1):91-108.
[4] CONSTANTINOPOULOS C，LIKAS A. Unsupervised learning of Gaussian mixtures based on variational component splitting[J]. IEEE Transactions on Neural Networks，2007，18(3)：745-755.
[5] CACHEDA F，CARNEIRO V，FERNÁNDEZ D，et al. Comparison of collaborative filtering algorithms：Limitations of current techniques and proposals for scalable，high-performance recommender systems[J]. ACM Transactions on the Web，2011，5(1)：161-171.
[6] NEWTON J，GREINER R. Hierarchical probabilistic relational models for collaborative filtering [C]// Proceedings of the 21st International Conference on Machine Learning—Workshop on Statistical Relational Learning. New York：ACM Press，2004：249-163.
[7] BELL R，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. San Jose：ACM Press，2007：95-104.
[8] DAKHEL G M，MAHDAVI M. A new collaborative filtering algorithm using *K*-means clustering and neighbors' voting [C]// Proceedings of the 11th International Conference on Hybrid Intelligent Systems. Melacca，Malaysia：IEEE Press，2011，：179-184.
[9] FU H G，PENG J. Improved collaborative filtering algorithm based on model users [J]. Computer Engineering，2011，37(3):70-71，74.
[10] WEI S Y，YE N，ZHANG S，et al. Collaborative filtering recommendation algorithm based on item clustering and global similarity[C]// Proceedings of the 5th International Conference on Business Intelligence and Financial Engineering. Lanzhou，China：ACM Press，2012：69-72.
[11] SALAKHUTDINOV R，MNIH A，HINTON G. Restricted Boltzmann machines for collaborative filtering[C]// Proceedings of the 24th International Conference on Machine Learning. Corvallis，USA：ACM Press，2007：791-798.
[12] STRUNJAS S. Algorithms and models for collaborative filtering from large information corpora[D]. University of Cincinnati，USA，2008.
[13] SHINDE S K，KULKARNI U V. Hybrid personalized recommender system using fast *K*-medoids clustering algorithm [J]. Journal of Advances in Information Technology，2011，2(3)：152-158.
[14] JIANG J，LU J，ZHANG G Q，et al. Scaling-up item-based collaborative filtering recommendation algorithm based on hadoop[J]. Computer Science Technology，2011，7(4)：123-126.