DOI:10.11830/ISSN.1000-5013.201812074

Slope One-BI 算法的改进及其在大数据平台的并行化

刘佳耀,王佳斌

(华侨大学 工学院,福建 泉州 362021)

摘要: 针对大数据时代下 Slope One 算法推荐效率不高的问题,提出结合聚类和动态 K 近邻的双极 Slope One 推荐算法.首先,结合 Canopy 和 K-medoids 的聚类算法把相似的用户汇聚到一起.然后,在所属聚类中,根据用户之间相似度的具体情况动态地寻找最近邻,并用 Slope One-BI 算法推荐预测.最后,在 Spark 平台上实现并行化.在电影数据集上的实验结果表明:基于 Spark 平台的优化算法与其他协同过滤算法相比,推荐精度具有明显优势.

关键词: Slope One-BI 算法;聚类;Spark;推荐算法

中图分类号: TP 391.1 **文献标志码:** A **文章编号:** 1000-5013(2019)06-0786-07

Improved Slope One-BI Algorithm and Parallelization on Big Data Platform

LIU Jiayao, WANG Jiabin

(College of Engineering, Huaqiao University, Quanzhou 362021, China)

Abstract: Since it is not so efficient to recommend the method of Slope One algorithm with the linear regression model to solve the problem caused by the sparse data is proposed. Firstly, the clustering algorithm combining Canopy and K-medoids could bring similar pooled customers. Secondly, the nearest neighbor is dynamically searched in the affiliated clustering according to the specific situation of similarity between costumers by using the Slope One-BI algorithm to predict. Finally, parallelization is implemented on the Spark platform. The experiment shows that, comparing with the other Slope One algorithm by the film data set, the optimized algorithm based on Spark platform has improved the precision of recommendation.

Keywords: Slope One-BI algorithm; clustering; Spark; recommendation algorithm

推荐系统[1]中使用频率最高的是协同过滤[2]技术,而 Slope One 算法[3]是通过预测评分形式进行推荐的一种特殊的协同过滤推荐算法,简单且易于维护.在研究推荐算法时,数据稀疏性、冷启动、用户兴趣变化等问题都会影响推荐的准确性.推荐系统中突出的问题是数据的稀疏性,因为与项目相比,用户只占小部分(用户对项目的评分肯定较稀少),使计算项目(用户)之间的相似度误差太大,影响推荐精度.杨阳等[4]运用奇异值分解(SVD)方法降维来降低数据稀疏性,但在高维的项目空间进行降维效果差,部分信息缺失,推荐效果不佳.黄皓璇等[5]在 Slope One 算法的基础上,结合时间兴趣衰减函数,根

收稿日期: 2018-12-28

通信作者: 王佳斌(1974-),男,副教授,主要从事物联网、云计算、大数据、智能仪器的研究.E-mail:fatwang@hqu.edu.cn

基金项目: 国家自然科学基金青年科学基金资助项目(61505059);福建省厦门市科技局产学研协同创新项目(3502Z20173046);华侨大学研究生科研创新能力培养计划项目(1611422006)

据兴趣随时间变化的规律提升推荐的准确性. 龚敏等[6]利用改进的 K-means 聚类技术对用户进行聚类，产生相关程度高的用户聚类，减少噪声数据混入计算，提高推荐精度. 在大数据的背景下，如果仅仅依靠单机迭代相关的推荐算法，效率极其低下. 马瑞敏等[7]采用大 Hadoop[8]平台运行推荐算法，提高推荐效率，然而，MapReduce 框架存在需要反复在磁盘读取数据的劣势，降低推荐的实时性[9-10]. 本文在 Slope One-BI 算法的基础上，先用聚类算法过滤掉相似度低的噪声用户数据，再在聚类中动态筛选相似度大于阈值 λ 的 K 近邻，对 Slope One-BI 算法进行优化.

1　改进的 Slope One-BI 算法

1.1　Slope One-BI 算法

首先，对 Slope One-BI 和 Weighted Slope One 两种推荐算法在相同数据集 ml-100k 下的平均绝对误差(MAE)进行对比，如表 1 所示.

表 1　单机环境下两种算法的 MAE 的值对比

Tab. 1　MAE value comparison of two algorithms in stand-alone environment

最近邻个数	MAE	
	Weighted Slope One 算法	Slope One-BI 算法
20	0.844	0.838
40	0.834	0.830
60	0.831	0.825
80	0.828	0.825
100	0.825	0.820
120	0.825	0.820

由表 1 可知：Slope One-BI 算法推荐准确性略优于 Weighted Slope One 算法. 在评分预测的时候，运用 Slope One-BI 算法进行预测推荐的准确性更高，因为每个用户对项目的喜欢程度不同，依据用户对项目的评分是否高于此用户对全部项目的评分的均值 $S^{\text{like}}(u)=\{i\in S(u)\mid u_i>\bar{u}\}$ 和 $S^{\text{dislike}}(u)=\{i\in S(u)\mid u_i<\bar{u}\}$，可以把被用户评分过的商品分为 like 和 dislike 两种类别. $p_{j,i}^{\text{like}}$ 代表用户对项目喜欢偏差的预测评分；$p_{j,i}^{\text{dislike}}$ 代表用户对项目不喜欢偏差的预测评分.

$c_{j,i}^{\text{like}}$ 和 $c_{j,i}^{\text{dislike}}$ 分别表示同时喜欢和同时不喜欢项目 j 和 i 的用户数，则预测当前活跃用户 u 对目标项目评分的计算式为

$$\text{dev}_{j,k}^{\text{like}}=\sum_{u\in S_{j,i}^{\text{like}}(S_{u,j})}\frac{r_{u,j}-r_{u,i}}{\text{card}(I_{j,i}^{\text{like}})},\qquad P^{\text{BI}}(u)_j=\frac{\sum\limits_{i\in R_j}p_{j,i}^{\text{like}}\,c_{j,i}^{\text{like}}+\sum\limits_{i\in R_j}p_{j,i}^{\text{dislike}}\,c_{j,i}^{\text{dislike}}}{\sum\limits_{i\in R_j}(c_{j,i}^{\text{like}}+c_{j,i}^{\text{dislike}})}. \tag{1}$$

1.2　选择动态近邻

获得 K 近邻的值一般取决于相似性的计算，利用修正余弦相似性计算用户之间的相似度作为 Slope One-BI[11]算法的动态 K 近邻的度量标准. 所以用户之间的相似度大小计算式为

$$\text{Sim}(i,j)=\frac{\sum\limits_{u\in U_{i,j}}(r_{u,i}-\overline{r_i})(r_{u,j}-\overline{r_j})}{\sqrt{\sum\limits_{u\in U_{i,j}}(r_{u,i}-\overline{r_i})^2}\sqrt{\sum\limits_{u\in U_{i,j}}(r_{u,j}-\overline{r_j})^2}}. \tag{2}$$

式(2)中：$\overline{r_j}$，$\overline{r_i}$ 为用户对项目 j 和 i 的评分的均值；$U_{i,j}$ 为评价过 j 和 i 的用户集合.

要提高 K 近邻的优势，用相似度 λ 作为区别的阈值，即要成为动态的 K 近邻集合，必须与目前用户的相似度的值要大于等于 λ，每个用户最多只能有 K 个近邻. 实验的阈值都是基于 $\lambda=0.1$，用 KNN(i) 代表某个用户 i 的动态 K 近邻，即

$$\text{KNN}(i)=\{i\mid \text{Sim}(i,j)\geqslant\lambda\}. \tag{3}$$

1.3　基于 Canopy 的 K-medoids 聚类算法

与 K-means 算法对比，K-medoids 聚类算法是对 K-means 算法的中心点选取的一种改进算法，以簇中的某个点到其他点的距离之和最小值作为中心点，避免中心点偏离簇. Canopy 聚类方法的优势是可以对海量的数据进行遍历，快速地粗聚类，得到 Canopy 的值[12]，选取 Canopy 的值作为 K 值. 结合两种聚类算法的优势，利用 Canopy[13]算法遍历 MovieLens 数据集，获得聚类个数 K 值，然后，K-medoids 算法就可以利用指定的聚类个数进行聚类，能够提高聚类质量. 改进的聚类算法有如下 8 个步骤.

Input：数据集 D.

Output:聚类个数.

1) 将输入的数据集 D 先向量化成 List=$\{I_1, I_2, \cdots, I_n\}$,排序后放入内存,选择两个距离阈值 T_1, T_2($T_1 > T_2$)作为 Canopy 算法的初始阈值.

2) 从 List 中任选取一个数据向量 $\boldsymbol{P}$,并获得所选取向量 $\boldsymbol{P}$ 和 List 中剩余样本数据向量之间实际的距离 d.

3) 将 d 小于 T_1 的数据向量归类到 Canopy 集合中,并将 d 小于 T_2 的数据向量从候选中心向量名单中删掉.

4) 重复步骤 2),3),直到 List 的向量集变为 null,Canopy 算法结束.

5) 以 Canopy 的个数作为 K-medoids 算法的聚类个数 K.

6) 计算对象到 K 个中心点的距离,将其划分到最近的簇中,把到中心点的距离加和,记为 Sum.

7) 任意簇中随机选择一个非中心点作为 Ctest,通过计算它与簇中剩余的点的距离,并把求得的所有距离汇总求和,记为 Sumtest. 如果满足 Sumtest<Sum 的条件,则 Ctest 代替原本簇中的中心点 C_i,直到中心点稳定,不再变化为止.

8) 重复迭代步骤 6),7),到每个簇的中心点不再发生变化时,得到最终聚类结果.

1.4 在单机上改进的 Slope One-BI 算法

把改进的 Slope One-BI 算法在单机上运行,有如下 4 个步骤.

1) 对要进行实验的数据集预处理,形成项目评分矩阵.

2) 利用 Canopy 的 K-medoids 聚类算法,得到 K 个聚类,将目标用户划分到最相似的聚类中.

3) 在目标用户 u_i 所属的聚类中,通过评价过项目的用户 u_i,计算 u_i 与其他用户的相似度,筛选出相似度最大的 K 个用户中阈值大于 λ 的用户作为动态 K 近邻用户集合.

4) 在已经筛选过的动态 K 近邻居集合中,辨别相邻的用户对 i 的评分有没有高于评分的均值,再分别得到用户对 j 和 i 评分的 like 和 dislike 偏差值. 最后,通过式(1)计算相关用户 u 对项目 j 的评分预测值 $P^{\mathrm{BI}}(u)_j$.

2 实验设计与结果分析

2.1 系统配置

Spark[14]是一种基于内存的分布式计算框架,与 MapReduce 计算框架中大量磁盘读写相比,可以大幅提高数据读或写的效率,有效地减短 I/O 操作的时间,而且 Spark 大数据平台在许多场景下都可以进行处理数据.

用 VMware 中三台虚拟机组建 Spark 集群,选择里面内存配置最好的 1 台虚拟机作为 Master 节点,其他 2 台作为 Slave 节点. 大数据平台具体的软件配置信息如下:Centos 7 操作系统;JDK 1.8.0;Scala 2.11.4;Hadoop 3.0.0;Spark 1.6.0.

选择 MovieLens 数据集进行实验,包括 3 种量级不同的电影数据集 ml-100k,ml-1M 和 ml-10M. 实验运用五折交叉验证法,把 ml-100k 的数据集[15-18]不按顺序地分为互不重叠的 5 个子集,测试集从中任意选择 1 个,余下的即为训练集. 将 5 次重复实验得出的均值作为最后结果. 实验从并行化过程、数据并行化加速比、Spark 平台算法 MAE 的高低 3 个方面进行比较.

2.2 算法程序流程图

优化的算法在大数据 Spark 平台并行化计算,具体算法流程,如图 1 所示.

2.3 改进的 Slope One-BI 算法在 Spark 平台并行化的任务划分

在海量数据下,不断迭代的相似用户聚类、筛选动态 KNN()和评分预测都需要反复计算评分预测值和用户之间的相似度. 由于需要不断迭代计算,而在内存计算能够提高效率,所以可以把改进的 Slope One-BI 算法在 Spark 平台并行化. 并行化过程分两步进行:

1) 在 Spark 平台上,先进行结合 Canopy 和 K-medoids 聚类算法[15]的并行化;

2) 在 Spark 平台上,再进行结合动态 K 近邻的 Slope One-BI 算法的并行化.

结合 Canopy 和 K-medoids 聚类算法并行计算过程中的任务划分(Map)和汇聚(Reduce)如下.

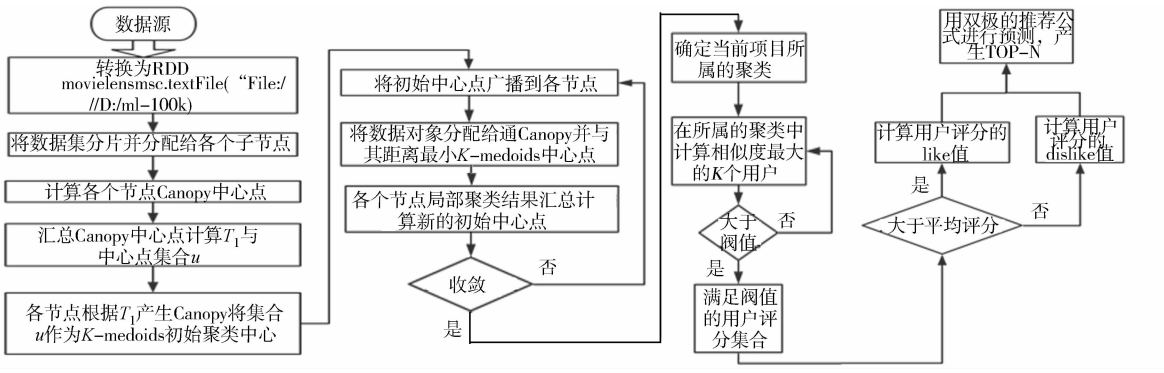

图 1　算法流程图

Fig. 1　Flow chart of algorithm

Map 1：为了得到各个分片的 Canopy 中心点，用 maPartition()函数计算当前中心点和数据向量之间的距离；

Reduce 1：用 reduceByKey(_+_)函数得到每个分片的 Canopies，产生总的 Canopies；

Map 2：将各个分片中的 Canopy 中心点划分到各个分片中，将其作为聚类算法 K-medoids 所需的聚类个数，通过相似度距离公式对数据向量进行划分；

Combine：汇聚 Canopy 中心点所属的数据向量，得到相同 Canopy 中心的数据向量的总和；

Reduce 2：汇聚各个分片聚类结果，不断获得新的 Canopy，确认最终能否收敛，直到算法结束.

在所属聚类中，动态的筛选相似度阈值大于 λ 的组成 K 近邻，并用 Slope One-BI 算法评分预测并行计算过程中的任务划分和汇聚.

Map 1：将输入的 MovieLens 数据集上传到 HDFS()上进行一一的向量化处理，并划分到每个 Slave 机器中，得到相应的 User-Item 的倒排表，从中获得每个 Slave 节点中有过共同评分的用户；

Reduce 1：汇聚用户对项目评过分的用户集合；

Map 2：计算每个分片的用户之间的相似性；

Reduce 2：汇聚所有分片的用户之间的相似性；

Map 3：计算每个分片中用户之间的相似度是否大于相似度阈值 λ；

Reduce 3：汇聚分片中大于 λ 的所有用户组成最近邻集合；

Map 4：在筛选的动态 K 近邻居集合中辨别相邻的用户对 i 的评分有没有高于评分的均值；

Reduce 4：汇聚大于用户评分过商品的平均评分和小于用户评分过商品的平均评分；

Map 5：在汇聚的用户具体评分中，计算用户评分过的商品分为 like 和 dislike 的偏差值；

Reduce 5：从用户具体评分中，得到[(user,item),like]和[(user,item),dislike]的形式，进而得到评分预测值，按评分值从大到小进行推荐.

2.4　加速比的分析

不同量级的加速比，如图 2 所示. 对比测试 3 种量级的电影数据集，加速比为单节点处理数据消耗的时间和多节点并行化后处理数据消耗的时间的比值，用其衡量并行化的优劣，加速比的具体数值越大，说明并行化的效果越好，其计算方式为

$$\text{Speedup}(p) = T_1/T_p, \qquad p = 1,2,3,\cdots. \quad (4)$$

式(4)中：T_1 为单机环境下处理数据消耗的时间值；T_p 为多节点下的处理数据消耗的时间值.

由图 2 可知：随着 Spark 平台中节点数的增加，加速比不断上升；当节点个数到达 3 时，处理越大规模的数据，其加速比越大，说明随着节点数的增加，并行化的性能优势越明显.

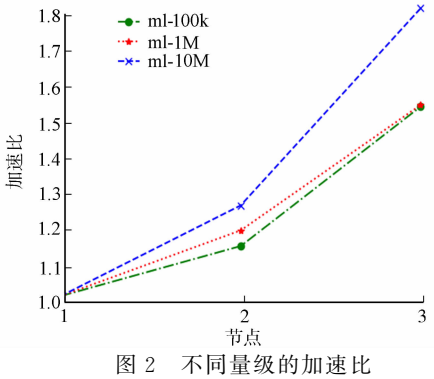

图 2　不同量级的加速比

Fig. 2　Speedup of different scales

2.5 运行效率对比

在Spark平台和单机环境下,用运行相同的ml-100k数据集时间长短衡量运行效率,如图3所示.由图3可知:改进的算法的运行时间随着Spark平台中的机器个数的增加而减少,而且在Spark平台中仅包含单节点的运行效率也远远高于单机下的运行效率.

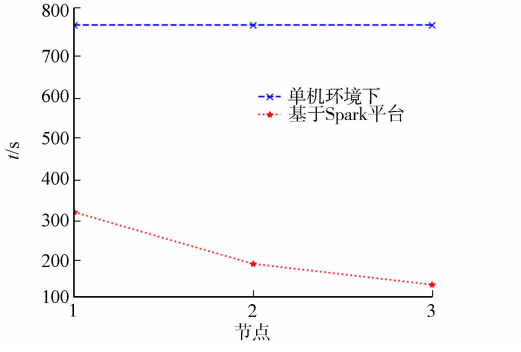

图3 运行效率对比

Fig. 3 Comparison of running efficiency

2.6 平均绝对误差

通常采用平均绝对误差衡量推荐效果的优劣,MAE是推荐算法计算得到的预测评分和实际情况下的评分之间的差值,根据MAE的实际值对比推荐精度,其计算式为

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}|p_i-q_i|}{N}. \tag{6}$$

式(6)中:N为测试集中的总个数;p_i为用户对物品i预测评分值;q_i为对应的用户对物品i的实际评分值.所以,MAE的数值越小,推荐算法的推荐精度越高.

对比Slope One-BI算法、KNN-Slope One-BI算法、MF-KNN算法[16]和文中优化算法在Spark平台的并行化推荐效果的好坏,最近邻个数从20到160的变化过程中,4种算法的MAE值也随之变化,结果如表2所示.

表2 4种算法的MAE值变化

Tab. 2 MAE value changes of four algorithms

K	MAE			
	Slope One-BI算法	KNN-Slope One-BI算法	MF-KNN算法[16]	文中优化算法
20	0.838	0.790	0.833	0.840
40	0.830	0.779	0.822	0.820
60	0.825	0.753	0.815	0.750
80	0.825	0.749	0.790	0.730
100	0.825	0.739	0.762	0.712
120	0.820	0.738	0.760	0.701
140	0.820	0.738	0.759	0.694
160	0.820	0.740	0.759	0.689

由表3可知:最近邻个数从20到160的变化过程中,4种算法的实际MAE值都在不断减小,推荐的准确性也随之变高;当$40<K<60$时,文中优化算法的推荐准确性略优于其他的Slope One算法,但优势不明显;当最近邻个数$60<K<120$,文中优化算法的MAE值已经明显优于其他算法;当K大于120后,4种比较算法的MAE值下降趋势都相对放缓,文中优化算法的MAE值最小,推荐精度优于其他算法.

2.7 单机的算法对比

单机上5种算法对比,如图4所示.为了比较准确性,在单机上利用相同的数据集ml-100k进行实验,对比的算法分别为KNN-Slope One-BI算法、基于用户的协同过滤算法(UBCF)算法、Slope One-BI算法、MF-KNN算法[16],以及文中的优化算法.

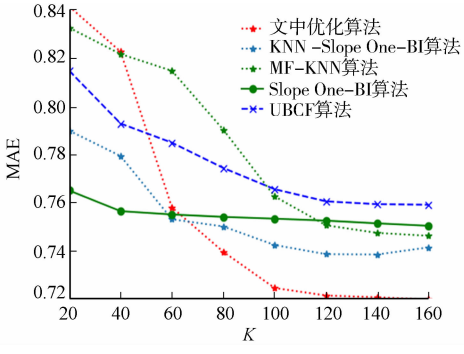

图4 单机上5种算法对比

Fig. 4 Comparison of 5 algorithms on single machine

由图4可知:与Slope One-BI算法进行对比的其他4种算法所属的MAE值都出现逐渐减小的变化,而Slope One-BI算法的MAE值却稳定在一个相对固定的值;当$K<40$时,文中优化算法的推荐精度低于其他推荐算法的精度;当$K>100$时,MAE下降幅度慢慢变小;当$K=120$时,与Slope One-BI算法进行对比的其他4种算法对应的MAE值达到相对低的点,此时推荐效果较好.因此,在Slope One-BI算

法中结合 K 近邻和聚类的方法可以大幅提高推荐精度.

2.8　不同大小数据集验证

为对比提出的优化算法在大小不同数据集上的推荐效果，分别对 ml-100k，ml-1M，ml-10M 和 ml-20M 这 4 个量级的数据集对比. 对 Slope One-BI 算法和文中优化算法进行实验(最近邻个数为 120)，实验结果如表 3 所示.

表 3　不同数据集的 MAE 值对比

Tab. 3　MAE values of different data sets

数据集	MAE	
	Slope One-BI 算法	文中优化算法
ml-100k	0.752	0.690
ml-1M	0.780	0.685
ml-10M	0.825	0.662
ml-20M	0.855	0.651

由表 3 可知：随着电影数据集量级的不断增加，电影数据的用户对项目的评分稀疏性也越来越大，原始的 Slope One-BI 算法的推荐准确性越来越低，而文中的优化算法仍然有相对高的推荐精度.

2.9　单机上和 Spark 平台上的优化算法对比

对电影数据集 ml-100k 进行实验，对比算法在不同环境下的 MAE 值，如图 5 所示. 由图 5 可知：最近邻个数从 20 到 160 的变化过程中，基于 Spark 平台的文中优化算法 MAE 值与单机下几乎保持一致. 这说明提出的优化算法在 Spark 平台上的推荐准确性与单机环境下几乎保持一致. 结合运行效率对比可知：大数据平台并行化不仅能够保证推荐准确性，还能提高改进推荐算法的运行效率.

2.10　大数据平台下的算法对比

针对相同的电影 ml-100k 数据集，比较其他算法和文中优化算法在大数据平台上的推荐精度. 选用的算法分别是基于 Spark 平台的 UBCF 算法、Spark MiLib 库中的最小二乘法(ALS)算法模型、基于 Spark 平台的 ALS-KNN 算法[19]，以及文中优化算法，衡量指标为 MAE 值，算法对比如图 6 所示.

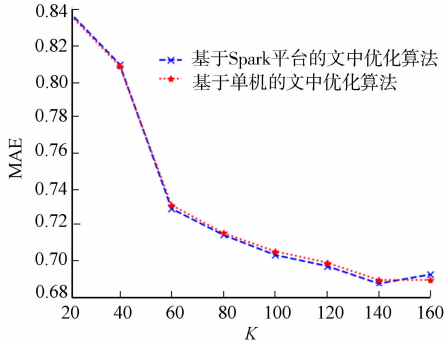

图 5　不同环境下对比算法的 MAE 值

Fig. 5　MAE values of comparison algorithms in different environments

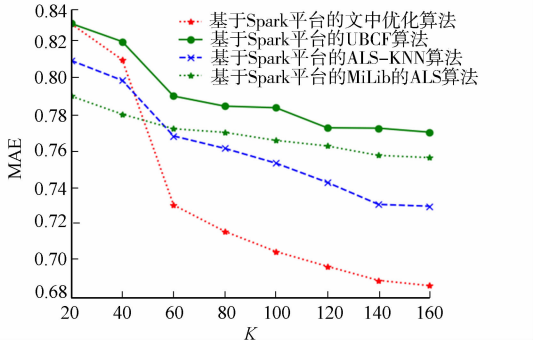

图 6　大数据平台下算法对比

Fig. 6　Comparison of algorithms under big data platform

由图 6 可知：4 种算法在 Spark 大数据平台中进行对比，当 $K<40$ 时，文中优化算法仅优于基于 Spark 平台的 UBCF 算法；但在 $K>60$ 时，文中优化算法明显优于 Spark ML 库中的 ALS 算法、ALS-KNN 算法和 UBCF 算法，说明在 Slope One-BI 的基础上，加入 Canopy 和 K-medoids 聚类并行算法和结合动态 K 近邻的并行算法，可在一定程度上提高推荐的准确性.

3　结束语

在以往的协同过滤算法中，由于电影数据集稀疏性问题和单机的内存迭代的瓶颈，导致推荐效率低下. 文中利用结合 Canopy 和 K-medoids 聚类算法形成相似用户的聚类以缩短搜索相似用户的时间，再在所属聚类中用相似度阈值大于 λ 的 K 近邻去除噪声评分，最后，用 Slope One-BI 算法进行推荐，推荐准确性大幅提高. 将文中改进算法在大数据平台上进行并行化，既能保证与单机环境下的推荐准确率几乎一致，又能提高算法的运行效率.

参考文献：

[1]　AMATRIAIN X. Past, present, and future of recommender systems: An industry perspective[C]// International

Conference on Intelligent User Interfaces. [S. l.]: ACM, 2016: 1. DOI: 10. 1145/2856767. 2856798.

[2] 项亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012.

[3] 孙丽梅, 李悦, 曹科研. 简化的 Slope One 在线评分预测算法[J]. 计算机应用, 2018, 38(2): 497-502. DOI: 10. 11772/j. issn. 1001-9081. 2017082493.

[4] 杨阳, 向阳, 熊磊. 基于矩阵分解与用户近邻模型的协同过滤推荐算法[J]. 计算机应用, 2012, 32(2): 395-398. DOI: 10. 3724/SP. J. 1087. 2012. 00395.

[5] 黄皓璇, 邢延. 基于用户兴趣变化的 Slope One 协同过滤推荐算法[J]. 工业控制计算机, 2017, 30(7): 112-113, 115. DOI: 10. 3969/j. issn. 1001-182X. 2017. 07. 048.

[6] 龚敏, 邓珍荣, 黄文明. 基于用户聚类与 Slope One 填充的协同推荐算法[J]. 计算机工程与应用, 2018, 54(22): 139-143. DOI: 10. 3778/j. issn. 1002-8331. 1707-0336.

[7] 马瑞敏, 卞艺杰, 陈超, 等. 基于 Hadoop 的电子商务个性化推荐算法: 以电影推荐为例[J]. 计算机系统应用, 2015, 24(5): 111-117.

[8] WHITE T. Hadoop 权威指南[M]. 3 版. 王海, 等译. 北京: 清华大学出版社, 2015: 205-240.

[9] 闫永刚, 马廷淮, 王建. KNN 分类算法的 MapReduce 并行化实现[J]. 南京航空航天大学学报, 2013, 45(4): 550-555. DOI: 10. 3969/j. issn. 1005-2615. 2013. 04. 019.

[10] 李淋淋, 倪建成, 于苹苹, 等. 基于聚类和 Spark 框架的加权 Slope One 算法[J]. 计算机应用, 2017, 37(5): 1287-1291, 1310. DOI: 10. 11772/j. issn. 1001-9081. 2017. 05. 1287.

[11] 原福永, 温志慧, 梁顺攀, 等. 融合项目分类的加权 Slope One 算法[J]. 小型微型计算机系统, 2017, 38(9): 2090-2095. DOI: 10. 3969/j. issn. 1000-1220. 2017. 09. 032.

[12] 尹成祥, 张宏军, 张睿, 等. 一种改进的 *K*-Means 算法[J]. 计算机技术与发展, 2014, 46(1): 30-33.

[13] 徐鹏程, 王诚. *K*-Means 算法改进及基于 Spark 计算模型的实现[J]. 南京邮电大学学报(自然科学版), 2017, 37(4): 113-118. DOI: 10. 14132/j. cnki. 1673-5439. 2017. 04. 018.

[14] 卡劳, 肯维尼斯科, 温德尔, 等. Spark 快速大数据分析[M]. 王道远, 译. 北京: 人民邮电出版社.

[15] 毛典辉. 基于 MapReduce 的 Canopy-*K*means 改进算法[J]. 计算机工程与应用, 2012, 48(27): 22-26. DOI: 10. 3778/j. issn. 1002-8331. 2012. 27. 005.

[16] 王振军, 黄瑞章. 基于 Spark 的矩阵分解与最近邻融合的推荐算法[J]. 计算机系统应用, 2017, 26(4): 124-129. DOI: 10. 15888/j. cnki. csa. 005743.

[17] 汪永旗, 王惠娇. 旅游大数据的 MapReduce 客户细分应用[J]. 华侨大学学报(自然科学版), 2015, 36(3): 292-296. DOI: 10. 11830/ISSN. 1000-5013. 2015. 03. 0292.

[18] 樊晓冬, 王佳斌, 蔡灿辉. ZigBee 室内定位算法中数据预处理[J]. 华侨大学学报(自然科学版), 2018, 39(2): 286-291. DOI: 10. 11830/ISSN. 1000-5013. 201601053.

[19] 刘佳耀, 王佳斌. 基于 Spark 的 *K* 近邻 ALS 的推荐算法[J]. 电脑知识与技术, 2018, 14(11): 6-7, 10. DOI: 10. 14004/j. cnki. ckt. 2018. 1179.

(**责任编辑**: 陈志贤 **英文审校**: 吴逢铁)