

# 基于标签匹配的协同过滤推荐算法研究

马婉贞, 钱育蓉

(新疆大学 软件学院, 新疆 乌鲁木齐 830000)

**摘 要:**随着微博用户数量的上升,微博信息量成倍增长,基于冗杂的微博信息向微博用户快速推荐感兴趣的好友是不容回避的技术问题。针对这一问题,基于微博大数据,以 Hadoop 为平台, HBase 为基础, MapReduce 为编程框架,提出了基于 Apriori 算法与 Item-based 协同过滤算法的组合算法,并构建了推荐好友系统。该系统通过 Apriori 算法对冗杂的微博内容记录进行频繁项集的计算,得出能表达用户喜好的标签,以提升系统的时间性能;通过 Item-based 算法对标签进行匹配推荐,以缩短系统的推荐时间以及资源占用率。为了验证所构建系统的有效性和可靠性,分别进行了两组对比实验,第一组实验为添加了 Apriori 算法的协同过滤算法与传统协同过滤算法在时间性能方面的对比测试,第二组实验则为 Apriori 算法混合 Item-based 协同过滤算法与混合  $K$ -means 算法的对比测试。实验结果表明,在庞大的微博容量下,与传统协同过滤算法相比,所提出算法的运行时间缩短了 24%~44%;与混合  $K$ -means 聚类算法相比,所提出算法在算法运行时间和 CPU 占用率均有 1.2~1.5 倍的提升。可见,提出的算法可显著缩短推荐时间,减少资源消耗率,提高推荐效率。

**关键词:**协同过滤算法;标签计算;Hadoop;MapReduce;标签匹配

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2017)07-0025-04

doi:10.3969/j.issn.1673-629X.2017.07.006

## Investigation on Collaborative Filtering Recommendation Algorithm with Tag Matching

MA Wan-zhen, QIAN Yu-rong

(School of Software, Xinjiang University, Urumqi 830000, China)

**Abstract:** With the rising of micro-blogging users, microblog information capacity has grown rapidly. Fast recommendation of interested friends for micro-blogging users based on the jumbled microblog information becomes inevitable problem. Therefore faced with massive data of microblog, with Hadoop as platform and MapReduce as program frame and based on HBase, a hybrid algorithm of Apriori & Item-based collaborative filtering recommendation algorithm has been proposed and a recommended friends system has been established, in which system computation of frequent item set with massive microblog content records has been conducted to express users' favorites with tags for promotion of its time performances via Apriori algorithm and thus recommendation of tags has been matched via Item-based algorithm for decrease of recommendation time and occupancy rate of system resource. In order to verify its effectiveness and reliability, two groups of contrast experiments have been conducted, in which the first one involves contrast tests of time performances with collaborative filtering algorithm based on Apriori algorithm vs traditional collaborative filtering algorithm and the other one is composed of contrast tests of hybrid algorithm combined Apriori algorithm with Item-based collaborative filtering algorithm vs hybrid  $K$ -means algorithm. The results of contrast experiments show that in large micro-blogging capacity, compared with hybrid  $K$ -means clustering algorithm, the proposed algorithm has decreased the running time by 24%~44% and has lifted 1.2~1.5 times in operation time and CPU occupancy rate. Obviously, the time and recommended resource consumption can be greatly reduced and efficiency recommended improved for proposed algorithm.

**Key words:** collaborative filtering algorithm; tag computing; Hadoop; MapReduce; tag matching

## 0 引言

目前,推荐系统<sup>[1]</sup>在电子商务、信息检索以及移动

应用、互联网广告等众多领域中取得了较大进展,其中协同过滤推荐算法应用较为广泛<sup>[2]</sup>。协同过滤是通过

收稿日期:2016-08-27

修回日期:2016-12-01

网络出版时间:2017-06-05

基金项目:国家自然科学基金资助项目(61562086,61462079,61363083,61262088);新疆“万人计划”后备项目(wr2015bj01)

作者简介:马婉贞(1992-),女,硕士,CCF 会员,研究方向为高性能计算;钱育蓉,博士,副教授,CCF 高级会员,研究方向为网络计算和遥感图像处理。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20170605.1510.072.html>

将过滤操作在一大群人中扩散,用于过滤大量信息的一种机制。协同过滤推荐算法就是基于这种过滤操作而产生推荐的算法。根据过滤操作对象的不同,协同过滤算法可以分为基于用户(User-based)<sup>[3]</sup>和基于项目(Item-based)<sup>[4-5]</sup>的算法。User-based 协同过滤推荐基于用户的历史资料,找出相近的用户作为相似用户进行推荐<sup>[6]</sup>。

在目前的社交网络中,用户的历史资料随着用户的与日俱增变得稀疏且冗杂,无法获知用户的偏好,导致其性能下降。Item-based 协同过滤推荐则是将计算用户间的相似度转化为计算从历史资料中抽取的项目间的相似度,项目间相似度可离线计算,这也在一定程度上解决了推荐系统的实时性问题。而从历史资料中抽取能维持项目间相似度稳定性的项目,使得标签渐渐进入人们的视野,它被用来作为解决数据稀疏性问题的一种有效工具<sup>[7]</sup>。它既能反映出用户的兴趣爱好,又能体现资源特征。针对目前社交网络微博的数据集,鲜有标签字段,因此对于原始资源数据利用 Apriori 算法<sup>[8]</sup>做了处理。

然而对于同样利用项目间相似性来寻找项目群,达到推荐目的的 K-means 聚类算法和 Item-based 算法,传统的聚类算法存在单位时间内处理量小、处理时间较长、难以达到预期效果的缺陷<sup>[9]</sup>;传统的 Item-based 算法则存在处理时间较长、内存量消耗大等问题<sup>[10]</sup>。

因此,将聚类分析或协同过滤算法与并行计算相结合,成为一个比较流行的研究思路。例如,文献[11]实现了基于 Hadoop 的并行 K-means 算法;文献[12]实现了基于 Hadoop 的并行 Item-based 算法,解决了处理时间较长等问题。而 Item-based 算法本身具有项目相似性比较稳定、推荐结果较为准确、实时性较好等优点。

为此,采用并行 Item-based 算法,对标签数据进行匹配,找到与目标项目相似的项目群,通过计算用户间的相似性来产生推荐。为验证提出的并行混合 Item-based 协同过滤算法的有效性及其性能,与基于相同项目群可达到相同推荐目的的并行 K-means 聚类算法进行了对比实验。

## 1 标签的计算与匹配

### 1.1 基于频繁项集的标签计算

在推荐系统中,针对一个用户 U,有 n 条微博内容的集合  $I = \{I_1, I_2, \dots, I_n\}$ 。面对数据量大且冗杂的集合 I,想要对 Item-based 协同过滤算法进行有效运用,采用了通过计算频繁项集得出频繁出现的关键词,设为标签,作为 Item-based 协同过滤算法的有效输入

数据。

Apriori 算法作为最有影响力的挖掘布尔关联规则的频繁项集的算法之一<sup>[13]</sup>,采用逐层搜索的迭代方法,简单、易理解、数据要求低,因此采用它计算频繁项集,推荐有效标签。

算法描述如下:

定义:

支持度:一个项集的支持度定义为数据集中包含该项集的记录所占的比例。

Apriori 规则 1:任一频繁项集的所有非空子集也必须是频繁的。

Apriori 规则 2:计算的支持度不低于用户设定的最小支持度阈值 minSup。

输入:数据集 D,最小支持度阈值 minSup;

输出:数据集 D 中的频繁项集  $L_2$ 。

步骤 1:经过算法的第一次迭代,对数据集 D 进行一次扫描,计算出 D 中所包含的每个项目出现的次数,生成候选 1-项集的集合  $C_1$ ;

步骤 2:根据设定的最小支持度 ( $\text{minSup} = 0.5$ ),从  $C_1$  中根据 Apriori 规则 2 确定频繁 1-项集  $L_1$ ;

步骤 3:由  $L_1$  产生候选 2-项集  $C_2$ ,然后扫描数据库,对  $C_2$  中的项集进行计数;

步骤 4:根据最小支持度,从候选集  $C_2$  中根据 Apriori 规则 2 确定频繁集  $L_2$ 。

### 1.2 基于标签匹配的用户相似性计算

针对标签的匹配<sup>[14]</sup>,采用基于谷本系数的相似性度量(Tanimoto Coefficient-based Similarity):设标签项目集为 X 和 Y,此处标签项目集为标签的字数,例如标签 X 为伤感,标签 Y 为悲伤,那么标签 X 与标签 Y 就可以因为“伤”字进行匹配度计算。标签 X 和标签 Y 的匹配为:

$$\text{Jaccard}(X, Y) = \frac{X \cap Y}{X \cup Y} \quad (1)$$

从式(1)即可以看出其值等于两个用户共同关联(不管喜欢还是不喜欢)的物品数量除于两个用户分别关联的所有物品数量,也就是关联的交集除于关联的并集。

对于计算用户间的相似度则采用皮尔逊相关系数(Pearson Correlation Coefficient):

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (2)$$

其中, r 为相关系数。设用户 X 和用户 Y,因为这是根据标签匹配上的个数的多少来进行推荐的,所以将  $\bar{X}$  和  $\bar{Y}$  设为 0。

## 2 基于标签匹配协同过滤算法的并行

### 2.1 HBase 表的设计

在 HBase 中需要创建如下四个数据表:

(1)matching 表:包含了所有用户对标签的偏好表,它是以用户 id 为主键,列值为标签 id。

(2)tags 表:包含了所有标签的信息,以标签 id 为主键,列值为标签名。

(3)neighbours 表:存储了所有用户的最近邻居的表,它是以用户 id 为主键,包含了一个列簇,这个列簇记录了与目标用户对比达到相似度的用户,每个列的列名即为邻居用户的用户 id,列值为目标用户与该邻居用户的相似度值。

(4)recommends 表:存储了针对客户端选择的 recommend size 大小的推荐用户列表,它是以目标用户 id 为主键,定义一个 recommend 列簇,列簇中包含 recommend size 大小的列,列名为推荐用户的 id,列值为此推荐系统为目标用户推荐的推荐用户的相似度值。

### 2.2 基于标签匹配的协同过滤算法

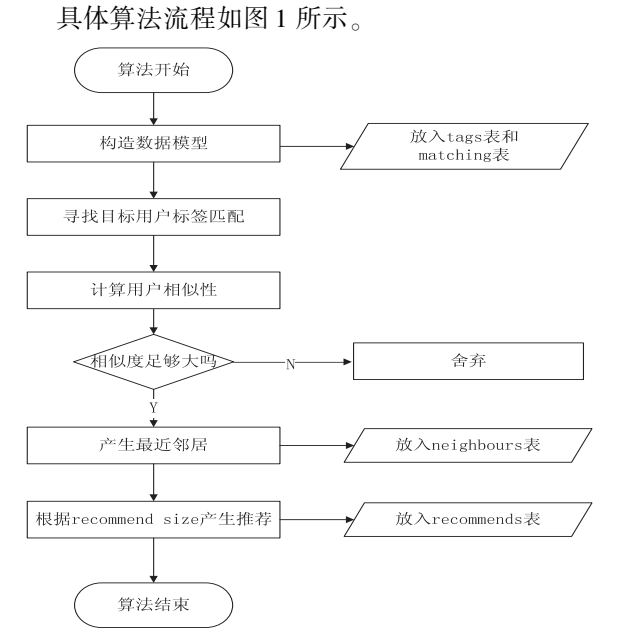


图 1 算法流程图

## 3 实验结果及分析

### 3.1 实验数据集

采用的实验数据集是北京理工大学网络搜索挖掘与安全实验室发布的微博数据集 (<http://www.nlpir.org/>),其中包括 100 万条博主的各种信息(内部 id、性别、家庭住址、粉丝数目、微博数量、关注数量等),23 万条微博内容(文章编号、文章内容、评论数、时间、来源、所属人物 id 等)。

### 3.2 实验环境的配置

Hadoop 集群分为三种模式:完全分布式、伪分布

式以及单机模式。所搭建的是伪分布式集群,计算机既是 Master 又是 Slave,即在一台机子上同时运行着 NameNode、DataNode、JobTracker 和 TaskTracker 四个进程。此实验中节点配置如表 1 所示。

表 1 Hadoop 集群节点配置参数	
配置项	值
CPU	I7-4790 3.6 GHz
内存 RAM/GB	4
操作系统 OS	Ubuntu14.04 LTS
硬盘容量/GB	80
内网带宽/(M/s)	100
Hadoop 版本	Hadoop-1.0.4
HBase 版本	HBase-0.94.18

为验证采用 Apriori 算法混合 Item-based 协同过滤算法为目标用户推荐用户的正确性及有效性,做了两组实验。

### 3.3 混合 Item-based 协同过滤算法与传统协同过滤算法的比较

这组实验的目的是为了针对标签计算模块来比较传统协同过滤算法与提出的 Apriori 混合 Item-based 协同过滤算法的不同推荐性能,通过比较来验证采用 Apriori 混合 Item-based 协同过滤算法推荐的时间性能。实验数据集按照导入的规模分成四组,分别为 10.13 万、14.67 万、18.19 万、23 万条微博内容记录,结果如图 2 所示。

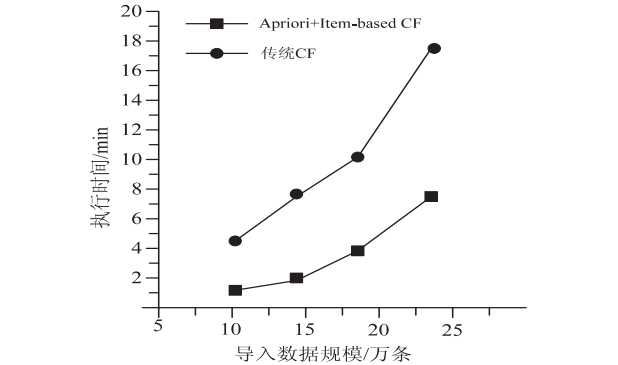


图 2 标签计算结果对比图

由图 2 可知,在数据规模较小时,如 10.13 万条时,两种算法的时间性能仅相差了 25%,但随着数据规模渐渐增大,两种算法的时间性能差成倍增长,并且传统协同过滤算法在数据规模大于 20 万条时呈指数级增长。这是由于传统协同过滤算法在提取标签时,采用将微博内容利用分词器进行分词得出关键字,对关键字进行遍历匹配,产生推荐。因此当数据规模越来越庞大且冗杂时,在算法执行的时间性能方面,所提出的算法明显优于传统协同过滤算法。

### 3.4 Item-based 算法与 K-means 算法的比较

众所周知,在评价两种算法的推荐性能时,除了对

比 MAE、精确率等指标外,还有一种更常用、更直观方便的对比方式—算法的执行时间;但对于能够实现同一目的的不同算法所设计出的程序,其 CPU 的占用率也是各不相同的。因此这组实验将从算法的执行时间和 CPU 占用率两方面来比较 Apriori 混合  $K$ -means 聚类算法与所提出的 Apriori 混合 Item-based 协同过滤算法,通过比较来验证采用 Item-based 协同过滤算法进行标签匹配从而为目标用户推荐用户的推荐性能。实验结果如图 3 和图 4 所示。

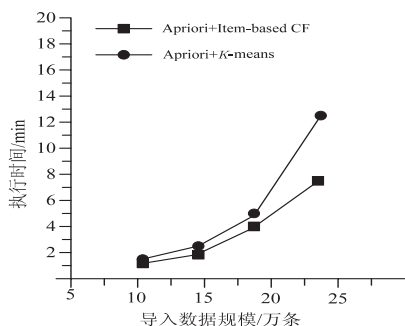


图 3 执行时间对比图

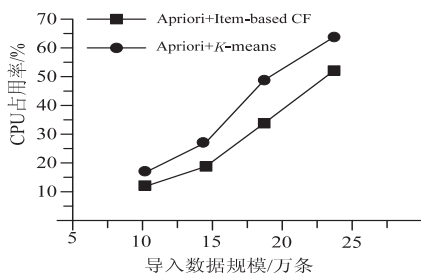


图 4 CPU 占用率对比图

从图中可以看出,对比混合  $K$ -means 聚类算法,所提出的混合 Item-based 协同过滤算法都获得了更低的执行时间和 CPU 占用率,推荐性能更好。在算法的执行时间上,随着数据规模的增大,两种算法的执行时间也都成倍增长,但混合  $K$ -means 聚类算法总是高于混合 Item-based 协同过滤算法,如最初导入 10.13 万条规模的数据时,混合  $K$ -means 与混合 Item-based 协同过滤算法相差 1.2 倍;数据规模到达 23 万条时,混合  $K$ -means 与混合 Item-based 协同过滤算法依然相差将近 1.5 倍。因此,所提出的混合 Item-based 协同过滤算法获得了更好的时间性能。在算法的 CPU 占用率上,很明显,在数据规模达到 14.67 万条后,两种算法的 CPU 占用率均呈指数级增长趋势,这是因为 Apriori 算法采用了 MapReduce 编程框架,随着数据规模的增加,资源消耗就会呈指数级增加,尽管如此,在不同的数据规模下,混合 Item-based 协同过滤算法依旧低于混合  $K$ -means 聚类算法。因此,通过实验结果可得出,混合 Item-based 协同过滤算法的推荐性能均比混合  $K$ -means 聚类算法有所提升。

万方数据

## 4 结束语

针对基于冗杂的微博信息如何向微博用户快速推荐感兴趣的好友的问题,以 Hadoop 为平台, HBase 为基础, MapReduce 为编程框架,提出了基于 Apriori 算法与 Item-based 协同过滤算法的组合算法,并进行了对比验证。实验结果表明,在大数据集处理速度与资源消耗率上,该算法可显著缩短推荐时间,减少资源消耗率,提高推荐效率。

### 参考文献:

- [1] 许海玲, 吴 潇, 李晓东, 等. 互联网推荐系统比较研究[J]. 软件学报, 2009, 20(2): 350-362.
- [2] 蔡 强, 韩东梅, 李海生, 等. 基于标签和协同过滤的个性化资源推荐[J]. 计算机科学, 2014, 41(1): 69-71.
- [3] 程 飞. 基于用户相似性的协同过滤推荐算法研究[D]. 北京: 北京交通大学, 2012.
- [4] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th world wide web conference. New York: ACM Press, 2001: 285-295.
- [5] 燕 存, 吉根林. Item-Based 并行协同过滤推荐算法的设计与实现[J]. 南京师大学报: 自然科学版, 2014, 37(1): 71-75.
- [6] 吴泓辰, 王新军, 成 勇. 基于协同过滤与划分聚类的改进推荐算法[J]. 计算机研究与发展, 2011, 48(8): 205-212.
- [7] 王金辉. 基于标签的协同过滤稀疏性问题研究[D]. 合肥: 中国科学技术大学, 2011.
- [8] Song Y, Zhang L, Giles C L. Automatic tag recommendation algorithms for social recommender systems[J]. ACM Transactions on the Web, 2011, 5(1): 1-31.
- [9] Ekanayake J, Pallickara S, Fox G. MapReduce for data intensive scientific analysis[C]//IEEE fourth international conference on escience. Piscataway: IEEE, 2008: 277-284.
- [10] Schafer J B, Frankowski D, Herlocker J, et al. Collaborative filtering recommender systems[M]. Berlin: Springer, 2007: 291-324.
- [11] 周丽娟, 王 慧, 王文伯, 等. 面向海量数据的并行 KMeans 算法[J]. 华中科技大学学报: 自然科学版, 2012( S1 ): 150-152.
- [12] Jiang J, Lu J, Zhang G, et al. Scaling-up item-based collaborative filtering recommendation algorithm based on Hadoop[C]//2011 IEEE world congress on services. Washington: IEEE, 2011: 490-497.
- [13] 黄创光, 印 鉴, 汪 静, 等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报, 2010, 33(8): 1369-1377.
- [14] 张 斌, 张 引, 高克宁, 等. 融合关系与内容分析的社会标签推荐[J]. 软件学报, 2012, 23(3): 476-488.