基于用户推荐影响度的并行协同过滤算法

王 硕[1] 孙光明[2] 邹静昭[3] 李伟生[2]
(河北科技大学信息科学与工程学院 石家庄 050035)[1]
(北京交通大学计算机与信息技术学院 北京 100004)[2] (河北中医学院公共课教学部 石家庄 050200)[3]

摘 要 基于共同评分与项目全集的相似度未甄别近邻的推荐影响力,导致推荐质量低,可扩展性差。为此,提出了一种基于推荐影响度的并行协同过滤算法。该算法通过非共同评分项目、共同评分项类以及用户访问次数来计算用户推荐新颖度与兴趣重合度以度量用户推荐能力,并融入相似性计算来抑制相似度高但推荐力不强的用户,避免在项目全集上计算相似度,从而提高推荐质量;通过MapReduce并行化,使其具备良好的实时性和可扩展性。实验结果表明,该算法在海量数据集上的推荐质量更高,可扩展性更强。

关键词 推荐影响度,推荐新颖度,兴趣重合度,MapReduce并行化
中图法分类号 TP312 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.09.047

Parallel Collaborative Filtering Algorithm Based on User Recommended Influence

WANG Shuo[1] SUN Guang-ming[2] ZOU Jing-zhao[3] LI Wei-sheng[2]
(School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050035, China)[1]
(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100004, China)[2]
(Department of Public Course Teaching, Hebei University of Chinese Medicine, Shijiazhuang 050200, China)[3]

Abstract The similarity based on common scores and full item sets has failed to identify the nearest neighbor recommendation influence, which brings about lower recommend quality and poor scalability. Through non-common rating items, common score item categories and user visited times, this paper proposed a parallel collaborative filtering algorithm based on user recommendation influence. It computes the user recommended novelty degree and interest coincidence to measure user recommendation influence ability. By adding it to calculate similarity, the algorithm can effectively restrain the highly recommended users with high similarity, avoid similarity computation on full item sets and improve the quality of recommendation. Further more, by using MapReduce parallelization, this algorithm has good real-time performance and scalability. The experimental results show that the parallel algorithm is of higher recommendation quality and better scalability on big data.

Keywords Recommendation influence degree, Recommendation novelty degree, Interest coincidence degree, MapReduce paralleliation

随着移动互联网、物联网的迅猛发展,Web用户及应用的规模爆发式增长,而传统基于关键字的搜索引擎难以提供个性化信息服务,"信息过载"现象严重。推荐算法通过用户信息并基于网络群智思想预测用户偏好,能较好地解决该问题。其中,协同过滤推荐算法最为成功且应用广泛。但是,由于用户缺少评分习惯,且用户、项目数量的大规模化,导致评分矩阵越来越稀疏,产生的近邻往往与实际不符,而且,传统的单节点计算模式也难以满足算法的实时性与可扩展性需求。

针对于此,研究人员从提高评分密度及优化相似度计算的角度提出了诸多改进算法。文献[1-3]利用项目并集来增加评分数量和项目类别,缩小了评分范围,提高了评分密度,克服了数据稀疏导致的相似性计算不准确的问题;文献[4-5]基于云模型计算用户或项目整体特征的相似性,避免了当前严格匹配对象属性的相似度因数据稀疏而不准确的问题,并通过用户双重聚类计算进一步提高了推荐准确度,降低了时间开销;文献[6-8]利用概率矩阵分解模型降低评分矩阵维度,减少数据稀疏性,预填充未评分项,提高了推荐精度与效率;文献[9-11]通过社会网络中用户间的信任或朋友关系、对象间的关联关系来弥补评分信息的不足,并利用复杂网络分析、矩阵分解、最小二乘法等优化相似性计算,提高推荐准确度,这些研究在一定程度上降低了评分数据的稀疏度,提高了算法的推荐质量。但是,它们忽视了相似度很高的用户的推荐能力却不一定强的客观事实。如图1所示,目标用户与其"粉丝"相似度很高,但"粉丝"却不一定能为其提供新颖可接受的推荐信息。另一方面,不同近邻与目标用户的兴趣重合

到稿日期:2016-08-13 返修日期:2016-11-09 本文受河北省高等学校科学技术研究重点项目(ZD2014061),青年基金项目(QN2016108)资助。
王 硕(1979—),女,硕士,讲师,主要研究方向为推荐系统;孙光明(1979—),男,硕士,讲师,主要研究方向为数据挖掘、分布式计算,E-mail:sungmwhu@whu.com;邹静昭(1979—),女,硕士,讲师,主要研究方向为数据挖掘;李伟生(1945—),男,教授,主要研究方向为分布并行计算。

程度也不尽相同，但现有研究在计算相似度时未区别对待，弱化了具有强推荐能力的用户的作用。

针对上述问题，本文提出了一种基于推荐影响度的并行协同过滤算法。该算法在现有基于评分交集的相似度的基础上，通过准近邻与目标用户的非共同评分项目及其所属类别定义用户推荐新颖度，以度量准近邻的推荐影响力；结合共同评分项类及不同用户对相同项目的访问次数，定义用户兴趣重合度，以度量不同用户与目标用户兴趣的一致性；将用户推荐新颖度和兴趣重合度作为推荐影响力约束因子融入相似性计算中，抑制相似度虚高用户的负面推荐作用，并利用项目类别避免在项目全集上计算相似度时存在的大量不必要的运算，提高计算效率。另一方面，利用 MapReduce 实现算法的并行化，克服海量推荐数据环境下单节点计算模式实时性与可扩展性差的问题。

1　相关工作

协同过滤推荐算法一般分为 3 步：建立用户-项目评分矩阵、计算相似度并产生近邻列表、依据近邻预测用户对项目的评分并产生推荐。其中，相似性计算的准确度直接关系到推荐质量的优劣，成为了研究热点。

1.1　传统相似度计算方法

用户-项目评分矩阵是传统协同过滤算法计算相似度的依据，一般使用启发式模糊聚类统计方法进行计算，主要包括余弦相似度、修正余弦相似度和相关相似度 3 种[12]。

1)余弦相似度。将用户对项目的评分作为 n 维空间的向量，表示用户的兴趣特征。用户间的相似度通过向量间夹角的余弦来计算，如式(1)所示：

$$Sim(u,v)=\frac{\sum_{i\in I_u\cap I_v} r_{u,i}\cdot r_{v,i}}{\sqrt{\sum_{i\in I_u} r_{u,i}^2}\sqrt{\sum_{i\in I_v} r_{v,i}^2}} \tag{1}$$

2)修正余弦相似度。余弦相似度没有考虑用户评分标准的不同，即用户倾向评高分还是低分。为此，修正余弦相似度通过减去评分均值来弥补余弦相似度的不足，计算公式如式(2)所示：

$$Sim(u,v)=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i\in I_u}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum_{i\in I_v}(r_{v,i}-\overline{r_v})^2}} \tag{2}$$

3)相关相似度。两用户兴趣特征向量的线性相关性用 Pearson 相关系数来计算，如式(3)所示：

$$Sim(u,v)=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum_{i\in I_u\cap I_v}(r_{v,i}-\overline{r_v})^2}} \tag{3}$$

与修正余弦相似度不同，相关相似度仅计算用户评分交集中的项目。

由上述 3 种常用的相似度计算方法可知，它们均以评分交集为计算依据，在数据稀疏时存在分母为零而不能计算的问题。

1.2　改进的相似性计算及存在的问题

与用户和项目规模的指数级增长相反，用户评分不超过 1%[1]，共同评分则少之又少，评分矩阵极其稀疏。对于依赖共同评分计算相似度的传统相似性计算方法而言，即使余弦和修正余弦相似度通过缺省值填充，但因其不能反映用户的真实偏好，产生的近邻往往与实际情况不符，使得算法的推荐质量急剧下降[12]。

针对于此，文献[1]提出利用用户对相似项目的评分来预测评分并集中未评分项目分值的方法，提高了近邻计算的准确度。但是，评分项目并集中存在与目标用户偏好毫不相关的用户，而文献[1]并未将其剔除，浪费了大量的计算时间。文献[13]对此做了改进，将评分并集用户区分为无推荐能力和有推荐能力两种，并通过“领域最近邻”计算有推荐能力用户的相似性，提高了算法效率。但文献[13]中的用户推荐能力通过设定的评分阈值来度量，评分虚高而无实际推荐能力的“粉丝”、“跟风”用户等仍难以滤掉。此外，选取能客观鉴别用户推荐能力的评分阈值也比较困难。文献[14]通过准近邻评分项目中目标用户未评分项目的个数定义其推荐贡献度，消除了“跟风”类用户对相似性计算的影响。然而，这些准近邻评过分但目标用户未评分的项目本身就可能不是目标用户的兴趣对象，基于这些项目定义的推荐贡献度因子一般为 0，反而削弱了准近邻的与目标用户间的实际相似性。

针对上述问题，本文通过目标用户与准近邻的非共同评分项目、共同评分项类及访问次数等偏好差异，定义准近邻推荐新颖度与兴趣重合度来度量其推荐影响力，并利用 MapReduce 并行化，使算法满足海量推荐数据环境下的实时性和可扩展性要求。

2　基于推荐影响度的并行协同过滤算法

具有相同兴趣的用户间的不同偏好能够为彼此提供新颖的、有价值的推荐参考对象，反映了用户的推荐影响力，可辨别出基于共同评分的相似度高但推荐能力不强的用户，扩展了相似度计算的用户及项目范围，缓解了数据稀疏情况下共同评分不足导致相似度无法计算的困境。

准近邻非共同评分项目反映了具有相似兴趣的用户间的偏好差异。以此为切入点，结合项目类别，提出用户推荐新颖度计算公式，以度量近邻的推荐影响力；通过共同评分项类及不同用户对相同项目的访问次数提出用户兴趣交叉度及计算公式，以度量不同近邻与目标用户的兴趣的差异度；将二者融入相似性计算，抑制相似度虚高用户的负面推荐作用，避免在项目全集上计算相似度时存在的大量不必要的运算。最后，实现算法的 MapReduce 并行化，并使其适应推荐数据日趋大数据化的实时性与可扩展性要求。

2.1　用户推荐新颖度及计算方法

如图 1 所示，准近邻 v 的非评分交集 I_{vuv} 中的项目可能为目标用户 u 推荐新颖且感兴趣的对象，体现了用户 v 的推荐影响力。但是，I_{vuv} 中也可能存在用户 u 根本不感兴趣的项目，它们不会为用户 v 增加任何推荐影响度。因此，如何从 I_{vuv} 中将这些项目甄别出来，直接关系到用户 v 对 u 的推荐影响力。

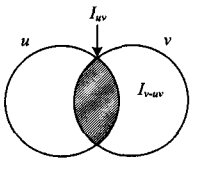

图 1　用户 u 与 v 的评分情况示意图

一般地，如果 I_{v-uv} 中存在能为用户 u 推荐感兴趣对象的项目，即使用户 u 因缺少评分习惯等因素未给该项目评分，该项目也应该包含在用户 u 的兴趣范围内。而用户 u 已评分的项类能够体现其兴趣范围。因此，本文基于准近邻非评分交集与目标用户项目类别来定义用户推荐新颖度以度量用户的推荐影响力。

定义 1(用户推荐新颖度) 设用户 u 的评分项类集合 $C_u=\{c_{u1},c_{u2},\cdots,c_{um}\}$，$m$ 为用户 u 评分项目所属项类总数；用户 v 与 u 的非共同评分项目交集 I_{vuv} 中项目类别集合为 $C_{vuv}=\{c_{vuv1},c_{vuv2},\cdots,c_{vuvn}\}$，$n$ 为 I_{vuv} 中项目所属项类总数，则 I_{vuv} 中所有项目隶属于项类 C_u 的总数与其在项类 I_{vuv} 中总项目数的比值称为用户 v 对用户 u 的推荐信新颖度，记为 N_v。

即设 q 为用户 v 的非评分交集 I_{vuv} 中项目的个数，则对于 $\forall I_j\in I_{vuv}(1\leqslant j\leqslant q)$，$QI_{uij}$ 为项目 I_j 隶属于项类 $C_{ui}(1\leqslant i\leqslant m)$ 的个数，QI_{vuvkj} 为项目 I_j 隶属于项类 $C_{vuvk}(1\leqslant k\leqslant n)$ 的个数，则 N_v 的计算方法如式(4)所示：

$$N_v=\frac{\sum_{j=1}^{q}\sum_{i=1}^{m}QI_{uij}}{\sum_{j=1}^{q}\sum_{k=1}^{n}QI_{vuvkj}} \tag{4}$$

从式(4)可见，N_v 越大，表明 I_{vuv} 中隶属于目标用户 u 的兴趣领域的项目越多，越能给 u 推荐感兴趣的新颖项目，用户 v 对用户 u 的推荐影响力越强。值得注意的是，N_v 有效过滤了直接使用 I_{vuv} 中的项目数量度量用户 v 的推荐能力时用户 u 毫不感兴趣的无效项目。另一方面，当用户 v 为用户 u 的“粉丝”用户时(见图 2)，用户 v 的非评分交集 I_{vuv} 为空集，$N_v=0$，即用户 v 与用户 u 虽然有很高的评分交集相似度，但用户 v 对 u 的推荐新颖度为 0，表明用户 v 不会为 u 推荐任何有价值的项目；而那些对几乎所有项目都有评分的用户或几乎被所有用户评过分的项目，也不能为目标用户提供任何个性化的推荐。这类用户与目标用户的非评分交集中的所有项目的类别都不包含在目标用户的兴趣领域内，即 $\sum_{j=1}^{q}\sum_{i=1}^{m}QI_{uij}=0$，故 $N_v=0$，此类相似度虚高但推荐能力不强的用户也能被有效过滤掉。

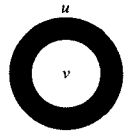

图 2 用户 v 为用户 u 的“粉丝”用户

2.2 用户兴趣重合度及计算方法

一般地，不同近邻与目标用户主观行为的差异反映了用户间兴趣重合程度的不同，而传统的单一评分相似度使用相同的权重同等地度量不同用户间的相似性，忽略了这些主观行为的差异所蕴含的用户偏好取向问题，也使偶然评分相似性[14]有了存在的可能性，导致传统相似性计算产生的近邻往往与事实不符。

共同评分项类中既包含用户间共有的评分项目，也包含各自单独的评分项目，体现了用户兴趣的最大交叉领域。例如，表 1 列出了用户-项目评分表，表 2 列出了项目-项类隶属表，其中，$u,v,w,\cdots,z$ 表示用户；$I_1,I_2,\cdots,I_n$ 表示项目；$C_1,C_2,\cdots,C_n$ 表示项目类别；“√”用于标识用户对项目有评分或项目属于某一类别。由表 1 可知，用户 u 的评分项目集合为 $I_u=\{I_1,I_2,I_3,I_n\}$，用户 v 的评分项目集合为 $I_v=\{I_1,I_2\}$；结合表 2 可知，$I_1\in\{C_1,C_3,C_n\}$，$I_2\in\{C_1,C_2,C_3\}$，$I_3\in\{C_2\}$，故用户 u 的项类集合 $C_u=\{C_1,C_2,C_3,C_n\}$，用户 v 的项类集合 $C_v=\{C_1,C_2,C_3,C_n\}$，u 和 v 的共同评分项类 $CC_{u,v}=\{C_1,C_2,C_3,C_n\}$。$CC_{u,v}$ 包含了用户 u 和用户 v 共同感兴趣的所有项目，项类数、项目个数越多，表明用户 u 和用户 v 的兴趣交叉度越大，他们之间单一评分相似度所占权重就越大。

表 1 用户-项目评分表

	I_1	I_2	I_3	…	I_n
u	√		√	…	√
v		√	√	…	
w	√	√	√	…	
⋮				…	
z				…	

表 2 为项目-项类隶属表

	C_1	C_2	C_3	…	C_n
I_1	√		√	…	√
I_2	√	√	√	…	
I_3		√		…	
⋮				…	
I_n				…	

另外，用户对项目的访问频度体现了用户对项目的偏好度，不同用户对同一项目的访问次数也反映出用户间关于该项目兴趣一致性的程度。为此，本文通过用户兴趣交叉度来度量用户兴趣重合程度，并将其作为相似度性约束因子，以弥补单一评分相似度忽略用户行为所蕴含的兴趣偏好取向的不足。

定义 2(用户兴趣交叉度) 设用户 u 的评分项类集合为 $C_u=\{c_{u1},c_{u2},\cdots,c_{um}\}$，用户 v 的评分项类集合为 $C_V=\{C_{V1},C_{V2},\cdots,C_{Vn}\}$，$u,v$ 共同评分项类集合为 $CC=C_u\cap C_v$，K_C 为 CC 中项目数，N_{CC} 为项类 CC 中项目类别数，N_C 为项目类别总数；$N_{u,i}$，$T_{u,i}$ 为用户对 CC 中项目 i 的访问次数，则用户 u 与用户 v 的兴趣交叉度 $CS_{u,v}$ 的定义如式(5)所示：

$$CS_{u,v}=\frac{N_{CC}}{N_C}\cdot\sum_{i=1}^{K_C}\frac{T_{u,i}+T_{v,i}}{1+|T_{u,i}-T_{v,i}|} \tag{5}$$

从式(5)可以看出，任意两用户的共同评分项类及项目越多，其对相同项目访问越频繁且次数越接近，用户兴趣交叉度越大，这客观反映了评分项类、访问次数等用户主观行为所蕴含的用户偏好一致性程度。

2.3 融合用户推荐新颖度与兴趣交叉度的近邻选择与推荐

为避免使用相同的权重度量目标用户与准近邻间的相似性，进而导致计算的近邻往往与实际不符，本文提出用户推荐新颖度、用户兴趣交叉度及其计算方法，并将用户推荐新颖度和兴趣交叉度作为相似度权重约束因子融入用户间相似性计算，使推荐新颖度高、与目标用户兴趣交叉度大的用户有更高的优先级成为近邻。

Pearson 相关系数以平均评分来区分用户评分的差异，较余弦相似度和修正余弦相似度有更高的准确度。本文以

Pearson 相关系数为基础，融入用户推荐新颖度与用户兴趣交叉度作为权重约束因子，提出相似性计算方法，如式(6)所示：

$$Sim'(u,v)=N_v \cdot CS_{u,v} \cdot \frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum_{i\in I_u\cap I_v}(r_{v,i}-\overline{r_v})^2}} \quad (6)$$

通过相似度计算产生目标用户 u 的 K 最近邻列表 $NH(u)$ 后，使用式(7)预测目标用户对项目的评分，以产生推荐列表。

$$P(u,i)=\overline{r_u}+\frac{\sum_{v\in NH(u)}Sim'(u,v)\cdot(r_{v,i}-\overline{r_v})}{\sum_{v\in NH(u)}Sim'(u,v)} \quad (7)$$

其中，$P(u,i)$ 为用户 u 对项目 i 的预测评分；$\overline{r_u}$，$\overline{r_v}$ 为用户 u，v 基于共同评分项类的评分均值。

2.4　基于推荐影响度的协同过滤算法及其 MapReduce 并行化

设 $MR(u,i)$ 为用户-项目评分矩阵，$MR_{u,i}$ 表示用户 u 对项目 i 的评分；$MT(u,i)$ 为用户-项目访问次数矩阵，$MT_{u,i}$ 表示用户 u 对项目 i 的访问次数；$IC(i,c)$ 为项目-项目类别矩阵，$IC(i,c)=1$ 表示项目 I_i 属于项目类别 C_c，$IC(i,c)=0$ 表示项目 I_i 不属于项目类别 C_c。基于推荐影响度的协同过滤算法表述如下：

1)由 $MR(u,i)$ 计算用户 u 和用户 v 共同评分项目交集 I_{uv}、用户 v 中与用户 u 的非共同评分项目 I_{vuv}；

2)由 $MR(u,i)$ 和 $IC(i,c)$ 分别计算用户 u 和用户 v 的评分项类集合 C_u，C_v 及 I_{vuv} 中的项类集合 C_{vuv}；

3)由 C_u 和 I_{vuv}，依据式(4)计算用户 v 对用户 u 的推荐新颖度 N_v；

4)由 C_u，C_v 及 I_{uv}，计算用户 u 和用户 v 的共同评分项目类别集合 $CC_{u,v}$；

5)由 $MR(u,i)$ 及 $CC_{u,v}$，依据式(5)计算用户 u 和用户 v 的兴趣交叉度 $CS_{u,v}$；

6)由 N_v，$CC_{u,v}$ 及 I_{uv}，依据式(6)计算用户 u 和用户 v 之间融合推荐新颖度与兴趣交叉度的相似性 $Sim'(u,v)$；

7)重复步骤 3)－步骤 6)，产生用户 u 的 Top-K 最近邻 $NLST_{uk}$；

8)由 $NLST_{uk}$ 及 $MR(u,i)$，按式(7)预测目标用户 u 对预推荐项目 i 的评分，并产生推荐列表。

MapReduce[16] 是 Google 提出的处理海量数据的并行编程模型，其成功应用在 Google 新闻推荐系统上。MapReduce 过程分为 Map 和 Reduce 两阶段：首先，按块分割大数据集，Map 阶段负责将这些数据块分发到各节点，并由用户自定义的 map 函数执行相应操作并产生一个(*key*，*value*)队列；当完成所有 Map 任务后，MapReduce 主控进程按 *key* 对输出进行分组，并作为特定 Reduce 任务的输入，由相应的 reduce 函数对其进行组合并输出。在这个过程中，用户只需编写 map 及 reduce 映射函数，实现输入输出键值对间的转换，而任务的装载、调度和节点间的通信则由 MapReduce 自动完成。

作为 MapReduce 的开源实现，Hadoop 平台可由上千台廉价的 PC 组成大规模集群，为其上开发的应用程序提供可靠且兼容的 PB 级大数据并行处理能力。基于 Hadoop 平台实现推荐系统的并行化，能够较好地解决目前大规模用户及项目数据量环境下推荐系统实时性低、可扩展性差的问题。本文算法的 MapReduce 并行化流程如图 3 所示。

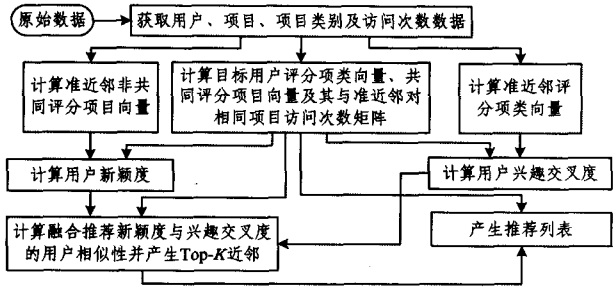

图 3　本文协同过滤算法的 MapReduce 并行化流程

依据算法的 MapReduce 并行化流程，本文提出的并行协同过滤算法由 4 个 MapReduce 作业(Job)实现，如图 4 所示。

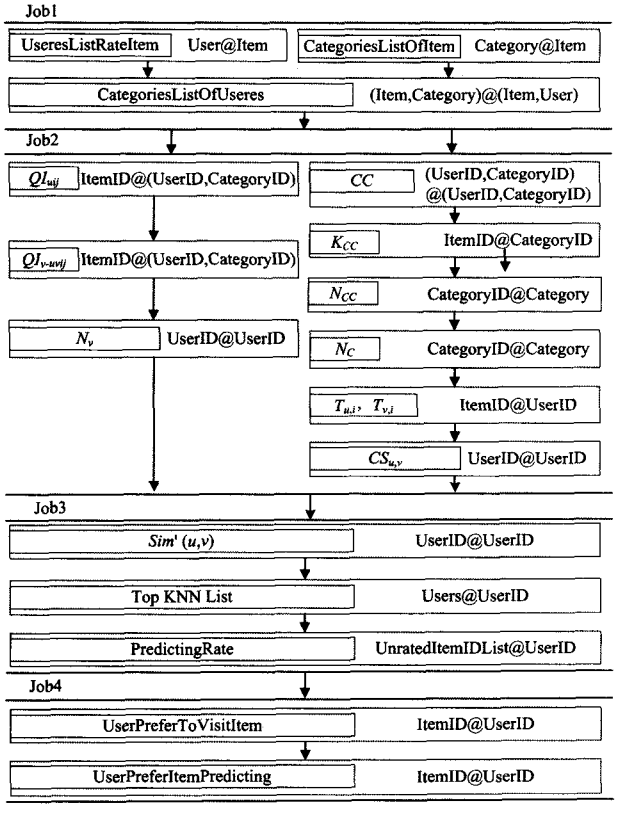

图 4　基于推荐影响度的协同过滤并行算法设计

1)Job1：计算用户、项目、项目类别特征向量，包括用户-项目评分向量、项目-项目类别向量及用户-项目类别向量 3 个计算子任务，该过程由 3 个 MapReduce 程序完成。

2)Job2：计算用户推荐新颖度及兴趣交叉度，通过 9 个 MapReduce 程序完成。

3)Job3：计算用户相似度、用户 Top-K 最近邻并预测未评分项目评分，通过 3 个 MapReduce 程序完成。

4)Job4：计算用户推荐列表，通过 2 个 MapReduce 程序完成。

在 Hadoop 分布式计算环境下，上述 MapReduce 过程中各计算任务相互独立地分散在不同节点上并行执行，并且一

些基于历史数据的计算任务可离线完成，从而有效地解决了当前用户、项目大规模数据集趋势下传统推荐算法的效率瓶颈，满足了对推荐系统实时性、可扩展性越来越高的要求。

3 实验结果与分析

3.1 实验数据集与环境

为了满足本文算法所需的数据稀疏性和大数据环境测试条件，实验采用 Minnesota 大学的 GroupLens 研究组提供的 MovieLens 公开数据集上 3 组不同规模的电影评分数据，数据集情况如表 3 所列。

表 3 实验数据集概况

数据集	用户	电影	评分电影	评分数	稀疏度/%
ml-100K	943	1682	≥20	100000	93.7
ml-1M	6040	3900	≥20	1000209	95.8
ml-10M	71567	10681	≥31	10000054	98.7

此外，为了测试算法的性能，本文使用 MovieLens 自带的脚本程序将数据集按照一定比例随机划分为测试数据集和训练数据集两部分。训练数据集作为用户访问电影的历史记录，测试数据集则用于比对算法的计算结果，并且训练数据集和测试数据集中不能有重复的记录。

实验单机配置为 Intel Core™ i5 双核处理器、8GB 内存、1TB 硬盘。Hadoop 分布式计算环境为 Hadoop2.7.1。为了使实验节点性能略低于当前主流 PC 的配置，以便更好地说明算法的性能，使用 VMware10.0.1 在实验单机上创建 9 个虚拟机来搭建 MapReduce 集群。其中，一台虚拟机作为 NameNode 和 JobTracker 服务 Master 节点，剩余 8 台虚拟机作为 DataNode，负责具体计算任务的 Slave 节点。操作系统为 Ubuntn12.04，开发环境为 jdk-7u80-linux-x64。

3.2 评价标准

本文主要从推荐准确性、可扩展性和实时性 3 个方面对算法进行评价。平均绝对偏差（Mean Absolute Error，MAE）[17]通过计算项目实际评分与预测值间的偏差来直观度量算法的推荐准确性，是当前评价推荐质量最常用的方法，其值越小，推荐质量越高。本文也采用 MAE 来度量算法的推荐准确性。

设$\{r_1,r_2,\cdots,r_n\}$为用户实际评分集，$\{p_1,p_2,\cdots,p_n\}$为预测评分集，r_i,p_i 为用户对项目 $I_i(1\leqslant i\leqslant n)$的评分，则 MAE 定义为：

$$MAE=\frac{\sum_{i=1}^{n}|p_i-r_i|}{n} \tag{8}$$

在 Hadoop 分布式计算环境下，加速比是衡量推荐系统扩展性与实时性的主要性能指标。本文采用加速比来测试算法在 MapReduce 集群下运行时的可扩展性和计算时效性。

3.3 实验结果与分析

本文算法通过推荐影响度来优化传统相似性计算方法。为了验证其推荐的准确度，通过 Hadoop 分布式计算平台，将 MovieLens 中的 ml-1M 数据集按 4:1 的比例随机分为 5 组训练数据集和测试数据集，对本文基于推荐影响度的相似性计算方法、Pearson 相关相似度、修正余弦相似度进行 5 次实验，以避免单次实验可能存在的不确定性对推荐效果的影响。5 次实验中 MAE 的平均值随用户邻居数量的变化情况如图 5 所示。

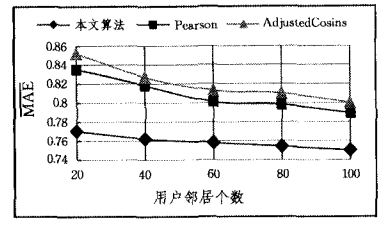

图 5 3 种相似度算法的 MAE 随用户近邻个数的变化情况

由图 5 可知，近邻个数的增多使得评分数据密度增大，导致 3 种相似度算法在 Hadoop 平台上的 MAE 平均值随之减少，与客观实际相符合。同时，由于本文在相似性计算方法融合了用户推荐新颖度和兴趣重合度，消去了传统相似度计算中相似性较高而实际推荐影响力较弱的近邻，使得推荐更准确，并且本文算法具有最小的 MAE 值（在图 5 中得到了证明）。值得注意的是，近邻个数的增加能够提高算法的推荐准确度，但也降低了算法的计算效率。实验表明，在本文计算环境下近邻个数为 80 时较为合适。

另一方面，为了测试本文算法在不同数据规模及不同稀疏程度环境下的性能，在 ml-1M，ml-10M 两个数据集上，用户最近邻固定为 60 个，通过调整训练数据集与测试数据集的比例，对比本文算法与文献[13]算法的 MAE 值，实验结果如图 6、图 7 所示。

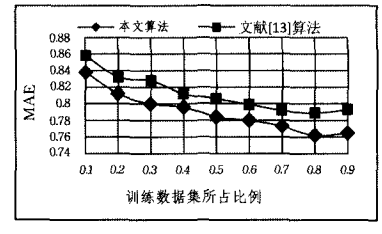

图 6 ml-1M 训练集比例对 MAE 的影响

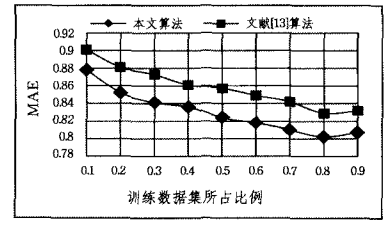

图 7 ml-10M 上训练集比例对 MAE 的影响

图 6、图 7 表明，在 ml-1M，ml-10M 两组不同规模的数据集上，两种推荐算法的 MAE 值都随着训练数据集所占比例的增大而减小，推荐准确性更高。当训练数据集所占比重为 80%时，推荐最为准确，并且本文算法较文献[13]算法在 Hadoop 集群上拥有更高的推荐质量。

为了验证本文的推荐算法在 MapReduce 并行化后在大数据环境下实时性和可扩展性的优势，分别在单机及 Hadoop 集群上将 ml-100K，ml-1M，ml-10M 数据集按 4 ∶1 的比例分为训练集和测试集，对本文算法进行实验。在单机及 Hadoop 平台上的实验结果如表 4 所列。

表 4　单机及 Hadoop 集群平台上算法计算性能的比较

数据集	单机		Hadoop 集群(9 节点)		
	计算时间/min	MAE	计算时间/min	MAE	加速比
ml-100K	16	0.738	34	0.744	2.96
ml-1M	403	0.756	98	0.758	3.18
ml-10M	内存溢出	—	1413	0.802	2.39

从表 4 可以看出，算法在单机及 Hadoop 集群环境上对相同数据集进行推荐的 MAE 值处于同一数量级，这表明本文算法的 MapRedece 并行化在 Hadoop 上保持了较好的推荐准确性。另一方面，在单机环境下随着数据集规模的突增，算法的计算时间显著增加，并且在 ml-10M 数据集上出现内存溢出而不能计算的情况；在 MapReduce 集群上计算时间变化不大，不存在因数据规模增大而出现不可计算的问题。这说明本文算法受数据集规模的影响不大，具有良好的可扩展性，并且本文算法在大数据环境下的计算效率明显优于在单机上串行执行的效率。值得注意的是，在较小的数据集上，本文算法的计算效率要低于单机上串行执行的效率，主要是因为存在集群启动、节点间通信的时间消耗，尤其在算法运行时间占整个计算时间的比例较小时这种现象更加明显。

在 ml-1M 及 ml-10M 大规模推荐数据环境下，本文算法的时效性优势及集群节点数对算法效率的影响情况如图 8 所示。

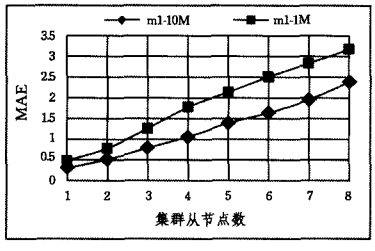

图 8　不同数据集加速比随 Hadoop 集群节点数的变化情况

加速比是衡量并行计算系统可扩展性的一个主要参数指标。由图 8 可知，在不同数据集规模下，本文算法的加速比随 Hadoop 集群节点个数的增加而增大，这表明增加节点可以明显提升本文算法对不同大规模海量推荐数据的处理能力，提高算法的计算效率。但是，当节点数到达 6 时，加速比提升趋势减缓。另一方面，算法的计算效率也未随节点个数的增加而成比例的提升，主要原因在于通过增加节点提高处理能力的同时，整个集群的通信消耗也随之增大。因此，不能为了提升系统的推荐效率，一味地增加节点数据，否则将降低系统的性价比，效率也得不到有效提升。

结束语　本文利用用户非评分交集中项目所体现的用户推荐新颖性及评分项类交集来反映用户兴趣最大重合度的特性，提出了融合用户推荐新颖度及兴趣重合度的个性化协同过滤算法，并在 Hadoop 集群实现了其 MapReduce 并行化。实验结果表明，本文算法较好地解决了大数据推荐环境下协同过滤算法因数据稀疏导致的相似性计算不准确及单机上计算效率低与可扩展性差的问题。如何获取更多关联用户兴趣的有效数据是进一步提高推荐质量的研究方向，例如，利用社会网络中用户的关系数据来提高相似度计算的准确性。同时，推荐数据具有一定的结构特征，如何利用这些特征优化 MapRuduce 流程，提升算法的并行度和可扩展性，都是下一步的研究目标。

参 考 文 献

[1] DENG A L,ZHU Y Y,SHI B L. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Journal of software,2013,14(9):1621-1628. (in Chinese)
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.

[2] WEI S Y,YE N,JI G L,et al. Collaborative filtering recommendation algorithm based on item category and interest[J]. Journal of Nanjing University(Natural Sciences),2013,49(2):142-149. (in Chinese)
韦素云，业宁，吉根林，等. 基于项目类别和兴趣度的协同过滤推荐算法[J]. 南京大学学报(自然科学版)，2013，49(2)：142-149.

[3] HAN Y N,CAO H,LIU L L. Collaborative filtering recommendation algorithm based on score Matrix filling and user interest [J]. Computer Engineering,2016,42(1):36-40. (in Chinese)
韩亚楠，曹菡，刘亮亮. 基于评分矩阵填充与用户兴趣的协同过滤推荐算法[J]. 计算机工程，2016，42(1)：36-40.

[4] CHEN P H,CHEN C Y. A user dual clustering recommendation algorithm based on cloud model[J]. Computer Engeering & Science,2015,37(7):1245-125. (in Chinese)
陈平华，陈传瑜. 基于云模型的用户双重聚类推荐算法[J]. 计算机工程与科学，2015，37(7)：1245-1251.

[5] ZHANG G W,LI D Y,LI P,et al. A collaborative filtering recommendation algorithm based on cloud model[J]. Journal of software,2007,18(10):2403-2411. (in Chinese)
张光卫，李德毅，李鹏，等. 基于云模型的协同过滤推荐算法[J]. 软件学报，2007，18(10)：2403-2411.

[6] LIAN D, ZHAO C, XIE X, et al. Joiny geographical modeling and matrix factorization for point-of-interest recommendation [C]// Proc. of the KDD. New York: ACM Press, 2014: 831-840.

[7] YIN H,SUN Y,CUI B,et al. LCARS: A location-content-aware recommender system[C]// Proc. of the KDD. New York: ACM Press,2013:221-229.

[8] SUN G F,WU L,LIU Q,et al. Recommendation based on collaborative filtering by exploiting sequential behaviors[J]. Journal of Software,2013,24(11):2721-2733. (in Chinese)
孙光福，吴乐，刘淇，等. 基于时序行为的协同过滤推荐算法[J]. 软件学报，2013，24(11)：2721-2733.

[9] GUO L,MA J,CHEN Z M,et al. Incorporating item relations for social recommendation[J]. Chinese Journal of Computers, 2014,37(1):219-228. (in Chinese)
郭磊，马军，陈竹敏，等. 一种结合推荐对象间关联关系的社会化推荐算法[J]. 计算机学报，2014，37(1)：219-228.

[10] QUIJANO-SANCHEZ L,RECIO-GARCIA J A,DIAZ-AGUDO B,et al. Social factors in group recommender systems[J]. Acm Transactions on Intelligent Systems & Technology,2013,4(1): 1199-1221.

[11] YU C H,LIU X J,LI B. Implicit feedback personalized recommendation model fusing context-aware and social network process[J]. Computer Science, 2016, 43(6): 248-279. (in Chinese)
俞春花，刘学军，李斌. 隐式反馈场景中融合社交信息的上下文感知推荐[J]. 计算机科学，2016，43(6)：248-279.

(下转第 271 页)

的 Spark 大数据处理框架，根据中文地址的语义信息进行地址要素解析，并提出了中文字符串、数字和字母分离的解析预处理方法，有效降低了匹配过程中的数据规模；其次，根据定义的多种距离信息构建了贝叶斯网络，结合多准则决策模型，给出了在预选集合中的智能匹配算法。最后，基于芜湖市 514967 条个人信息脱敏后的燃气地址与社区网格化地址库中的地址（共计 1770979 条）进行匹配验证，主要从匹配率、准确率以及算法效率进行分析。实验结果显示，相比于传统匹配算法，本文提出的算法在匹配率和精确率上具有更好的均衡性。特别是在处理大规模数据时，相比于传统地址匹配算法，该算法在一定程度提升了性能、效率，对于实际的中文地名地址信息处理具有一定的应用价值。

参 考 文 献

[1] REMERO, BARRIGA, MOLANO. Big Data Meaning in the Architecture of IoT for Smart Cities [C] // International Conference on Data Mining and Big Data. Springer International Publishing, 2016: 457-465.

[2] DELMASTRO F, ARNABOLDI V, CONTI M. People-centric computing and communications in smart cities [J]. IEEE Communications Magazine, 2016, 54(7): 122-128.

[3] LIU D, PEI Y, LI C. Research on Establishment of Grid-based Intelligent Community Synergistic Service Platform [J]. Bulletin of Surveying and Mapping, 2015, 3(12): 98-100.

[4] PU Z, XU L. Research to the Community Resources Integration Under Grid City Management [J]. Asian Social Science, 2009, 4(7): 64-68.

[5] LI D R, CAO J J, YAO Y. Big data in smart cities [J]. Science China Information Sciences, 2015, 58(10): 1-12.

[6] HASHEM I A T, CHANG V, ANUAR N B, et al. The role of big data in smart city [J]. International Journal of Information Management, 2016, 36(5): 748-758.

[7] GOLDBERG D W, WISON J P, KNOBLOCK C A. From text to geographic coordinates: the current state of geocoding [J]. Urisa Journal, 2007, 19(1): 33-46.

[8] DRUMMOND W J. Address Matching: GIS Technology for Mapping Human Activity Patterns [J]. Journal of the American Planning Association, 1995, 61(61): 240-251.

[9] SUN Y, CHEN W. Address Matching Technology Based on Word Segmentation [C] // China Geographic Information System Association Annual Meeting. 2007: 1-12.

[10] MA Z, LI Z, SUN W, et al. An Automatic Geocoding Algorithm Based on Address Segmentation [J]. Bulletin of Surveying and Mapping, 2011, 4(2): 59-62.

[11] TIAN Q, REN F, HU T, et al. Using an Optimized Chinese Address Matching Method to Develop a Geocoding Service: A Case Study of Shenzhen, China [J]. ISPRS International Journal of Geo-Information, 2016, 5(65): 1-17.

[12] WEI J, ZHONG Z. An Approach to Address Matching Based on Confidence [J]. Science of Surveying and Mapping, 2015, 40(1): 122-125.

[13] HUANG K, MA S. Chinese Web Page Classification Based on Statistical Word Segmentation [J]. Journal of Chinese Information Processing, 2002, 16(6): 25-31.

[14] XIAO J. Method of Recognition and Match of Place Name Based on Statistic [J]. Journal of Geomatics Science and Technology, 2014, 31(4): 408-412.

[15] SONG Z. Address matching algorithm based on chinese natural language understanding [J]. Journal of Remote Sensing, 2013, 17(4): 788-801.

[16] MA L, GONG J. Application of Spatial Information Natural Language Query Interface [J]. Geomatics and Information Science of Wuhan University, 2003, 28(3): 301-305.

[17] ZHANG X. A knowledge-based agent prototype for Chinese address geocoding [C] // Geoinformatics 2008 and Joint Conference on GIS and Built environment: Advanced Spatial Data Models and Analyses. 2008: 1-10.

[18] JING Z, QI L. Research on the application of geocoding [J]. Geography and Geo-Information Science, 2003, 3(19): 22-25.

[19] QIN B, WANG Q Y, LI C. Effective Strategy for Sensitive Analysis of Bayesian Networks [J]. Journal of Chinese Systems, 2016, 37(4): 732-737.

[20] GE S, XIA X. An Intelligence Decision Model Based on Probabilistic Influence Analysis [J]. Computer Engineering, 2016, 42(6): 213-217.

[21] PEARL J. Fusion, propagation, and structuring in belief networks [J]. Artificial Intelligence, 1986, 29(3): 241-288.

[22] YAO X, LI X, PENG L. A Novel Fuzzy Chinese Address Matching Engine Based on Full-text Search Technology [C] // Proceedings of Science. 2015: 1-9.

（上接第 255 页）

[12] SUN G M, WANG S. Compute adaptive fast recommendation algorithm statisfied user interest drift [J]. Application Research of Computers, 2015, 32(9): 2669-2673. (in Chinese)
孙光明，王硕. 满足用户兴趣漂移的计算自适应快速推荐算法 [J]. 计算机应用研究，2015，32(9)：2669-2673.

[13] LI C, LIANG C Y, MA L. A Collaborative filtering recommendation algorithm based on Domain nearest neighbor [J]. Journal of Computer Research and Development, 2008, 45(9): 1532-1538. (in Chinese)
李聪，梁昌勇，马丽. 基于领域最近邻的协同过滤推荐算法 [J]. 计算机研究与发展，2008，45(9)：1532-1538.

[14] WANG X M, ZHANG X M. Collaborative recommendation algorithm based on contribution factor [J]. Application Research of Computers, 2015, 32(12): 3551-3554. (in Chinese)
王兴茂，张兴明. 基于贡献因子的协同过滤推荐算法 [J]. 计算机应用研究，2015，32(12)：3551-3554.

[15] LEE H C, LEE S J, CHUNG Y J. A Study on the Improved Collaborative Filtering Algorithm for Recommender System [C] // Acis International Conference on Software Engineering Research, Management & Applications. IEEE Computer Society, 2007: 297-304.

[16] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters [J]. Communications of the ACM, 2008, 51(1): 107-113.

[17] ZHAO Z D, SHANG M S. User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop [C] // International Conference on Knowledge Discovery and Data Mining. IEEE, 2010: 478-481.