

文章编号:1007-5321(2017)01-0074-05

DOI:10.13190/j.jbupt.2017.01.013

利用模糊分块改进协同过滤的扩展性和准确性

王晓军, 付 超

(南京邮电大学 信息网络技术研究所, 南京 210003)

摘要: 项目的协同过滤方法利用项目之间相似性预测用户对项目的评分,但相似项的选择面临可扩展性和准确性的问题. 为此,提出分布式协同过滤方法,利用模糊分块技术将项目集分成若干块,然后仅在各块内比较项目的相似性. 通过裁剪相似关系图进一步改善效率,从图中去除不可能相似的项目之间的边. 最后,利用图的分区技术,将相似关系图分割为若干较小的区,在各分区上并行计算项目的相似度. 实验结果表明,该方法能改善推荐系统的准确性和可扩展性.

关 键 词: 推荐系统; 个性化推荐; 协同过滤; 数据分块; 模糊聚类

中图分类号: TP391

文献标志码: A

Enhancing Scalability and Accuracy of Collaborative Filtering Using Fuzzy Blocking

WANG Xiao-jun, FU Chao

(Institute of Information and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: The ratings of items based on the similarities between items are predicted by traditional item-based collaborative filtering methods. However, the selections of the similar ones are suffering from limited scalability and accuracy. A distributed collaborative filtering method was proposed. This method clusters items into several blocks using fuzzy blocking, and performs comparisons solely among the items within each block. Additional efficiency enhancements can be achieved through the pruning of the similar relationship graph: edges between items that are not likely to be similar can be removed from the graph. It divides this graph into multiple smaller partitions from each which similarity degrees between items is calculated efficiently in parallel. Experiments show that the proposed method can improve the recommendation scalability and accuracy.

Key words: recommender systems; personalized recommendation; collaborative filtering; data blocking; fuzzy clustering

互联网上不断增加的数据量使用户需要花费很多时间才能找到有价值的信息. 协同过滤(CF, collaborative filtering)被认为是解决信息超载的有效技术之一,已广泛应用于推荐领域. 为搜索目标项目或用户的相似项,CF需要对项目或用户进行两两比

较. 但推荐系统拥有数以百万计的用户和项目,并且其规模持续增长,用传统方法在大数据中搜索相似项难以保证在合理时间内提供较准确的结果. 为解决上述问题,笔者提出了一种利用模糊分块技术实现并行化的协同过滤方法(CFFB, collaborative fil-

收稿日期: 2016-09-13

基金项目: 国家自然科学基金项目(61003237)

作者简介: 王晓军(1968—),女,副研究员, E-mail: xjwang@njupt.edu.cn.

tering using fuzzy blocking), 主要贡献包括: 1) 利用模糊分块技术将项目集划分为若干块, 生成具有冗余特性的项目块集合; 2) 利用项目-块的隶属关系构建相似关系图, 通过对相似关系图的裁剪, 剔除多余的比较; 3) 利用图的分区技术, 实现项目相似度和预测推荐的并行计算, 改进 CF 的推荐精度和效率。

1 相关研究

CF 的推荐精度依赖于近邻的选择。在大量的多维数据中搜索前 K 个相似项是非常低效的。加之推荐系统中数据价值密度分布不均衡, 需要探索一种按需约简数据集的方法。

Sobia 等^[1]利用改进的 K -Means 算法将评分矩阵划分为 K 个簇, 然后根据与目标用户相似的簇中心的评分给出推荐。Hu 等^[2]从大数据集中动态地为目标用户构造局部项目-用户矩阵, 并在该局部矩阵上实施预测。为缓解数据稀疏性问题, Hu 等对每个用户簇中的用户评分采用了平滑策略, 在推荐阶段不仅考虑用户对相似项评分, 而且还融合志同道合的用户分别对相似项目和同一项目评分。Xu 等^[3]提出在实施推荐之前先估计缺失项, 将单评分转为共同评分, Son^[4]在用户相似度中集成了模糊相似度和硬相似度。前者根据人口统计数据计算获得, 后者依据用户的历史评分而计算得到。为解决扩展性问题, Xu^[3]和 Xie 等^[5]采用 Map-Reduce 范式对推荐系统中的数据进行并行处理。Apache 的 Mahout 开源项目也采用 Map Reduce 实现 CF。

2 问题定义

为避免在全量数据集上计算项目相似性, CFFB 利用重叠分块方法依据项目的特征将项目集划分为多个块, 获得块集合 B , 然后只在块集合 B 中的每个分块内进行项目的两两比较和相似度计算。在一般情况下, 块内不必要的比较分为 2 类: 1) 冗余比较。在各块中重复比较相同的项目对, 这增加了比较次数, 导致低效率; 2) 多余比较。多余比较指在不可能相似的项目之间进行比较。CFFB 需要丢弃冗余和多余的比较, 保证推荐的效率和精度。

分块技术有 2 类相互竞争的度量: 效率和效益。效率直接关系到块内的比较次数, 即每块中需比较的项目对总数, 由式(1)定义, 其中 $|b|$ 为块 $b \in B$ 包含的项目数, $\|b\|$ 为在块 b 内需比较的项目对数。

$$\|B\| = \sum_{b \in B} \|b\| = \sum_{b \in B} \frac{|b|(|b| - 1)}{2} \quad (1)$$

效率也可用缩减比 (RR, reduction ratio) 度量, 记为 R , 用于测量效率相比于基准块集 B_{bs} 增强的程度, 由式(2)定义, R 取值范围为 $[0, 1]$, 其数值越高, 表示块内比较的次数越少, 分块的效率越高。

$$R(B) = 1 - \frac{\|B\|}{\|B_{bs}\|} \quad (2)$$

基于分块技术的推荐算法的效益可用推荐算法的精度度量。这类指标依赖于相似项选择的正确性。项目对比较的次数越多, 搜索到正确相似项的可能性越大, 但它的效率也越低。因此, 块集合的设置存在效益和效率之间的权衡。为识别项目间的冗余和多余的比较, CFFB 引入了相似关系图 G_B 结构。

定义 1 共现对。已知推荐系统中项目集 $I = \{i_k | k = 1, \dots, n\}$ 、项目块集合 B 。若项目 $i \in I$ 和 $j \in I$ 同时出现在块集合 B 的同一个块 $b \in B$ 中, 则称项目 i 和 j 为共现对, 记为 $\langle i, j \rangle$ 。

定义 2 相似关系图 G_B 。已知项目集合 I 和块集合 B , 从它衍生出来的无向图 $G_B = \{V_B, E_B\}$, V_B 为顶点集合, 每个顶点对应项目集 I 中 1 个项目, 即 $\forall v_i \in V_B: \exists i \in I \wedge b \in B \wedge i \in b$ 。 E_B 为无向边集合, 每条边对应 1 个共现对, 即 $\forall e_{i,j} = \langle i, j \rangle \in E_B: i \neq j \wedge \exists b_k \in B \wedge i \in b_k \wedge j \in b_k$ 。

在 G_B 中, 边 $e_{i,j} \in E_B$ 邻接的 2 个顶点 v_i 和 v_j 对应的项目 i 和 j 称为边 $e_{i,j}$ 的共现项目。边 $e_{i,j}$ 表示项目间存在可能的相似关系, 需在后续步骤中通过相似度计算, 确定项目 i 和 j 是否为相似对。

3 利用模糊分块技术改进 CF

CFFB 包括 2 个阶段: 相似度计算阶段和预测推荐阶段, 其中相似度计算阶段其处理流程如下:

- 1) 采用笔者^[6]提出的方法构建项目偏好向量;
- 2) 利用项目偏好和基本特征对项目进行模糊分块, 产生块集合 B ;
- 3) 剪裁块集合 B 对应的相似关系图 G_B ;
- 4) 并行计算项目相似度。

3.1 模糊分块

由于 K -Means 聚类算法只允许每个项目指派给 1 个簇, 并依赖于初始簇中心的选择; 模糊 K -Means 可避免出现局部最优。因此, CFFB 利用模

糊 K -Means 算法实现重叠分块,将具有相似项目偏好和基本特征的项目组合在一起,构成 K 个项目簇,并将项目 $i \in I$ 按一定概率划分到多个簇中. 每个项目簇对应 1 个块,由此产生含 $K(K \geq 1)$ 个块的集合 B .

3.2 构建和裁剪相似关系图

对于块集合 B 中每个共现对 $\langle i, j \rangle$,若顶点 v_i 和 v_j 之间不存在 1 条边,则将项目 i 和 j 对应的顶点 v_i 和 v_j 加入到相似关系图 G_B 中,并用 1 条边 $e_{i,j}$ 连接顶点 v_i 和 v_j ;否则意味着共现对 $\langle i, j \rangle$ 为冗余对,需剔除此冗余的比较. 因此,每对共现对至多对应 1 条边. 为进一步剔除多余的比较,需量化图 G_B 中每条边的权重. 边的权重表示了边的共现项目成为相似对的可能性. 在理想状态下,项目对越相似,权重值应越大.

已知 $B_i \subseteq B, B_j \subseteq B$ 分别为项目 i 和 j 隶属的块集合, $B_{i,j} = B_i \cap B_j$ 为项目 i 和 j 共享的集合. 项目 i 和 j 共享的块越多(即 $|B_{i,j}|$ 值越大),它们越有可能是相似对. 除了考虑 $|B_{i,j}|$,边的权重还依赖于其共现项目所隶属的块的总数以及边的邻接顶点的度. 当某个边共现项目隶属的块数越少,该边的权重应越高;当某个边顶点的度越小,该边的权重应越高. 因此,边 $e_{i,j}$ 的权重 $w_{i,j}$ 依赖于 B_i 和 B_j 集合的 Jaccard 相似度,有

$$w_{i,j} = \frac{|B_{i,j}|}{|B_i| + |B_j| - |B_{i,j}|} \log \frac{|E_B|}{|v_i|} \log \frac{|E_B|}{|v_j|} \quad (3)$$

其中 $|v_i|$ 为顶点 v_i 的度. 上述加权方案依赖于模糊分块的分块原则:在实施分块时,将相似概率高的项目指派到同一个块中. 因此,当 2 个项目分享的块数越多,它们越可能相似. CFFB 可依据最小权重阈值 w_{\min} 删除图 G_B 中权重低于 w_{\min} 的边,以剔除多余的比较.

3.3 并行计算项目相似度

为有效地并行处理拥有大量顶点或边的图,需将图分割成若干更小的区. 由于相似关系图中每条边表示一次项目对的比较,所以适宜采用顶点分割法,将边 $e_{i,j} \in E_B$ 唯一指派给某个图的一个分区,从而使得相似度计算任务可以划分为多个并行执行的子任务,每个子任务处理一个分区.

E_B^p 为分区 p 中包含的边. 在分区 p 中,利用式(4)计算边 $e_{i,j} \in E_B^p$ 的共现项目 i 和 j 之间相似度为 $S(i,j) = \omega S_R(i,j)(1 - \alpha_p(i,j)) + (1 - \omega) S_F(i,j)$

其中 ω 为项目评分相似度 $S_R(i,j)$ 的权值, S_F 为项目特征相似度,由式(5)中的欧氏距离定义,其中 $F_i = (f_{i,1} \cdots f_{i,q})$ 为项目 i 的特征向量.

$$S_F(i,j) = \frac{1}{1 + \sqrt{\sum_{l=1}^q (f_{i,l} - f_{j,l})^2}} \quad (5)$$

用户对项目的公共评分数对评分相似度有较大影响,项目的公共评分数越多,其越有助于产生更好的推荐. 为此,式(4)利用式(6)的项目公共评分稀疏度 $\alpha_p(i,j)$ 调整评分相似度. 公共评分越多,其相似度越大.

$$\alpha_p(i,j) = 1 - \frac{n_{i,j}}{\max_{e_{i,j} \in E_B^p} (n_{i,j})} \quad (6)$$

其中 $n_{i,j}$ 为同时对项目 i, j 评分的用户数. 在每个分区中完成相似度的计算后,需依据相似项搜索方法输出每个项目的相似项,并汇总来自各分区的计算结果实施推荐.

由于 CFFB 只对相似关系图中的边实施比较,项目相似度计算总次数为 $|E_B|$,所以对于项目总数为 $|I|$ 的推荐系统,其缩减比 $R(B) = 1 - |E_B| / \|B_{bs}\|$,其中基准块集中比较次数 $|B_{bs}| = |I|(|I| - 1)/2$. 只要在模糊分块过程中,每一块 $b \in B$ 满足 $|b| < |I|$ 条件,则依据构图原则, $|E_B| < |B_{bs}|$ 成立,所以 $0 < R(B) < 1$.

4 算法评估

CFFB 采用 Apache Mahout 应用程序编程接口实现模糊聚类,采用 Spark 实现相似度计算和推荐. 为比较各算法的推荐质量和性能,在相同的开发环境中分别实现了新的质心选择(NCS, novel centroid selection)^[1]、采用平滑融合的协同过滤(CFSF, collaborative filtering using smoothing and fusion)^[2]和基于混合用户的模糊协同过滤(HU-FCF, hybrid user-based fuzzy collaborative filtering)^[4]算法. CFFB、HU-FCF 和 Mahout 基于项目的协同过滤(ICF, item-based collaborative filtering)均使用 Pearson 相关系数测量项目之间的评分相似度. 搭建的 Hadoop 平台拥有 12 台物理计算机,每台计算机的 Java 虚拟机运行在 VMWare 中的 32 位、内存为 2 G 的 ubuntu 操作系统上. 采用的 MovieLens 数据集和 Yahoo! Movies 数据集信息如表 1 所示,且分别按 8:2 比例划分为训练集和测试集.

表 1 MovieLens 数据集统计属性

数据集类型	用户数	项目数	总的评分数
Yahoo!	1 699	1 167	79 088
ML-100 K	943	1 682	100 000
ML-1 M	6 040	3 952	1 000 209
ML-10 M	71 567	10 681	10 000 054

5.1 对精度的影响

采用召回率 F_R 、准确率 F_P 和综合评价指标 F_1 度量评估算法的精度. 通常情况下, F_R 和 F_P 并不是孤立讨论的, 常采用 F_1 值度量, 对于块集 B 的 F_1 计算如下.

$$F_1(B) = \frac{2F_R(B)F_P(B)}{F_R(B) + F_P(B)} \tag{7}$$

5.1.1 w_{\min} 对缩减比和推荐质量的影响

为进一步消除多余的比较, 通过设置恰当的权重阈值 $w_{\min} > 0$, 剔除图中低于权重的边, 提高缩减比. 表 2 显示, 若 w_{\min} 值过高, 尽管缩减比提高了, 但由于错过有价值的近邻, 所以影响推荐精度.

表 2 w_{\min} 的影响

数据集类型	w_{\min}	$R/\%$	F_1
Yahoo! ($K=2$)	0	9.16	0.317 57
	21.0	10.62	0.317 68
	21.5	13.71	0.317 79
	22.0	17.68	0.317 45
ML-1 M ($K=4$)	0	5.74	0.420 41
	12.0	7.44	0.420 54
	13.0	15.33	0.420 58
	14.0	22.54	0.420 25

5.1.2 稀疏度对精度的影响

为了观察算法在不同稀疏度上的特性, 将 ML-100 K 数据集划分为不同稀疏度的训练集和测试集. 图 1 和图 2 显示 CFFB 的召回率和准确率均优于另外 4 种算法. 特别是当数据集稀疏时, 改善更为显著.

5.2 扩展性分析

各算法在相似度计算阶段的运行时间如图 3 所示. CFSF 采取了平滑策略, 其运行时间较长, 因此, 实验只测试 CFSF 在 ML-1 M 和 ML-100 K 下的运行时间. 尽管 Mahout CF 采用了 Map-Reduce 范式, 但由于只在随机选取的 500 个用户评分数据集上计算项目相似度矩阵, 所以其相似度计算阶段的运行时

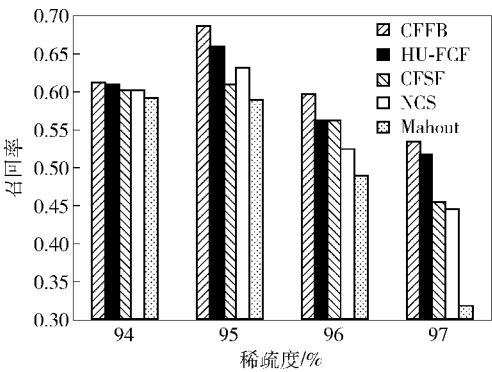


图 1 对召回率的影响

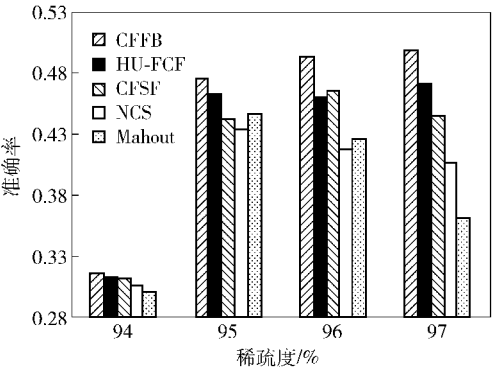


图 2 对准确率的影响

间在各个算法中是最快的, 但从图 1、图 2 可见, 这也影响了其推荐精度. NCS 在小数据集上计算相似度的时间与 Mahout CF 接近, 但随着数据集的扩大, 其相似度计算时间增长幅度高于 CFFB.

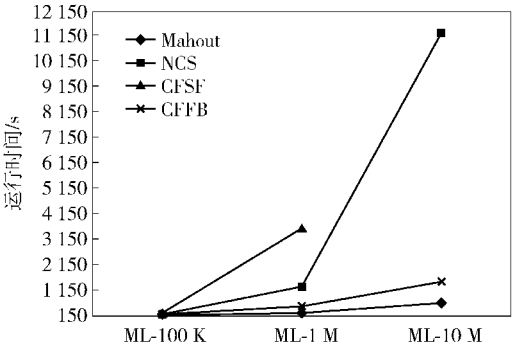


图 3 相似度计算阶段运行时间

6 结束语

提出了一种基于模糊分块的协同过滤方法. 为缓解数据集的稀疏性, 该方法在相似度计算过程中融合了项目特性信息. 为解决扩展性问题, 利用模糊聚类算法将项目集分成若干块, 然后利用项目和块的隶属关系构建相似关系图, 通过裁剪最低权重

的边,剔除多余比较. 最后在 Hadoop 和 Spark 环境中验证了方法的有效性. 但该方法依赖于块集合的冗余程度,图裁剪也需通过调整相应参数得到,因此,还需要探寻一种调参少、依赖于块集合的冗余程度低的分块方法.

参考文献:

- [1] Sobia Z, Mustansar A G, Asra K, et al. Novel centroid selection approaches for K -means-clustering based recommender systems[J]. Information Sciences, 2015, 320(11): 156-189.
- [2] Hu Long, Lin Kai, Hassan M M, et al. CFSF: on cloud-based recommendation for large-scale e-commerce[J]. Mobile Networks and Applications, 2015, 20(3): 380-390.
- [3] Xu Ruzhi, Wang Shuaiqian, Zheng Xuwei, et al. Distributed collaborative filtering with singular ratings for large scale recommendation[J]. The Journal of Systems and Software, 2014, 95(9): 231-241.
- [4] Son L H. HU-FCF: a hybrid user-based fuzzy collaborative filtering method in recommender systems[J]. Expert Systems with Applications, 2014, 41(15): 6861-6870.
- [5] Xie Feng, Chen Zhen, Xu Hongfeng, et al. TST: threshold based similarity transitivity method in collaborative filtering with cloud computing[J]. Tsinghua Science and Technology, 2013, 18(3): 318-327.
- [6] 王晓军. 推荐系统中分布式混合协同过滤方法[J]. 北京邮电大学学报, 2016, 39(2): 25-29.
Wang Xiaojun. A distributed hybrid collaborative filtering method in recommender systems[J]. Journal of Beijing University of Posts and Telecommunications, 2016, 39(2): 25-29.