基于启发式聚类模型和类别相似度的协同过滤推荐算法

王兴茂,张兴明,吴毅涛,潘俊池
(国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 基于 k-近邻的协同过滤推荐算法对于邻居数量 k 的确定过于主观,并且推荐时以 k-近邻均值加权推荐不够准确.针对这两个问题,本文首先引入并改进最大最小距离聚类算法,进而设计启发式聚类模型将用户进行不规定类别数的自由聚类划分,目标用户所在类的用户为邻居用户,客观确定邻居数量;然后在推荐时定义类别相似度,针对性地建立目标用户未评分和评分项目的潜在类别关系,改进 k-近邻均值加权算法.实验结果表明,该算法提高了推荐准确度(约0.035MAE).

关键词: 协同过滤;推荐算法;聚类算法;启发式聚类模型;类别相似度
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2016)07-1708-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.07.027

A Collaborative Recommendation Algorithm Based on Heuristic Clustering Model and Category Similarity

WANG Xing-mao, ZHANG Xing-ming, WU Yi-tao, PAN Jun-chi
(*National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, Henan* 450002, *China*)

Abstract: The collaborative recommendation algorithm based on kNN confirms the number of neighbours subjectively, and is not accurate enough to predict by kNN mean weighting calculating. To address these two problems, the maximum and minimum distance clustering algorithm was introduced and improved to design the heuristic clustering model, the model divided the users allodially without the determination of the category numbers, the neighbours of the target users were the users who were in the same category with the target users; then the category similarity was defined to build the category relation between the unscore and score items of the target user in prediction, and the kNN mean weighting calculating was advanced based on the category similarity. The experiments show that this algorithm improves the degree of accuracy (reducing about 0.035 MAE).

Key words: collaborative; recommendation algorithm; clustering algorithm; heuristic clustering model; category similarity

1 引言

协同过滤是推荐系统中应用最广泛、最成功的推荐算法[1].它能根据用户的历史行为来预测用户的偏好,进而为用户进行个性化的推荐,已成为学术界研究的热点[2~4].但随着互联网的爆炸式扩张,数据稀疏性成为推荐系统最突出的问题[5],导致目标用户选择出的邻居不合理,进而导致推荐结果准确度降低.很多学者采用聚类算法来提高推荐的效果[6],G Adomavicius 和 Tuzhilin 根据用户评分的相似性对用户进行聚类,离线时处理数据,在线时寻找最近邻居[7].Truong 等对项目进行 k-均值聚类,计算了目标项目与聚类中心的相似性,然后在项目聚类中寻找目标项目最近邻居并产生推荐列表[8].张莉等综合用户和项目特征,采用 k-均值对相似用户聚类,然后结合用户兴趣类别活跃度进行推荐[9].Rashid 等在用户聚类基础上生成每个聚类的代理用户,然后基于目标用户的最相似代理用户进行最近邻协同过滤推荐[10].George 等提出对用户和项目同时进行聚类的协同过滤方法[11].以上的研究缓解了数据稀疏度对推荐系统的影响,但存在两个基本问题,一是算法主观地选择了聚

收稿日期:2014-12-12;修回日期:2015-10-15;责任编辑:李勇锋
基金项目:国家973重点基础研究发展计划(No.2012CB315901);国家863高技术研究发展计划(No.2011AA01A103)

类时的类别数量和邻居的数量;二是在推荐时采用 k-近邻均值加权的算法以评分项目均值加权的方式太过一般性,没有考虑被预测的项目与目标用户所评价项目的潜在关系,即本文引入的类别相似度.

针对这两个问题,本文首先以用户之间欧氏距离为标准,在推荐系统中引入并改进最大最小距离聚类算法进而设计启发式聚类模型对用户进行客观的类别划分,以目标用户所在类的集合为邻居集合,不主观规定类别的数量和邻居的数量 k;然后在预测评分时引入项目之间的类别相似度,对传统的 k-近邻均值加权公式进行根本的改进,使预测评分更加合理.

2　基于启发式聚类模型和类别相似度的协同过滤推荐算法

2.1　启发式聚类模型的提出

最大最小距离聚类算法[12]的最大优势就是复杂度较低,但它存在第一个聚类中心随机选取带来的聚类不准确而且参数 α 的选取主观问题. 本文以最大最小聚类算法为基础设计启发式聚类模型来客观地对用户的类别进行划分,如图 1 所示,该模型采用启发式的思想来确定第一个聚类中心. 为了更好地对模型进行说明以及方便后文的计算,先介绍相关定义:

定义 1　用户 U_i 点密度:以用户 i 为中心,d 为半径,欧式距离为衡量标准,这个球状簇内所有用户的数量为用户 U_i 的点密度,设为 m_i.

定义 2　欧式距离矩阵:存储各个用户之间欧氏距离的矩阵,设为 $\boldsymbol{D}(m\times m)$.

该模型核心思想分为三步:

step1　通过用户之间的欧式距离计算建立欧式距离矩阵;

step2　采用启发式的思想来确定第一个聚类中心,即一个用户周围的用户数越多,就越适合作为聚类中心,本模型通过点密度的计算(见 2.2.2 小节)挑选出适合作第一个聚类中心的用户;

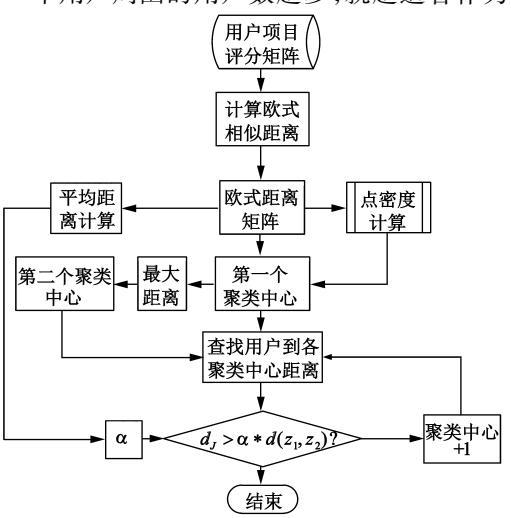

图1　启发式聚类模型

step3　计算剩余用户到各聚类中心的距离,假设已有 s 个聚类中心 $z_1,z_2,z_3,\cdots,z_s$,计算尚未作为聚类中心的第 i 个样本 x_i 到 s 类中心 $z_j(j=1,2,3,\cdots,s)$ 的距离 d_{ij},并计算 $d_j=\max\limits_i\{\min(d_{i,1},d_{i,2},d_{i,3},\cdots,d_{i,s}),i=1,2,3,\cdots,m-s\}$,如果 $d_J>\alpha*d(z_1,z_2)$,则建立第 $s+1$ 个聚类中心 z_{s+1},且 $z_{s+1}=x_J$,否则聚类结束. $d(z_1,z_2)$ 为第一个和第二个聚类中心的距离,α 通过计算获得.

2.2　算法的相关计算

2.2.1　欧式相似度

设用户的评分向量为 $(r_{i1},r_{i2},r_{i3},\cdots,r_{im})$,则两个用户 U_i 和 U_j 之间的欧式距离[13]如式(1):

$$d_{i,j}=\sqrt{\sum_{k=1}^{m}(r_{ik}-r_{jk})} \tag{1}$$

欧式相似度与欧式距离成反相关的关系,U_i 和 U_j 的欧式相似度如式(2):

$$Osim=\frac{1}{d_{i,j}+1} \tag{2}$$

设用户项目评分矩阵为 $\boldsymbol{R}(m\times n)$ 如式(3),这是 m 个用户对 n 个项目的评分矩阵,r_{ij} 为用户 i 对项目 j 的评分.

$$\boldsymbol{R}=\begin{pmatrix}r_{11}&\cdots&r_{1n}\\ \vdots&\ddots&\vdots\\ r_{m1}&\cdots&r_{mn}\end{pmatrix} \tag{3}$$

通过矩阵 $\boldsymbol{R}(m\times n)$ 和式(1),计算系统中任意两个用户 u_i 与 u_j 之间的欧式距离 $d_{i,j}$. 构成系统中用户的欧式距离矩阵 $\boldsymbol{D}(m\times m)$ 如式(4):

$$\boldsymbol{D}=\begin{pmatrix}d_{1,1}&\cdots&d_{1,m}\\ \vdots&\ddots&\vdots\\ d_{m,1}&\cdots&d_{m,m}\end{pmatrix} \tag{4}$$

进而利用矩阵 $\boldsymbol{D}(m\times m)$ 和式(2)计算这两个用户之间的欧式相似度 sim_{ij},构成欧式相似度矩阵 $\boldsymbol{OS}(m\times m)$ 如式(5):

$$\boldsymbol{OS}(m\times m)=\begin{pmatrix}Osim_{11}&\cdots&Osim_{1m}\\ \vdots&\ddots&\vdots\\ Osim_{m1}&\cdots&Osim_{mm}\end{pmatrix} \tag{5}$$

2.2.2　点密度和参数 α 的计算

模型中采用启发式的思想通过点密度的计算来确定第一个聚类中心,计算如下:

首先需要找合适的 d,本文令它为系统中所有用户之间距离之和的均值. 通过 2.2.1 小节的计算,知道 $\boldsymbol{D}(m\times m)$ 是一个角对称矩阵,只需要计算上三角,即 d 的计算如式(6):

$$d = \frac{\sum_{i=1}^{m}\sum_{j=i}^{m} d_{i,j}}{\sum_{i=1}^{m} i} \tag{6}$$

点密度计算：

设 $\boldsymbol{L}_i(d_{i,1},d_{i,2},d_{i,3},\cdots,d_{i,m})$ 为矩阵 $\boldsymbol{D}(m\times m)$ 的第 i 行的行向量，它为用户 U_i 与系统中其它用户的距离向量. 引入参数 $m_{i,j}$，对任意 $j\in\{1,2,3,\cdots,m\}$，如果 $d_{i,j}\geqslant d$，则 $m_{i,j}=1$，否则 $m_{i,j}=0$，则用户 U_i 的点密度 m_i 如式(7)：

$$m_i = \sum_{j=1}^{m} m_{i,j} \tag{7}$$

从 $m_i(1\leqslant i\leqslant m$，且 $i\in N)$ 中找出 m_k，满足对于任意的整数 $i\in[1,m]$，有 $m_k\geqslant m_i$. 这样就找出具有最大用户点密度的用户 U_k，我们把用户 U_k 设为第一个中心点 z_1.

α 的计算：

α 的确定关系着聚类的效果，它应该与数据集中用户点之间距离大小有关，本文给出通用计算如下：遍历欧式距离矩阵 $\boldsymbol{D}(m\times m)$，找出 $d_{\max}$，即对于任意 $d_{i,j}(1\leqslant i,j\leqslant m)$ 有 $d_{\max}\geqslant d_{i,j}$. α 的计算如式(8)，关于 d 的计算见式(6).

$$\alpha = d/d_{\max} \tag{8}$$

2.2.3 项目类别相似度计算

定义 3 项目类别相似度 两个项目 I_i 和 I_j 的类别接近程度，设为 $Isim(i,j)$.

我们将系统中项目的用户评分向量 $(r_1,r_2,r_3,\cdots,r_m)$ 进行二元化，即如果 $r_i>0$，则 $r_i=1$，否则 $r_i=0$. 然后引入 Tanimoto 系数计算两个项目 I_x 和 I_y 之间的类别相似度，如式(9)：

$$Isim(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\boldsymbol{x}\cdot\boldsymbol{x}+\boldsymbol{y}\cdot\boldsymbol{y}-\boldsymbol{x}\cdot\boldsymbol{y}} \tag{9}$$

$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 为项目 I_x 和 I_y 经过二元化的评分向量，$\boldsymbol{x}\cdot\boldsymbol{y}$ 是两个项目共同被评价的用户数，$\boldsymbol{x}\cdot\boldsymbol{x}+\boldsymbol{y}\cdot\boldsymbol{y}-\boldsymbol{x}\cdot\boldsymbol{y}$ 为两个项目所有属性的个数.

2.2.4 引入类别相似度的预测评分计算

传统的预测目标用户对未知项目评分公式[14] 如式(10)，式中采用 k-近邻平均加权的方式进行目标用户对未知项目分数的预测.

$$P_{a,i} = \bar{R}_a + \frac{\sum_{j=1}^{k}(R_j(i)-\bar{R}_j)*Sim(a,j)}{\sum_{j=1}^{n} Sim(a,j)} \tag{10}$$

其中 $Sim(a,j)$ 表示目标用户 a 和 j 之间相似性，k 是用户 a 的近邻用户组中用户的个数 $R_j(i)$ 表示用户 j 对项目 i 的评分，$\bar{R}_j$ 表示用户 j 对项目评分的均值，$P_{a,i}$ 表示用户 a 对项目 i 的预测评分.

本文把该公式分两部分组成，我们将其分别定义：

(1)目标用户对目标项目的基本评分值 R_b 如式(11)：

$$R_b = \bar{R}_a \tag{11}$$

(2)其它用户对目标用户的推荐贡献值 R_c 如式(12)：

$$R_c = \frac{\sum_{j=1}^{k}(R_j(i)-\bar{R}_j)*Sim(a,j)}{\sum_{j=1}^{n} Sim(a,j)} \tag{12}$$

显而易见，对于任意的未知的项目，目标用户的基本评分值 R_b 都直接为目标用户已评分项目的平均值 $\bar{R}_a$，这种方式不具有针对性，没有体现未知项目的类别属性对目标用户预评分的具体影响. 本小节引入项目类别相似度对这一部分进行改进. 设目标用户 u_a 评价过的项目为 $i_1,i_2,i_3,\cdots,i_a$，基本评分式如式(13)：

$$R_b^* = \frac{\sum_{j=1}^{a} R_a(j)\times Isim(i,j)}{2\times\sum_{j=1}^{k} Isim(i,j)} \tag{13}$$

式(13)中 $Isim(i,j)$ 为用户评价过的项目与目标项目之间的类别相似度，a 为目标用户 u_a 评价过的项目数，$R_a(j)$ 为目标用户 u_a 对项目 $i_j(1\leqslant j\leqslant a)$ 的评分. 该式通过 $Isim(i,j)$ 与 $R_a(j)$ 加权来计算目标用户对目标项目的评分，然后引入权重 1/2 来计算目标用户对目标项目的均值. 新的预测评分公式如式(14)：

$$\begin{aligned} P_{a,i} &= R_b^* + R_c^* \\ &= \frac{\sum_{j=1}^{k} R_j\times Isim(i,j)}{2\times\sum_{j=1}^{k} Isim(i,j)} + \frac{\sum_{j=1}^{n}(R_j(i)-\bar{R}_j)\times Osim(a,j)}{\sum_{j=1}^{n} Osim(a,j)} \end{aligned} \tag{14}$$

算法流程的设计

基于 2.2 小节的主要步骤的相关计算，本节直接给出本文算法的流程.

输入：用户项目评分矩阵 $\boldsymbol{R}(m\times n)$，目标用户 U_a

输出：为用户 U_a 提供的推荐列表

step1 根据式(1)、(2)和用户评分矩阵 $\boldsymbol{R}(m\times n)$ 计算用户之间的欧几里得距离矩阵 $\boldsymbol{D}(m\times m)$ 和欧式相似度矩阵 $\boldsymbol{OS}(m\times m)$ 分别如式(4)和(5)；

step2 利用式(6)、(7)和矩阵 $\boldsymbol{D}(m\times m)$，计算系统中用户的样本点密度 m_i，并挑选出具有最大点密度的用户 U_k；

step3 将 U_k 设为第一个聚类中心点 z_1，按照 2.1 小节的模型对系统中用户按照欧几里得距离进行不规定类别数 k 的聚类，聚类完成，分别 $G_1,G_2,G_3,\cdots G_k$；

step4 如果 $U_a\in G_g$，则类 G_g 为 U_a 的邻居集；

step5 根据式(13)和 G_g 为目标用户预测未评分的项目 I_i 的评分 $P_{a,i}$，$(1\leqslant i\leqslant n)$；

step6　根据评分 $P_{a,i}$ 大小排序,为目标用户 U_a 提供推荐列表.

3　算法的性能分析

准确度分析前文已经说明,这里不再赘述,所以本节主要对时间复杂度进行分析和比较. 本文的时间开销主要是欧式相似度矩阵计算、第一个样本点的确定、聚类划分、预测评分计算.

(1)欧式相似度矩阵

$sim=\frac{1}{d_{i,j}+1}$:用户之间欧式相似度计算,存储,构成欧式相似度矩阵 $\boldsymbol{OR}(m\times m)$　//执行 $m(m+1)/2$ 次

(2)第一个样本点的确定

m_i:用户点密度计算　//执行 $m(m+1)$ 次

(3)聚类划分

z_2:第 2 个样本点的确定　//执行 $(m-1)$ 次

z_3:第 3 个样本点的确定　//执行 $2(m-2)$ 次

z_4:第 4 个样本点的确定　//执行 $3(m-3)$ 次

…

z_k:第 k 个样本点的确定　//执行 $(k-1)(m-k+1)$ 次

(4)预测评分计算　//执行小于 $m/2$ 次(由于邻居数不定)

算法执行总次数:

$$f(n)=\frac{3}{2}m^2+\frac{k^2-k+3}{2}m-1-2^2-3^2-\cdots-(k-1)^2$$

根据复杂度计算规则计算得出时间复杂度为:

$$T(n)=O(m^2)$$

本文算法复杂度为 $O(m^2)$,经典的最基本协同过滤推荐算法的复杂度也为 $O(m^2)$,说明本文算法在提高了准确度的同时并不会付出相当大的额外时间开销,即本文算法是可行的.

4　仿真实验

仿真实验首先在 MovieLens-100k、Netflix_3m1k 和 Netflix_5m3k 三种数据集下验证本文算法 CluC-CF 的推荐性能,然后人为对数据集 MovieLens-100k 进行处理,进而验证稀疏度和用户数对本文算法 CluC-CF 性能的影响,仿真实验比较以下三种推荐算法:

(1)传统基于用户的协同过滤推荐算法(相似度计算采用 pearson 相关系数)(CF);

(2)基于传统聚类的协同过滤推荐算法(Ck-CF);

(3)基于聚类和类别相似度的协同过滤推荐算法(CluC-CF).

4.1　数据集

本实验是在基于 java 的 Eclipse 开发环境下进行的. 为了验证本文算法的有效性,实验中采用 Grouplens 提供的 MovieLens-100k 和电影租赁网站 Netflix 提供的 Netflix_3m1k 和 Netflix_5m3k 数据集,三种数据集的数据情况如表 1 所示,随机 2-8 分割,80% 为训练数据,20% 为测试数据,进行本文算法的仿真实验.

表 1　MovieLens-100k、Netflix_3m1k 和 Netflix_5m3k 三种数据集数据表

名称	用户数	项目数	评分总数	稀疏度
MovieLens_100k	943	1682	90409	94.3%
Netflix_3m1k	4427	1000	56136	98.7%
Netflix_5m3k	8662	3000	293299	98.9%

用 MovieLens_100k 来进行评分密度及稀疏度计算说明,评分密度 P 如式(15):

$$P=\frac{90409}{943*1682}\approx 0.057 \tag{15}$$

稀疏度 X 如式(16):

$$X=1-P=0.943 \tag{16}$$

4.2　评价指标

本文平均绝对偏差 MAE(Mean Absolute Error)来衡量算法的准确度[15],MAE 可以直观地对推荐质量进行度量,是最常用的一种推荐质量度量方法,时间指标主要考虑训练时间的长短.

(1)准确度指标 MAE:

平均绝对偏差 MAE 通过计算预测的用户评分与实际的用户评分之间的偏差度量预测的准确性,MAE 越小,推荐质量越高. 设预测的用户评分集合表示为 $\{p_1,p_2,p_3,\cdots,p_N\}$,对应的实际用户评分集合为 $\{r_1,r_2,r_3,\cdots,r_N\}$,则平均绝对偏差 MAE 如式(17):

$$MAE=\frac{1}{N}\sum_{i=1}^{N}|p_i-r_i| \tag{17}$$

(2)时间指标:

主要考虑算法的训练时间,因为推荐过程中训练占主导地位.

4.3　实验结果和分析

4.3.1　近邻数对算法精度的影响

当近邻数 k 为 5、30、50、70 时,在 MovieLens-100k、Netflix_3m1k 和 Netflix_5m3k 三个不同数据集下比较 Per-CF、Ck-CF 和 CluC-CF 三种算法的 MAE 大小. 实验结果如图 2 所示.

本文算法通过数据集的特性分析,自动选择邻居数,并不需要人为主观的进行选择,所以推荐准确度与邻居数无关,而 Per-CF 和 CK-CF 都会受邻居数的影响,Per-CF 受近邻数影响较大. 在 Netflix_3m1k 和 Netflix_5m3k 这种高稀疏度的数据集下本文算法 CluC-CF 的性能优势更加明显,因为 CluC-CF 能够根据数据集特性客观调整聚类数量和邻居数量,并且通过类别相似度更能较深地挖掘潜在的关联因素,在数据更稀疏情况下

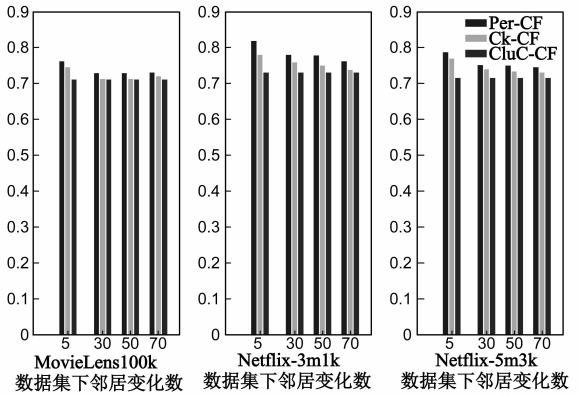

图2 MovieLens-100k、Netflix_3m1k和Netflix_5m3k数据集下三种算法精确度比较

较其它两种算法效果更佳. 而基于共同评价项目评分的 Per-CF 推荐性能恶化的很快,Ck-CF 通过主观聚类分析能一定程度缓解推荐性能的恶化,同时可以看到 CluC-CF 在数据集 Netflix_5m3k 下的推荐准确度要比在 Netflix_3m1k 下高,这因为 Netflix_5m3k 数据集的用户数更多,即用户更加稠密,每一类都不乏相似性高的邻居,根据类内邻居进行推荐会使推荐性能更优. 下面针对稀疏度和用户数对本文算法的影响进一步验证.

4.3.2 稀疏度对算法精度的影响

为了进一步验证稀疏度对本文算法 CluC-CF 推荐精度的影响,本小节在 MovieLens-100k 数据集中,保证用户数和项目数不变,随机减少评分矩阵的评分,增加稀疏度,当邻居数 $k=40$(此时 Per-CF 和 CK-CF 性能最优),比较三种算法的精度,实验结果如图 3.

当稀疏度变大时,相似性计算精度会变低,会导致主观规定邻居数进而硬性选择邻居更加不准确,而本文算法 CK-CF 会根据数据集特性自动选取邻居,并引入项目类别相似度提高推荐精度,所以表现会优于 Per-CF 和 CK-CF,也就是更适用于稀疏度高的数据集.

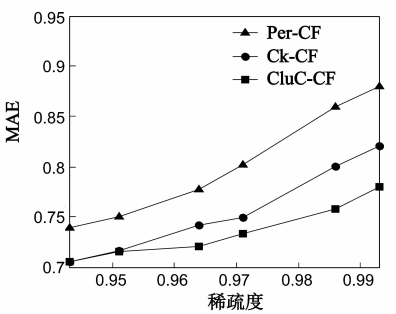

图3 稀疏度对三种算法精确度比较

4.3.3 用户数对算法精度的影响

本小节将验证推荐系统中的用户数对本文算法的影响,本实验在数据集 MovieLens-100k 中,$k=40$,保证项目和稀疏度不变,将用户数逐渐增加,比较三种算法的精度,实验结果如图 4.

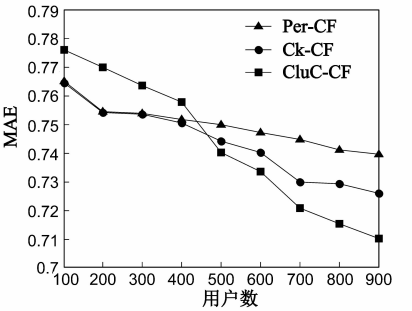

图4 用户数对三种算法精确度比较

随着邻居数的继续增加,本文算法的性能逐渐变优. 分析这一原因是,当用户数很少时,CluC-CF 聚类产生的每一类中用户数变少,即导致邻居缺乏,甚至可能出现“孤苦伶仃无邻居”的现象,所以准确度低的可怜. 而当用户数很多时,每一类都不乏相似性高的邻居,会使推荐性能更优. 而实际推荐系统中往往用户数都至少是数以万计,所以本文算法的实用性更强.

4.3.4 算法运行时间

为佐证 3.2 小节分析的 CluC-CF 时间复杂度,本实验来比较三种算法的运行时间,实验结果如图 5 所示.

可以看出 CluC-CF 的运行时间接近传统的推荐算法 Per-CF,而小于 CK-CF. 这是因为 CK-CF 采用 k-均值聚类后又要重新选择邻居,而本文算法不需要,所以时间要小于 CK-CF. 然而 CluC-CF 要通过聚类来客观选择邻居,所以时间大于 Per-CF.

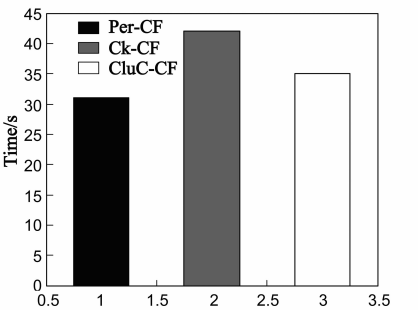

图5 MovieLens 100k三种算法运行时间比较

5 结束语

本文针对传统 k-近邻协同过滤推荐算法存在邻居数的确定过于主观和邻居推荐时的平均加权算法不具有针对性的问题,首先在推荐系统中设计了启发式聚类模型,根据数据集的特性客观的为目标用户划分邻居. 然后提出项目类别相似度的概念,对传统推荐时的平均加权算法进行根本改进,使目标用户对未知项目的基本评分更具有针对性,提高推荐性能. 基于这两个针对邻居选择和推荐的创新点,提出基于启发式聚类模型和类别相似度的协同过滤推荐算法,仿真实验佐证了这一算法的可行性,并且 CluC-CF 确实提高了推荐

准确度(较 Per-CF 提升约约 0.035MAE).然而数据集特性很差时,会存在孤立点自成一类的问题,本文下一步主要工作将进行这方面的研究.

参考文献

[1] 张锋,等.两方参与的隐私保护协同过滤推荐研究[J].电子学报,2009,37(1):84 - 89.
Zhang Feng,et al. Research on privacy-preserving two-party collaborative filtering recommendation[J]. Acta Electronica Sinica,2009,37(1):84 - 89. (in Chinese)

[2] 黄世平,黄晋,陈健,等.自动建立信任的防攻击推荐算法研究[J].电子学报,2013,41(2):84 - 89.
Huang Shi-ping, Huang Jin, Chen Jian, et al. Anti-attack recommender algorithm based on automatic trust establishment[J]. Acta Electronica Sinica, 2013, 41(2): 84 - 89. (in Chinese)

[3] 吴永辉,等.基于主题的自适应、在线网络热点发现方法及新闻推荐系统[J].电子学报,2010,38(11):2620 - 2624.
Wu Yong-hui, et al. Adaptive on-line web topic detection method for web news recommendation system[J]. Acta Electronica Sinica, 2010, 38(11): 2620 - 2624. (in Chinese)

[4] 韩立新.对搜索引擎中评分方法的研究[J].电子学报,2005,33(11):2094 - 2096.
Han Li-xin. A study on the ranking method of search engines[J]. Acta Electronica Sinica, 2005, 33(11): 2094 - 2096. (in Chinese)

[5] Song Y, Zhang L, et al. Automatic tag recommendation algorithm for social recommender systems[J]. ACM Transactions on the Web, 2011, 5(1): 4 - 39.

[6] 李聪.电子商务协同过滤可扩展性研究综述[J].现代图书情报技术,2010,(11):37 - 44.
Li Cong. Review of scalability problem in ecommerce collaborative filtering[J]. Modern Library and Information Technology, 2010, (11): 37 - 44. (in Chinese)

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

[8] Truong K, Ishikawa F, Honiden S. Improving accuracy of recommender system by item clustering[J]. IEICE-Trans Inf Syst, 2007, E90-D(9): 1363 - 1373.

[9] 张莉,秦桃,滕丕强.一种改进的基于用户聚类的协同过滤算法[J].情报科学,2014,32(10):24 - 28.
Zhang Li, Qin Tao, Teng Pi-qiang. An improved collabrative filtering algorithm based on user clustering[J]. Information Science, 2014, 32(10): 24 - 28. (in Chinese)

[10] Rashid A M, Lam S K, et al. ClustKNN: A highly scalable hybrid model & memory-based CF algorithm[A]. Proceedings of the KDD Workshop on Web Mining and Web Usage Analysis[C]. Philadelphia, Pennsylvania: ACM, 2006. 1-59593-444-8.

[11] George T, Merugu S. A scalable collaborative filtering framework based on co-clustering[A]. Proceedings of the Fifth IEEE International Conference on Data Mining[C]. Washington DC: IEEE Computer Society, 2005. 625 - 628.

[12] 李弼程,邵美珍,黄杰.模式识别原理与应用[M].西安电子科技大学出版社,2008.2.
Li Bi-cheng, Shao Mei-zhen, Huang Jie. Principle and Application of Pattern Recognition[M]. Xidian University Publisher, 2008. 2. (in Chinese)

[13] Haralambos, Marmanis, Dmitry. 智能 Web 算法[M].电子工业出版社,2011.7.

[14] KRZYWICKI A, WOBCKE W, CAI X. Interaction-based collaborative filtering methods for recommendation in online dating[A]. Web Information Systems Engineering-WISE 2010[C]. Berlin Heidelberg: Springer, 2010. 342 - 356.

[15] ZHANG J Y, PEARL P. A recursive prediction algorithm for collaborative filtering recommender systems[A]. Proceedings of the 2007 ACM Conference on Recommender Systems[C]. ACM, 2007. 57 - 64.

作者简介

王兴茂　男,1989 年生于辽宁营口,国家数字交换系统工程技术研究中心硕士生,主要研究方向为数据挖掘、用户行为分析、推荐算法.
E-mail: wxmcat@163.com

张兴明　男,1963 年生于河南新乡,国家数字交换系统工程技术研究中心教授,主要研究方向为通信与信息系统、宽带信息网络等.