doi:10.13582/j.cnki.1672-9102.2018.01.011

基于用户兴趣的动态近邻协同过滤算法

陈汝,符琦*

(湖南科技大学 计算机科学与工程学院,湖南 湘潭 411201)

摘　要:为了帮助人们从大量互联网资源中找到感兴趣的信息,推荐系统由此而生.其中,应用最广泛,也是最早出现的推荐算法包括协同过滤算法,但是该算法还存在着许多不足之处.该算法主要考虑用户的评分数据,未能结合项目进行考虑,同时在选取当前用户的最近邻用户时,通常统一规定了近邻用户数目,没有结合每个用户的实际数据,导致推荐的效果无法取得最优.因此,本文在充分考虑用户评分的情况下,还结合项目信息加入了用户的兴趣偏好,提出了一种基于用户兴趣的动态近邻协同过滤算法.综合用户的标签数据和评分数据来计算相似度,可以很好的缓解仅依靠评分数据带来的稀疏性问题.同时在得到用户之间的相似度之后,设定 2 个阀值,分布选取最近邻用户.只有当用户间相似度超过阈值,该用户才会被选择为最近邻的用户,动态的找到每一个用户的严格最近邻用户.通过实验,与常用的协同过滤算法相比,本文提出的算法推荐的误差更小,并且为以后的研究工作奠定了基础.

关键词:协同过滤;项目标签;用户兴趣;动态近邻;矩阵稀疏性

中图分类号:TP301　　**文献标志码**:A　　**文章编号**:1672-9102(2018)01-0063-08

Collaborative Filtering Algorithm for Dynamic Neighborhood Based on User Interest

Chen Ru, Fu Qi

(School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China)

Abstract: The recommended system help users to find what they want in a large amount of information. Collaborative filtering is one of the most widely used and oldest recommended technology. However, the algorithm mainly considers the score between users, fails to fully consider the labels of the items, and the user nearest neighbor number is unified, so the recommended result is not very good. A dynamic neighbor of collaborative filtering algorithm, based on items labels, was proposed , and the similarity between users was calculated by rates and labels with their respective weight, the similarity algorithm effectively reduced the sparsity of the similarity matrix of user rates. At the same time, after calculating the similarity of the user, by setting the similarity threshold, the nearest neighbor users of each user was dynamically found. At this time, only the similarity was greater than the similarity threshold, which was the nearest neighbor. Compared with the user-based collaborative filtering algorithm, it was verified that the algorithm got better recommendation results and layed the foundation for further research work.

Keywords: collaborative filtering; item tags; user interest; dynamic neighbor; matrix sparsity

随着互联网、移动技术的快速发展,网络用户逐渐地步入信息过载[1]的时代.一方面,海量的信息提供了丰富的资料开拓人们的视野;另一方面,对于用户而言,要在海量的信息里迅速并且准确的找到自己感兴趣的信息也并非易事,同样对于供应商而言,应该怎样让信息能够备受关注,得到用户们的喜爱也并不

收稿日期:2017-01-08

基金项目:湖南省自然科学基金资助项目(2017JJ2081);湖南省教育厅科学研究资助项目(17C0646)

* **通信作者**,E-mail:1524377@ qq.com

是件容易的事.推荐系统[2-3]可以非常有效的缓解当前信息过载带来的不便.

用户的个性化推荐主要是依据用户信息、物品信息以及用户与物品之间的联系,通过分析这些信息数据,实现对用户感兴趣信息的推荐.协同过滤[4-5]是推荐算法中较为成功的典例,通常分为 2 大类[6]:基于内存的协同过滤和基于模型的协同过滤.基于内存的协同过滤算法主要是通过分析用户的历史记录(购买,点击,观看,浏览等)计算用户或者项目的相似度,取与当前用户相似度较高的前几项对用户进行推荐.基于模型的协同过滤算法主要是解决用户对未评分项目的感兴趣程度,算法通过对用户已评分数据进行建模学习,预测目标用户对没有进行过评分行为的项目的评分,利用评分数据代表用户对项目的感兴趣程度,实现对用户的推荐.

在实际生活中,用户和项目的数目一般十分巨大,大多数项目只有小部分的用户进行过评分,因此用户间会因为共同评分项目数量极小而无法进行比较[8].另外,用户的评分标准各不相同,导致评分难以控制在同一个范围标准,数据的可靠性难以保证.

为了提高推荐算法的质量,实现更加精准和个性化的推荐,人们在已有的一些算法上做出了各种优化.在文章[9]中,考虑到用户评分矩阵的稀疏性问题以及冷启动问题,论文以用户的评分数据为基础建立一个假设的信任关系来进行改进,但是该方法中的用户信任关系并不能完全代表在社交中的信任关系,因此还需进一步计算用户之间的信任关系;在文章[10]中,采用双重标准来选取近邻用户从而实现用户推荐.根据用户的标签动态的选取与当前用户在兴趣上最相近的用户,并且建立一个信任关系,筛选出最近邻用户集合,最后利用可信的近邻用户的评分数据实现对目标用户的推荐.在协同过滤算法中,应用广泛的是 k 近邻模型,但是在相似度差别较大的情况下,k 值的选取往往会影响到最后的推荐效果,在文章[11]中,作者通过系统中项目的数量来选取合适的 k 值,采用相似度支持度来获取 k 近邻,通过对相似度支持和评分的相似度综合考虑,在保证推荐效果的情况下,推荐算法的计算复杂度有明显的降低.由于在协同过滤算法中,用户的评分数据往往很稀疏,因此在文章[12],Song Zhang 等人提出了基于粗糙集理论的评分预测方法,该方法通过使用粗糙集理论填补用户评分矩阵中的缺失值来解决数据稀疏性问题.协同过滤基于用户之间的喜好相似度而不是客观属性提供个性化的建议,这使得它能够推荐任何类型的项目,比如文本、音乐、视频和照片[13].文献[14]通过构建用户-标签-项目关系图,有效的缓解了评分数据稀疏性的问题,同时对于新加入的项目,可以根据项目的标签属性向用户进行推荐,对于冷启动问题的解决也有一定的贡献.Melville[15]等人通过分析用户与项目的文本信息,在用户的评分数据中增添一个额外的分数,结合额外加分高的用户的项目信息,将这些项目作为推荐给用户的最佳选择.

针对已有算法的不足之处,本文提出一种基于用户兴趣的动态近邻协同过滤推荐算法.该方法通过分析用户已有过评分行为项目的标签,利用 TF-IDF 算法的思想提取用户对标签的兴趣偏好,根据评分数据和兴趣偏好计算用户间的综合相似度,并且设定合适的阀值,找到每个用户的严格近邻邻居,对当前用户进行评分预测,最后将评分高的前几个项目推荐给用户,从而实现个性化推荐.

1 基于用户的协同过滤推荐

User CF 的基本思想是,通过用户已有评分数据找到与他最相近的邻居,然后根据这些邻居的评分预测出该用户没有进行过评分项目的评分,为当前用户推荐评分高的前几个项目.

User CF 主要分为 3 步:

1)计算用户间的相似度;

2)通过当前用户与其他用户的相似度值找出当前用户的最近邻用户;

3)利用当前用户的最近邻用户的评分数据计算出当前用户对项目的评分.

1.1 相似度度量的方法

User CF 算法首先需要找出当前用户的最近邻用户,现在最常用的度量方法有余弦相似度、皮尔逊相关系数和修正的余弦相似度.

1.1.1　余弦相似度

余弦相似度通过 2 个向量的夹角余弦值来度量 2 个个体之间的差异性,其值越大表示他们越相似.在协同过滤推荐算法中,用户 u_a 对项目 $I=\{i_1,i_2,i_3,\cdots,i_m\}$ 的评分可以表示为评分向量 $\boldsymbol{r}_{u_a}=(r_{a1},r_{a2},r_{a3},\cdots,r_{am})$,用户 u_b 对项目 $I=\{i_1,i_2,i_3,\cdots,i_m\}$ 的评分可以表示为评分向量 $\boldsymbol{r}_{u_b}=(r_{b1},r_{b2},r_{b3},\cdots,r_{bm})$,则用户 u_a 和用户 u_b 余弦相似度由式(1)计算:

$$\mathrm{sim}(u_a,u_b)=\cos(\boldsymbol{r}_{u_a},\boldsymbol{r}_{u_b})=\frac{\boldsymbol{r}_{u_a}\cdot\boldsymbol{r}_{u_b}}{\boldsymbol{r}_{u_a}\times\boldsymbol{r}_{u_b}}. \tag{1}$$

1.1.2　皮尔逊相关系数

在 CF 算法中,用户之间的相似度值还可以通过皮尔逊相关系数来表示,假设用户 u_a 和用户 u_b 的共同评分项目集合为 $\boldsymbol{I}_{ab}$,$\overline{r_a}$,$\overline{r_b}$ 分别表示 u_a 和 u_b 的平均评分,那么 u_a 和 u_b 的皮尔逊相关系数表达式为

$$\mathrm{sim}(u_a,u_b)=\frac{\sum_{i\in\boldsymbol{I}_{ab}}(r_{ai}-\overline{r_a})(r_{bi}-\overline{r_b})}{\sqrt{\sum_{i\in\boldsymbol{I}_{ab}}(r_{ai}-\overline{r_a})^2}\sqrt{\sum_{i\in\boldsymbol{I}_{ab}}(r_{bi}-\overline{r_b})^2}}. \tag{2}$$

1.1.3　修正的余弦相似度

在实际生活中,余弦相似度没有考虑到每个用户的评分标准可能不一样,假设当前情况下,项目的评分区间为 15,用户甲可能一直评分较低,所以可能他的 3 分就代表着喜欢,而用户乙习惯打高分,他的 3 分表示一般,因此这 2 个用户评分尺度差异比较大,评分的差异会在计算用户相似度时造成偏差.修正的余弦相似性通过平均值可以有效的解决评分差异性带来的问题.其具体如式(3):

$$\mathrm{sim}(u_a,u_b)=\frac{\sum_{i\in\boldsymbol{I}}(r_{ai}-\overline{r_a})(r_{bi}-\overline{r_b})}{\sqrt{\sum_{i\in\boldsymbol{I}}(r_{ai}-\overline{r_a})^2}\sqrt{\sum_{i\in\boldsymbol{I}}(r_{bi}-\overline{r_b})^2}}. \tag{3}$$

1.2　近邻居推荐

在 CF 算法中,普遍认为有着兴趣相似的用户的评分数据也会比较相近,因此通过式(3)可以找到与 u_a 相似度最大的前 k 个用户作为最近邻居,记为 $S(a,k)$,然后通过式(4)进行评分预测.

$$\mathrm{pre}(u_a,i)=\overline{r_a}+u_a\frac{\sum_{\mathrm{bes}(a,k)}\mathrm{sim}(u_a,u_b)(r_{bi}-\overline{r_b})}{\sum_{\mathrm{bes}(a,k)}\mathrm{sim}(u_a,u_b)}. \tag{4}$$

式(4)可以预测用户没有评过分项目的评分将评分进行排序,把前 n 项项目推荐给用户 u_a,即实现了对用户 u_a 的个性化推荐.

2　基于用户兴趣的动态近邻协同过滤推荐算法

在 User CF 算法中,主要考虑用户的评分数据从而得到用户间的相似度,但是,实际生活中,用户和项目的数量往往是非常庞大的[16],用户并不会对每个项目进行评分,所以评分矩阵成为了一个高维的稀疏矩阵,在寻找用户的近邻用户时会带来误差,从而影响最后推荐质量[17].而标签可以发挥集体的智慧,项目的标签可以转换为用户的兴趣偏好,从而可以得到用户的兴趣相似度,同时考虑用户的评分情况,得到综合相似度.然后通过设定 2 个相似值找到用户最严格的近邻用户,即首先根据综合相似度,找到大于设定的第 1 个相似值的用户作为选择近邻用户,其次在选择近邻用户中找到相似度大于设定的第 2 个相似值的用户,该用户即为最严格的近邻用户.

2.1　用户兴趣偏好计算

用户评分项目的标签在一定程度上可以反映出用户对不同类型项目的感兴趣程度,标签的频率可以表示用户的兴趣偏好,频率越高,表示用户对标有该标签的项目更加感兴趣.因此,文中通过 TF-IDF 算法,将标签出现的频率转换为用户的兴趣偏好.

TF-IDF 算法主要思想:假如某一字词在一篇文档中出现的频率很高,并且在其他文档中出现很少,那么这个字词可以很好的将该文档区别于其他文档.文中用户标签集合即为用户对应项目标签的集合,将用户标签集合作为一个文档,项目的标签作为字词.因此每个用户的标签集合中标签的个数即为文档中单词出现的次数,根据用户标签集合计算用户对某一标签的偏好,得到用户的兴趣偏好.

假设在推荐系统中,存在 1 个包含 n 个用户的用户集 $\boldsymbol{U}$, 1 个含有 m 个项目的项目集 $\boldsymbol{I}$ 和一个包含 k 个标签的标签集 $\boldsymbol{T}$,其中定义每个用户 $u(u \in \boldsymbol{U})$ 有过行为的项目集合为 $\boldsymbol{I}_u$, 且 $\boldsymbol{I}_u \in \boldsymbol{I}$, 每个项目 $i(i \in \boldsymbol{I})$ 包含一个标签集合 $\boldsymbol{T}_i$, 且 $\boldsymbol{T}_i \in \boldsymbol{T}$, 由此,每个用户也会对应一个标签集合 $\boldsymbol{T}_u = \{T_1, T_2, T_3, \cdots, T_i\}$, 其中 $i \in \boldsymbol{I}_u$.

用户对项的标签的感兴趣程度表示了用户的兴趣偏好.每个标签在用户 u_a 的标签集中出现的频率表示了该标签对当前用户的“重要性”,可以由词频 TF_{u_at} 来计算:

$$\mathrm{TF}_{u_at} = \frac{n_{u_at}}{\mathrm{count}(T_{ua})}. \tag{5}$$

式中: n_{u_at} 为标签 t 在用户 u_a 的标签集合 T_u 里出现的次数; $\mathrm{count}(T_{u_a})$ 为用户 u_a 的标签集合 T_{u_a} 里的标签总数.

$$\mathrm{IDF}_t = \log\left(\frac{n_u}{n_{u_t} + 1}\right). \tag{6}$$

式中: n_u 为推荐系统中的用户个数; n_{u_t} 为包含标签的用户的数目.因此,用户 u_a 对的兴趣偏好可以由式(7)定义:

$$q_{u_at} = \mathrm{TF}_{u_at} \times \mathrm{IDF}_t. \tag{7}$$

用户 u_a 的兴趣偏好即用户 u_a 对标签 $T = (t_1, t_2, t_3, \cdots, t_k)$ 的偏好组成用户 u_a 的兴趣偏好向量: $Q_{u_a} = (q_{u_a1}, q_{u_a2}, q_{u_a3}, \cdots, q_{u_ak})$.

2.2 用户相似度计算

2.2.1 用户评分相似度

给定用户集合 $\boldsymbol{U}$ 以及项目集合 $\boldsymbol{I}$, 用户-项目的评分矩阵 $\boldsymbol{R}$ 表示为 $\boldsymbol{R} = |\boldsymbol{U}| \times |\boldsymbol{I}|$. 在 $\boldsymbol{R}$ 中,一个行向量代表一个用户的评分向量,用户 u_a 对项目 $\boldsymbol{I} = \{i_1, i_2, i_3, \cdots, i_m\}$ 的评分表示为评分向量 $\boldsymbol{r}_{u_a} = (r_{a1}, r_{a2}, r_{a3}, \cdots, r_{am})$,用户 u_b 对项目 $\boldsymbol{I} = \{i_1, i_2, i_3, \cdots, i_m\}$ 的评分可以表示为评分向量 $\boldsymbol{r}_{u_b} = (r_{b1}, r_{b2}, r_{b3}, \cdots, r_{bm})$,通过式(8)计算用户的评分相似性:

$$S(u_a, u_b) = \frac{\sum_{i \in \boldsymbol{I}} (r_{ai} - \overline{r_a})(r_{bi} - \overline{r_b})}{\sqrt{\sum_{i \in \boldsymbol{I}} (r_{ai} - \overline{r_a})^2} \sqrt{\sum_{i \in \boldsymbol{I}} (r_{bi} - \overline{r_b})^2}}. \tag{8}$$

式中: $S(u_a, u_b)$ 为用户 u_a 和用户 u_b 的评分相似度; $\overline{r_a}, \overline{r_b}$ 为用户 u_a 和用户 u_b 对项目的平均评分.

2.2.2 用户兴趣偏好相似度

如果用户 u_a 和 u_b 的兴趣偏好相似度的值越高,说明他们的兴趣爱好越接近,他们感兴趣的项目也会存在很多相同的.式(9)可以计算用户在兴趣上的相似性:

$$C(u_a, u_b) = \frac{Q_{u_a} Q_{u_b}}{Q_{u_a} Q_{u_b}}. \tag{9}$$

式中: $C(u_a, u_b)$ 为用户 u_a 和用户 u_b 的兴趣相似度; Q_{u_a}, Q_{u_b} 为用户 u_a 和用户 u_b 的兴趣偏好.

2.2.3 用户综合相似度

由于评分数据很稀疏,仅靠评分数据得到用户间的相似度会带来误差,文中在计算用户之间的相似性同时考虑了用户评分相似性和兴趣偏好相似性.文中通过定义因子来分配 2 种相似性之间的权重,根据 α 的取值调整 $S(u_a, u_b)$ 和 $C(u_a, u_b)$ 对综合相似度的权重.调整之后的相似度计算如式(10):

$$\mathrm{sim}(u_a,u_b)=\alpha S(u_a,u_b)+(1-\alpha)C(u_a,u_b). \tag{10}$$

2.3 评分预测计算

2.3.1 基于用户的协同过滤算法评分预测

User CF 中,通过计算用户对项目的兴趣度之后,再筛选出 u_a 未评分的项目集合 $\boldsymbol{N}_a$, 同时由式(3)和式(4)计算出用户间的相似度,选择与用户 u_a 相似度最大的前 k 个用户组成该用户的最相近的用户集 $\boldsymbol{C}_a$, 通过以下公式得到目标用户 u_a 对未进行过评分项目 $i(i\in\boldsymbol{N}_a)$ 的评分:

$$\mathrm{pre}(u_a,i)=\overline{r_a}+\frac{\sum_{u_b\in C_a\cap C(i)}s(u_a,u_b)\times(r_{bi}-\overline{r_b})}{\sum_{u_b\in C_a\cap C(i)}s(u_a,u_b)}. \tag{11}$$

式中:$\overline{r_a}$ 为用户 u_a 评分的平均分;r_{bi} 为用户 u_b 对项目 i 的评分;$\boldsymbol{C}(i)$ 为对项目 i 评过分的用户集合.

2.3.2 基于用户兴趣的动态近邻协同过滤算法评分预测

CF 算法中,主要是通过用户评分来计算用户间的相似度,然而在实际生活中,用户评分数据很少,仅依靠评分数据计算用户间的相似度会带来误差,因此推荐效果不佳.文中在计算用户相似度时不仅考虑评分数据,还引入了用户的兴趣偏好.同时传统协同过滤算法中,当前用户的最近邻用户数目是统一规定的,即每个当前用户的最近邻用户都是 k 个,但是,每个用户的评分和兴趣都不相同,它们的近邻用户也各不尽相同,可能用户的最近邻用户不止定义的 k 个,或者用户的 k 个近邻用户中存在着与该用户相似度极低的用户,这些都会给评分的预测带来误差,导致较差的推荐效果.因此,在本文中设定 2 个阀值 β 和阀值因子 μ,$\overline{\mathrm{sim}_a}$ 表示用户 u_a 的综合相似度的平均值,$\overline{S_a}$ 表示用户 u_a 的平均评分相似度.首先找到相似度大于 $\beta\overline{\mathrm{sim}_a}$的用户集合,其次在该用户集合中找到相似度大于 $\mu\overline{S_a}$ 的用户作为严格近邻用户,即 $\mathrm{sim}(u_a,u_b)\geqslant\beta\overline{\mathrm{sim}_a}$,$S(u_a,u_b)\geqslant\mu\overline{S_a}$,其中阀值因子 $\beta\in(0,2)$,$\mu\in(0,2)$. 评分预测式(12)所示:

$$\mathrm{Ipre}(u_a,i)=\overline{r_a}+\frac{\sum_{u_b\in C_{a\beta\mu}\cap C(i)}\mathrm{sim}(u_a,u_b)\times(r_{bi}-\overline{r_b})}{\sum_{u_b\in C_{a\beta\mu}\cap C(i)}\mathrm{sim}(u_a,u_b)}. \tag{12}$$

式中:$\mathrm{Ipre}(u_a,i)$ 表示用户 u_a 对项目 i 的预测评分;$C_{a\beta\mu}$ 表示用户 u_a 的严格近邻用户集合.

2.4 推荐

通过计算得到当前用户对未评分项目的评分之后,就可选择预测分值最高的前 N 项项目对用户进行推荐.

算法 1 基于用户兴趣的动态近邻协同过滤推荐算法

输入:评分矩阵 $\boldsymbol{R}(n,m)$,项目标签集合 $\boldsymbol{T}_i$,用户标签集合 $\boldsymbol{T}_u$,目标用户 u_a,权重因子 α,阀值 β,阀值 μ.

输出:目标用户 u_a 的 top-N 推荐集 Ipre.

过程:

Step 1:根据式(7),用户标签集合和 TF-IDF 算法,得到用户的兴趣偏好,由此得出一组用户兴趣偏好向量 $Q_{u_{a1}},Q_{u_{a2}},Q_{u_{a3}},\cdots,Q_{u_{an}}$

Step 2:根据式(8)和式(9)分别计算用户在评分和兴趣偏好上的相似度,分配合适权重因子 α, 得出用户最后的综合相似度;

Step 3:根据 step 2 得到的用户相似度,再设定阀值 β 和 μ, 首先找到综合相似度大于 $\beta\overline{\mathrm{sim}_a}$ 的用户,作为选择近邻用户群,然后在用户的选择近邻用户群中找到评分相似度大于 $\mu\overline{S_a}$ 的用户,作为严格近邻用户;

Step 4:根据式(12)预测当前用户 u_a 没有评过分的项目的评分 $\mathrm{Ipre}(u_a,i)$;

Step 5: 将 Step 4 中计算所得评分按降序排列;

Step 6: 将评分排名的前项项目作为目标用户的推荐集 Ipre.

3 实验及分析

3.1 实验数据

文中实验采用的数据集是由 GroupLens 提供的 MovieLens 数据集,该数据集有 3 个不同的版本,文中选取的是较轻量级的数据集.一共包含 100 004 条评分数据,涉及到 671 个用户和 9 125 部电影,其中每个用户的评分数据最少为 20 条,并且评分取值为 15 分,同时还含有项目的 19 个标签.

3.2 推荐质量评价标准

有许多评价方法可以用来衡量推荐效果的性能,文中算法采用平均绝对误差和均方根误差评测推荐的性能.对于测试集中的用户 u 和项目 i,r_{ui} 表示 u 对 i 的实际评分,$\widehat{r_{ui}}$ 是通过算法计算得到的评分,那么 MAE 和 RMSE 定义如式(13)和式(14):

$$\mathrm{MAE}=\frac{\sum_{u,i\in T}|r_{ui}-\widehat{r_{ui}}|}{|T|}; \tag{13}$$

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{u,i\in T}(r_{ui}-\widehat{r_{ui}})^2}{|T|}}. \tag{14}$$

MAE 和 RMSE 的值越小,表示当前预测评分的误差越小,由此推荐的质量越高.

3.3 结果与分析

3.3.1 参数调整

在文中实验,通过引入了一个相似度因子 α,$\alpha\in[0,1]$,权衡用户的评分相似度和兴趣相似度. α 的值决定了推荐的质量,当 α 取值过大时,会倾向于 User CF 算法,取值过小则体现不出用户的兴趣偏好对推荐的影响.当 $\alpha=0$,忽略了用户的评分数据;当 $\alpha=1$,则为 User CF 算法.在本文实验中 α 的取值为 0.5. 实验结果如图 1 和图 2 所示.

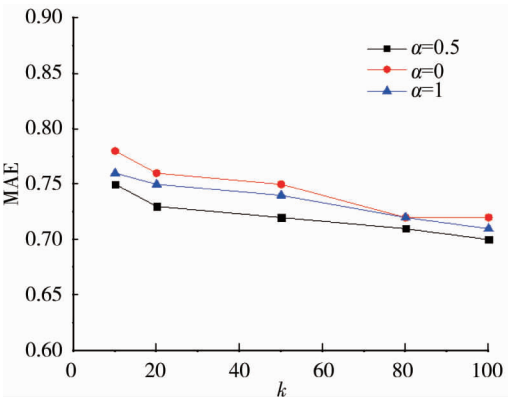

图 1 相似度因子 α 对 MAE 的影响

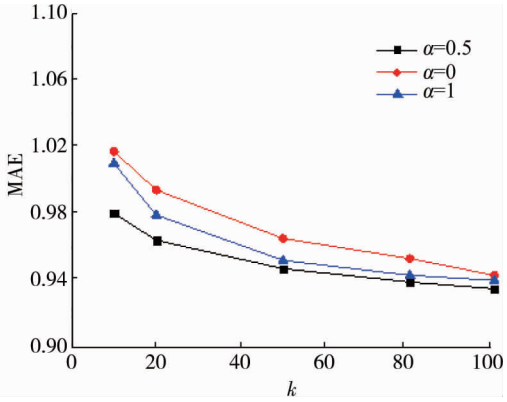

图 2 相似度因子 α 对 RMSE 的影响

从图 1 和图 2 可以看出,仅依靠用户的评分数据或者用户兴趣偏好的推荐算法,其误差比结合评分数据和兴趣偏好算法的大.

在文中提出的算法中,规定当用户间综合相似的程度大于 $\beta\overline{\mathrm{sim}_a}$,并且评分相似度大于 $\mu\overline{S_a}$ 的用户,作为严格近邻用户. β 和 μ 为阀值,$\beta\in(0,2)$,$\alpha\in(0,2)$ 在文中实验.取 β 的值为 1,对阈值 μ 进行调参. 因此阀值 μ 会影响到每个用户的最近邻的个数,而用户的最近邻用户数目会影响到推荐的效果,因此在该实验中,将 μ 的值从 0~2 进行取值,当 μ 越小时,最近邻用户数目越大;相反最近邻用户数目越小.当 $\alpha=0.5$时,取不同的 μ 得到的 MAE 值和 RMSE 值不同,如图 3 和图 4 所示.

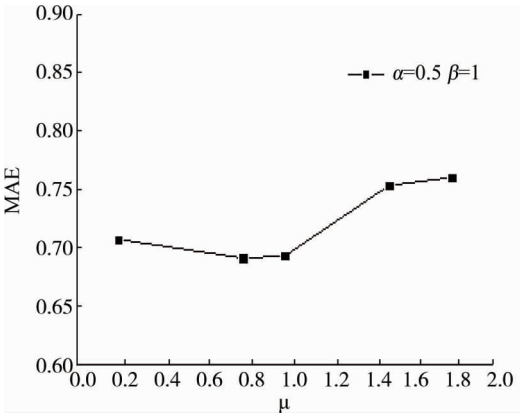

图 3　阀值 μ 对 MAE 的影响

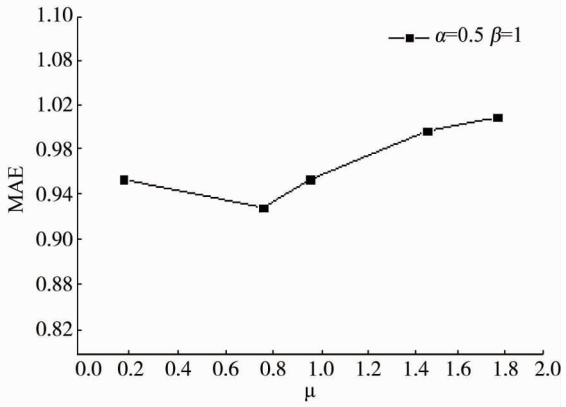

图 4　阀值 μ 对 RMSE 的影响

由上图可知, $\alpha = 0.5, \beta = 1, \mu$ 值取 0.8,MAE 和 RMSE 的值最小,当前误差最小.

3.3.2　实验结果比较

在文中算法,当 $\alpha = 0.5, \beta = 1, \mu$ 值取 0.8,MAE 和 RMSE 的值最小,即当前的推荐质量更高,然后在此基础上将文中提出的算法与 User CF 算法进行比较.实验中,User CF 算法里最近邻用户数目从 10 增到 100,结果如图 5 和图 6 所示.

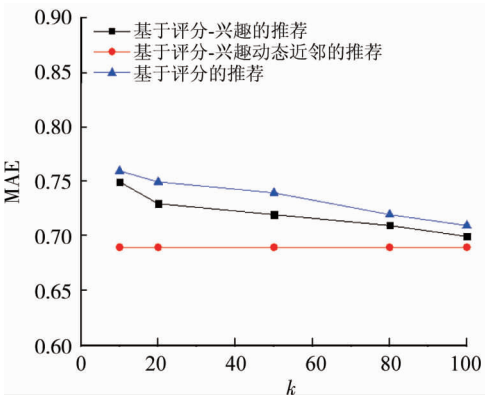

图 5　3 种推荐算法的 MAE 比较

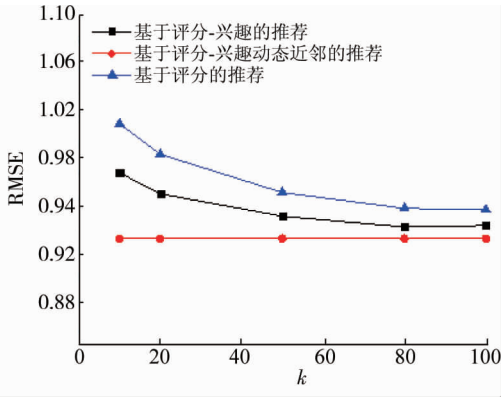

图 6　3 种推荐算法的 RMSE 比较

图 5 和图 6 中,基于评分-兴趣的推荐表示当 $\alpha = 0.5$ 时,根据用户的综合相似度计算得到相似用户,然后设定不同的近邻用户数目,由此进行评分预测.基于评分的推荐即为基于用户的协同过滤算法.基于评分-兴趣的动态近邻的推荐表示 $\alpha = 0.5, \beta = 1, \mu = 0.8$ 时的评分预测.

由图 5 和图 6 可知,综合考虑用户的评分数据和兴趣偏好,可以获得更小的 MAE 和 RMSE,这是因为引入了项目标签作为用户的兴趣,可以有效缓解评分数据不足带来的误差,因此可以寻找更加精确的最近邻邻居.通过项目的标签计算用户的兴趣偏好从而得到用户兴趣偏好的相似度,用户的兴趣偏好在协同过滤算法中也是需要考虑的因素,然后结合用户的评分得到综合相似度.同时,还通过设置相似度阀值,采用分布选取近邻用户,将统一的 k 个最近邻用户转换为动态的近邻用户,使得每位用户的最近邻用户都是严格的最近邻的.当 $\alpha = 0.5, \beta = 1, \mu = 0.8$ 时,评分预测结果更优.综上所述,文中提出的基于用户兴趣的动态近邻协同过滤算法可以取得更优的结果.

4　结论

1)在计算用户相似度时引入用户兴趣偏好,可以很好地缓解评分数据稀疏性的问题.

2)采用分布选择最近邻,使得用户可以寻找到严格的近邻用户群.

3)对于用户评分数据的稀疏性问题的有明显的改善,同时推荐的质量更好.

参考文献:

[1] Borchers A, Herlocker J, Konstan J, et al. Ganging up on information overload [J].Computer, 1998, 31(4): 106-108.

[2] Resnick P, Varian H. Recommender systems[J]. Communications of the ACM, 1997, 40(3): 56-58.

[3] Chen R M. Challenge, Value and coping strategy in the big data era[J]. Mobile Communications, 2012, 36(17): 14-15.

[4] Konstan J A.Introduction to recommender systems: Algorithmsand evaluation[J].ACM Transactions on Information Systems, 2004,22 (1): 1-4.

[5] Schafer J B, Konstan J A, Riedl J. E-commerce recommendation applications [J]. Data Mining and Knowledge Discovery, 2001, 5(1/2): 115-153

[6] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C] //Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc,1998: 43-52.

[7] Kim H N, Ji A T, Ha I, et al. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation [J]. Electronic Commerce Research and Applications, 2010, 9(1): 73-83.

[8] Zivkovic Z. Improved adaptive Gaussian mixture model For background subtraction [C] // Pattern Recognition, International Conference on IEEE Computer Society, 2004:28-31.

[9] Pitsilis G, Knapskog S J. Social trust as a solution to address sparsity-inherent problems of recommender systems[C] // Proceedings of the 4th ACM Conference on Recommender Systems, New York: ACM, 2009: 332-344.

[10] 贾冬艳,张付志.基于双重邻居选取策略的协同过滤推荐算法[J].计算机研究与发展,2013,50(5):1076-1084.

[11] 罗辛, 欧阳元新, 熊璋,等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报, 2010, 33(8): 1437-1445.

[12] Zhang S, Li C, Ma L, et al. Alleviating the sparsity problem of collaborative filtering using rough set[J]. COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 2013, 32(2): 516-530.

[13] Kim H N, El-Saddik A, Jo G S. Collaborative error-reflected models for cold-start recommender systems[J]. Decision Support Systems, 2011, 51(3):519-531.

[14] Zhang Z K, Liu C, Zhang Y C, et al. Solving the cold-start problem in recommender systems with social tags [J]. Europhysics Letters, 2010, 92(2):28002-28007.

[15] Melville P, Mooney R J, Nagarajan R. Content-boosted collaborative filtering for improved recommendations [C]// Eighteenth national conference on Artificial intelligence. American Association for Artificial Intelligence, 2002:187-192.

[16] 陈炎龙,段红玉.基于改进 Hadoop 云平台的海量文本数据挖掘[J].湖南师范大学自然科学学报,2016,39(3):84-88.

[17] 莫晓云,周杰明,金芳.历史相依决策模型的建立及相应过程的构造[J].湖南师范大学自然科学学报,2017,40(5): 88-94.