一种基于 Bhattacharyya 系数和项目相关性的协同过滤算法

臧雪峰 刘天琦 孙小新 冯国忠 张邦佐
(东北师范大学计算机科学与信息技术学院 长春 130117)

摘 要 在大数据时代,为了满足用户的信息需求,个性化推荐系统得到了广泛应用。协同过滤是一种简单有效的推荐算法。然而,许多传统的相似度计算方法仅仅基于用户的共同评分值,且不适用于稀疏数据环境,因此提出了一种新的基于 Bhattacharyya 系数的相似度方法。该方法使用了所有用户对项目的评分信息,不仅可以通过用户的评分行为获得用户的相似兴趣特征,而且可以获得用户已评分物品之间的相关性;同时由于不同的用户有不同的评分习惯,新方法也考虑了每个用户的评分偏好。通过考虑用户相似性的更多因素,可以为目标用户选择更恰当的邻域用户,以更有效地提升推荐性能。在两个真实数据集上进行的实验表明,所提方法优于其他当前最好的相似度方法。

关键词 协同过滤,Bhattacharyya 系数,项目相关性,评分偏好
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.12.010

Collaborative Filtering Algorithm Based on Bhattacharyya Coefficient and Item Correlation

ZANG Xue-feng LIU Tian-qi SUN Xiao-xin FENG Guo-zhong ZHANG Bang-zuo
(College of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China)

Abstract In order to satisfy the information needs of users in the big data era, the personalized recommender system has been widely used. Collaborative filtering is a simple and effective recommendation algorithm. However, most traditional similarity methods only compute the similarity based on the users' co-rated scores. In addition, they are not very suitable in sparse data environment. This paper proposed a new similarity method based on Bhattacharyya coefficient. It uses all users' rating information for items, which can not only obtain similar interest feature of users through the user's rating behavior, but also obtain the correlation between the items that the users have rated. Meanwhile, the new method also takes into account each user's rating preference, since different users have different rating habits. Considering more relevant factor about user similarity, more appropriate neighborhood can be selected for the target users, efficiently improving the recommendations. With experiments on two real data sets, the results show that our method outperforms the other state-of-the-art similarity metrics.

Keywords Cllaborative filtering, Bhattacharyya coefficient, Item correlation, User preference

随着互联网的高速发展,互联网上的数据量呈爆炸式增长,从而使人们进入了"信息过载"的困境,快速且高效地从浩如烟海的数据中获得所需要的信息变得越来越困难。推荐系统是解决上述问题的有效手段。推荐系统通过收集和分析用户的各种历史行为数据来学习用户的兴趣和行为模式,以推荐用户需要的服务[1]。在大数据时代,推荐系统得到了广泛的应用,例如亚马逊为用户推荐其感兴趣的商品,Netflix 为用户推荐其可能喜欢的各种电影,YouTube 为用户推荐有趣的视频。目前流行的推荐算法主要分为四大类:基于内容的推荐算法[2]、协同过滤推荐算法[3]、基于知识的推荐算法[4]和混合推荐算法[5]。

协同过滤是目前应用得最广泛且最成功的推荐算法,主要包括基于记忆的方法和基于模型的方法。传统的基于记忆[6-7]的协同过滤算法又可以分为基于用户的协同过滤算法和基于物品的协同过滤算法。这两种算法的区别在于计算相似度时度量的对象不同,前者计算用户之间的相似度,找到具有相似爱好或兴趣的用户,比较适用于用户数变化不大且远小于项目数的情况;而后者从计算项目之间的相似度出发,找到有相似评分行为的项目,适用于用户数远多于项目数的情况。基于模型的方法先构建一个描述用户行为的模型,再预测用户对物品的评分,其优点是推荐性能优于基于记忆的方法,缺点是很难解释为什么向目标用户推荐这些项目。一旦

到稿日期:2016-10-11 返修日期:2016-11-12 本文受国家自然科学基金项目(71473035,11501095),吉林省科技厅重点攻关项目(20150204040GX),吉林省发改委项目(2015Y055),东北师范大学自然科学基金项目(2014015KJ004)资助。

臧雪峰(1991-),男,硕士生,主要研究领域为推荐系统,E-mail:zangxf466@nenu.edu.cn;刘天琦(1992-),女,硕士生,主要研究领域为推荐系统;孙小新(1978-),男,博士生,主要研究领域为智能信息处理;冯国忠(1983-),男,博士,讲师,主要研究领域为数据挖掘;张邦佐(1971-),男,博士,副教授,主要研究领域为数据库与数据挖掘、推荐系统。

建立了推荐模型，在推荐的过程中就不需访问评分数据，因此基于模型的推荐算法可以缓解数据稀疏导致的问题。由于推荐可以看作是分类或预测问题，因此基于模型的算法可以借助分类、聚类、回归等问题来实现。由于基于模型的推荐算法可以获得更高的推荐精度，因此对其的研究较多，矩阵分解等降维技术（诸如 PCA[8]，SVD[9]等）在近年来受到较多研究者的关注。

推荐系统通过搜集用户对物品的评分数据来进行推荐，当数据量较少时，存在冷启动和数据稀疏等问题。如果系统得到的评分数据非常稀疏，那么两个用户共同评分的项目将很少，从而导致难以准确地计算出用户间的相似度。然而，随着社交网络的快速发展，一些研究工作把信任数据加入系统中，以在一定程度上缓解评分数据的稀疏问题，很多研究都证明将用户之间的社交信任信息融入到推荐系统中可以更好地提高推荐精度[10-13]。这与日常生活中的一些情景类似，比如我们在看电影或购物时会征求朋友的意见，通过参考朋友的推荐可以获得更加满意的效果。

协同过滤的核心是计算用户或物品之间的相似度。传统的相似度计算方法较多，如皮尔森相关系数 pcc（Pearson correlation coefficient）[14]、余弦相似度（cosine）[15]、均方差 msd（mean square deviation）[16]，但它们较为简单，不能更好地衡量用户或物品的相似度，尤其是对评分很少的冷启动用户。本文主要基于 Bhattacharyya 系数融合传统的相似度来提高推荐精度，在计算用户之间的相似度时除了考虑用户的共同评分项，还考虑采用 Bhattacharyya 系数计算两个项目之间的相关性。由于每个用户的评分偏好也是影响相似度的一个重要因素，因此考虑该因素可以提高推荐的准确度。通过在两个真实数据集上的实验，验证了所提出的新相似度方法有更好的推荐性能。

本文首先介绍了近年来推荐系统方面的相关工作；接着提出了新方法，并对其进行了详细的描述；然后介绍了数据集的情况，并通过真实数据集验证了新方法的推荐效果，通过多组对照实验来分析不同参数对推荐结果的影响；最后总结全文，并展望未来。

1　相关工作

基于记忆的推荐算法又可称为基于邻域的推荐算法。本节将详细介绍基于邻域的方法和已有的各种相似度计算方法。

1.1　基于邻域的方法

基于邻域的方法在商业领域被广泛应用。该方法首先通过使用用户-项目评分矩阵来计算用户或项目之间的相似度，然后构造目标用户或物品的邻域信息，最后根据邻域信息计算用户对目标物品的估计评分，并据此为用户推荐最相关的物品或项目列表。

本文考虑一个 $m\times n$ 的评分矩阵（有 m 个用户和 n 个项目），用户集合 $U=\{u_1,u_2,\cdots,u_m\}$，项目集合 $I=\{i_1,i_2,\cdots,i_m\}$，用户-项目评分矩阵 $R=(r_{ij})_{m\times n}(i=1,2,\cdots,m;j=1,2,\cdots,n)$，$|U|$ 和 $|I|$ 分别表示用户数和项目数。真实情况下，由于每个用户只会对部分项目评分，因此最终获得的数据集非常稀疏。

基于用户的推荐算法为目标用户 a 预测对给定项目 i 的评分 $\hat{r}_{a,i}$。该方法首先计算用户 a 与其他用户间的相似度，选取其他用户中为物品 i 评过分且与目标用户 a 最相似的 k 个用户作为邻域用户。对物品 i 的预测评分 $\hat{r}_{a,i}$ 为：

$$\hat{r}_{a,i}=\bar{r}_a+\frac{\sum_{b\in N}sim(a,b)(r_{b,i}-\bar{r}_b)}{\sum_{b\in N}sim(a,b)} \tag{1}$$

其中，$\bar{r}_a$ 和 $\bar{r}_b$ 分别表示用户 a 与 b 的评分均值，$sim(a,b)$ 表示用户 a 与 b 之间的相似度，$r_{b,i}$ 表示用户 b 对项目 i 的评分，N 表示用户 a 的邻域用户集合。

基于物品的协同过滤与基于用户的方法类似，区别在于其计算的是目标项目 i 与其他被用户 a 评过分的物品之间的相似度，然后找到与项目 i 最相似的 k 个项目作为邻域。用户 a 对项目 i 的预测评分 $\hat{r}_{a,i}$ 为：

$$\hat{r}_{a,i}=\bar{r}_i+\frac{\sum_{j\in K}sim(i,j)(r_{a,j}-\bar{r}_j)}{\sum_{j\in K}sim(i,j)} \tag{2}$$

其中，$\bar{r}_i$ 和 $\bar{r}_j$ 表示项目 i 和项目 j 的评分均值，$sim(i,j)$ 表示项目 i 和项目 j 之间的相似度，K 表示项目 i 的邻域项目集合。

在协同过滤推荐系统中，相似度计算是最重要的步骤，是推荐过程的核心。相似度计算在很多领域得到广泛应用。推荐系统方面的很多研究工作均直接使用传统的相似度，或者提出新的相似度方法。下文将介绍一些已有的相似度方法。

1.2　相似度方法

协同过滤推荐算法中常用的相似度计算方法有余弦相似度、修正的余弦相似度和 Pearson 相关系数。如果把用户-项目评分矩阵看作空间中的向量，那么可以利用这些向量描述用户的兴趣特征，而向量之间的余弦夹角可以度量用户之间的相似度，如式(3)所示：

$$\cos(a,b)=\frac{\vec{r}_a\cdot\vec{r}_b}{\|\vec{r}_a\|\cdot\|\vec{r}_b\|} \tag{3}$$

其中，$\vec{r}_a$ 和 $\vec{r}_b$ 分别表示用户 a 和 b 的评分向量，$\|\cdot\|$ 表示向量的长度。

余弦相似度计算公式并未考虑用户的评分偏好，修正的余弦相似度通过减去用户对项目的评分均值来弱化偏好的影响，如式(4)所示：

$$Acos(a,b)=\frac{\sum_{i\in L}(r_{a,i}-\bar{r}_a)(r_{b,i}-\bar{r}_b)}{\sqrt{\sum_{i\in I_a}(r_{a,i}-\bar{r}_a)^2}\cdot\sqrt{\sum_{i\in I_b}(r_{b,i}-\bar{r}_b)^2}} \tag{4}$$

其中，L 表示用户 a 和 b 共同评分的物品集合，I_a 和 I_b 分别表示用户 a 和 b 的评分物品集合。

Pearson 相关系数可以度量用户或项目之间的相关性，计算结果的取值范围为 $[-1,1]$，其值越大表示用户或项目越相似，计算公式如式(5)所示：

$$pcc(a,b)=\frac{\sum_{i\in L}(r_{a,i}-\bar{r}_a)(r_{b,i}-\bar{r}_b)}{\sqrt{\sum_{i\in L}(r_{a,i}-\bar{r}_a)^2}\cdot\sqrt{\sum_{i\in L}(r_{b,i}-\bar{r}_b)^2}} \tag{5}$$

当用户共同评分的项目较少时，Pearson 相关系数得到的结果较差。

除了上述 3 种常用的相似度方法，Jaccard 和均方差(msd)也是比较常用的方法。Jaccard 方法仅考虑了两用户共同评分的项目数，并未考虑用户对项目评分的数值；而均方差恰恰与其相反。上述相似度方法在计算过程中都面临一些问题，在推荐系统中仅使用上述方法时的应用效果欠佳，尤其是对稀疏数据的处理。为了提高推荐系统的推荐精确度，较多研究者从不同的角度提出了多种不同的相似度计算方法。

黄创光等人[17]提出的算法基于用户以及项目的相似性计算，自适应地选择目标对象的邻域对象作为推荐集，同时选择推荐集中被推荐概率较高的对象作为信任子集，最后通过不确定近邻的动态度量方法来对预测结果进行平衡的推荐。文献[18-19]根据用户的隐式行为信息寻找目标用户的邻域对象，提高了推荐系统的精确度。Strehl A 等人[20]考虑了用户之间共同评分物品所占的比例，将 Jaccard 和 pcc 进行了融合。由于用户之间共同评分项目的数量不同，Herlocker 等人[21]提出了一种基于权重的 Pearson 相关系数计算方法，该方法对共同评分物品数设定一个阈值，是针对共同评分物品数的一种权衡机制。类似的延伸方法也被 Jamali 等人[22]所使用，他们在 Pearson 相关系数的基础上融合了 Sigmod 函数。另外，在推荐系统中可以判断用户评分的心态是积极的还是消极的，文献[23]据此提出了受限 Pearson 相关系数。Nikolas 等人[24]提出了一种多层推荐方法，根据用户间共同评分的项目数将推荐分为几个层次，目的是帮助用户做出更好的决策。Lu 等人[25]提出了一种基于模糊集的理论，对不同的评分差异分配不同的权重值。Luo 等人[26]将用户相似度分为了两部分：局部用户相似度和全局用户相似度。若用户之间通过一些相似的邻域用户连接，则他们将会变得更加相似。与之相对应的改进是 Liu 等人[27]提出的一种新的用户相似度模型，该模型不仅考虑了用户的局部上下文信息，还考虑了用户行为的全局偏好。Wang 等人[28]提出了在协同过滤中用熵来度量用户相似度的方法。上述大部分方法都是根据用户间共同评分的项目来计算相似度，而忽视了剩余的项目评分，即未使用所有的评分信息。此外，还有一部分方法只把用户对项目的评分作为相似度度量的唯一依据，在遇到数据稀疏或冷启动问题时推荐效果较差。

鉴于上述问题，本文提出了一种多因素复合度量的协同过滤推荐新方法。在计算用户间的相似度时，首先通过 Bhattacharyya 系数计算用户所评项目之间的相关性来反映用户之间的相似性，然后基于 Pearson 相关系数通过用户自身的评分行为来计算用户之间的相似性，最后融合两种相似性来获得更加精确的用户相似度。在研究过程中发现，用户的评分偏好也是很重要的因素，当把它们融合在一起时，将使目标用户得到更加准确的邻域用户；更重要的是，在数据稀疏的情况下，较好地解决了因相似度计算数据不充分而导致的推荐不准确问题。

2 提出的新相似度方法

本节首先详细介绍 Bhattacharyya 系数，然后再具体分析新方法的实现步骤。整个相似度计算过程是在用户-物品评分矩阵之上进行的，表 1 给出了一个评分矩阵的例子。

表 1 评分矩阵

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	4		3		3	
u_2		3	4	2	2	4
u_3		4	2	1		1
u_4	5	2	4		3	2
u_5		2		4		
u_6	1		3		5	

2.1 Bhattacharyya 系数

Bhattacharyya 系数可以度量两个离散或者连续概率分布的相似性[29]，被广泛应用于符号处理[30]、图像处理[31]和模式识别[32]等领域。如果 $p(x)$，$q(x)$ 表示两个在连续域上的概率分布，那么这两个概率分布之间的 Bhattacharyya 系数公式如式(6)所示：

$$BC(p,q)=\int \sqrt{p(x)q(x)}\,\mathrm{d}x \tag{6}$$

如果是在离散域上，那么两个概率分布的公式如式(7)所示：

$$BC(p,q)=\sum_{x\in X}\sqrt{p(x)q(x)} \tag{7}$$

其中，X 表示离散域，x 表示离散域中的取值。例如，评分矩阵中的评分范围 $X=\{1,2,3,4,5\}$，那么可从评分矩阵中得到 $p(x)$ 和 $q(x)$。假设 i 和 j 是两个项目，那么所有用户对项目 i 和项目 j 评分的密度分布为 p_i 和 q_j，则项目 i 和项目 j 的相似性计算如式(8)所示：

$$BC(i,j)=BC(p_i,q_j)=\sum_{r=1}^{n}\sqrt{p_{ir}\cdot q_{jr}} \tag{8}$$

其中，n 为评分的最大值，p_{ir} 和 q_{jr} 分别表示用户对项目 i 和项目 j 评分为 r 的概率。p_{ir} 的计算公式如式(9)所示：

$$pir=\frac{r_{num}}{i_{all}} \tag{9}$$

其中，r_{num} 表示对项目 i 评分为 r 的用户数，i_{all} 表示对项目 i 评分的所有用户数。

本文采用如下例子来详细说明 Bhattacharyya 方法计算项目之间相似性的过程。假设用户对项目未评分表示为 0，从表 1 中选择两组数据，所有用户对项目 i_1 的评分向量记作 I_1，则 $I_1=\{4,0,0,5,0,1\}^{\mathrm{T}}$；对项目 i_4 的评分向量记作 I_4，则 $I_4=\{0,2,1,0,4,0\}^{\mathrm{T}}$。项目 i_1 和 i_4 的相似度计算如下：

$$\begin{aligned}BC(i_1,i_4)&=\sum_{r=1}^{5}\sqrt{p_{(i_1)r}\cdot q_{(i_4)r}}\\&=\sqrt{\frac{1}{3}*\frac{1}{3}}+\sqrt{\frac{0}{3}*\frac{1}{3}}+\sqrt{\frac{0}{3}*\frac{0}{3}}+\\&\quad\sqrt{\frac{1}{3}*\frac{1}{3}}+\sqrt{\frac{1}{3}*\frac{0}{3}}\\&=\frac{2}{3}\end{aligned} \tag{10}$$

其他项目的相似度计算过程与此类似。由上述的计算过程可知，即使没有不同用户对同一个项目评分，也可以通过 Bhattacharyya 系数获得项目之间的相似度。

2.2 新相似度模型

本节将给出所提新方法的数学表达式。在传统的相似度计算中考虑的是共同评分的数量，共同评分的项目信息越多，

得到的相似度就越大。然而在真实情况下，我们获得的数据集都较稀疏，有较多用户或项目无共同的评分，因此要充分利用数据集提供的信息来考虑影响相似度计算的各种因素，以使目标用户可以获得更加准确的邻域，并得到更好的推荐。本文从 3 个方面考虑用户的相似度：1)用户评过分的项目之间的关联性；2)用户的评分行为；3)用户之间不同的评分偏好。

2.2.1　用户评分项目之间的关联性

现实生活中，可通过一个人拥有的物品来推断其喜好，因此若两个人拥有较多相同或相似的物品，则在不考虑其他因素的情况下，可以根据两人拥有物品之间的相关性来判断两人的相似度。据此，可以通过用户评过分的项目之间的关联性来反映每个用户之间的相似性，这样就利用了用户的所有评分信息。假设用户 a 和 b 评过分的项目集合为 I_a 和 I_b，则用户 a 与 b 之间的相似度可采用物品之间的相关性来描述，计算公式如下：

$$sim(a,b)^{BC}=\sum_{i\in I_a}\sum_{j\in I_b}BC(i,j) \tag{11}$$

2.2.2　用户的评分行为

判断用户喜好的最好方法是观察用户的行为表现，这也是一种最直观的了解方式。传统的协同过滤推荐算法通过用户对项目的评分行为来计算用户之间的相似度，因此用户的评分行为可以很好地刻画用户对项目的喜好程度；判断两个用户的相似性时，若两个用户共同评分的项目越多，则他们之间的潜在相关性可能就越大，但这并非是绝对的。鉴于实际环境中用户评分数据较稀疏，不同用户对相同项目的共同评分较少，我们对无共同评分的用户给予一定的惩罚。根据用户的评分行为刻画的用户相似度如式(12)所示：

$$sim(a,b)^{Action}=\begin{cases}pcc(a,b), & I_C>0\\ \dfrac{\sum\limits_{i\in I_a}\sum\limits_{j\in I_b}(r_{a,i}-\bar{r}_a)\cdot(r_{b,i}-\bar{r}_b)}{\sqrt{\sum\limits_{i\in I_a}(r_{a,i}-\bar{r}_a)^2}\cdot\sqrt{\sum\limits_{i\in I_b}(r_{b,i}-\bar{r}_b)^2}}, & \text{otherwise}\end{cases} \tag{12}$$

其中，I_c 表示用户之间的共同评分项目数。

2.2.3　用户的评分偏好

由于每个用户的偏好不同，因此评分标准也不同。若用户对项目质量要求较高，则评分相对较低；若用户对项目质量要求较低，则评分可能相对较高。本文使用评分的均值和平方差对用户的行为偏好进行建模，如式(13)所示：

$$URP(a,b)=1-\frac{1}{1+\exp(-|\mu_a-\mu_b|\cdot|\sigma_a-\sigma_b|)} \tag{13}$$

其中，$\mu_a=\sum\limits_{i\in I_a}r_{a,i}/|I_a|$，$\sigma_a=\sqrt{\sum\limits_{i\in I_a}(r_{a,i}-\bar{r}_a)^2/|I_a|}$。

2.2.4　用户相似度计算

由上述分析可知，考虑的因素越多，对用户的兴趣特征预测得越准确。通过研究发现，融合以上 3 个因素能得到更加准确的结果，获得比较合适的用户邻域，有利于改善系统的推荐效果。最终的用户相似度计算公式如式(14)所示：

$$sim(a,b)^{finalSim}=sim(a,b)^{BC}\cdot sim(a,b)^{Action}\cdot URP(a,b) \tag{14}$$

由式(14)可知，若用户评分的物品之间越相似，则对用户之间的相似度影响就越深刻。

3　实验结果及分析

3.1　数据集

本文实验使用了两个通用的数据集：MovieLens-100k 和 FilmTrust。MovieLens-100k 数据集是 GroupLens 提供的一个电影评分数据集，其中包含 943 个用户对 1682 部电影的 100000 个评分，每部电影的评分范围为 1～5，评分矩阵的密度为 6.3%。FilmTrust 数据集包含 1058 个用户对 2071 部电影的 35497 个评分，评分范围为 0.5～4.0，每个评分是 0.5 的整数倍。由上述数据可知，两个数据集都非常稀疏。

本文使用这两个数据集，是因为它们在协同过滤推荐算法研究领域中被许多研究者广泛使用，具有很好的通用性。实验将数据集分为训练集和测试集，其中 80%作为训练数据集，剩余的作为测试集。此外，为了更好地说明各种方法的效果，在实验过程中采用不同的分割比例对数据进行了交叉验证，最后给出了结果的平均值。

3.2　评价指标

在推荐系统中，需要各种评价指标来比较每种方法的推荐性能，不同的评价方法反映推荐算法性能的不同方面。本文选择两种通用的评价指标，分别是平均绝对误差(MAE)和均方根误差(RMSE)，如式(15)和式(16)所示：

$$MAE=\sum_{i=1}^{n}\frac{|r_{u,i}-\hat{r}_{u,i}|}{n} \tag{15}$$

$$RMSE=\sqrt{\frac{\sum\limits_{i=1}^{n}(r_{u,i}-\hat{r}_{u,i})^2}{n}} \tag{16}$$

其中，n 表示测试集的评分数，$\hat{r}_{u,i}$ 表示预测值。若 *MAE* 或 *RMSE* 的结果值越小，则推荐效果越好，方法越精确。

3.3　比较方法

为了说明本文所提方法的有效性，将其与当前常用的以及最好水平的相似度方法进行比较，如 pcc，cpcc，cos，msd 和 exJaccard[33]，它们都被广泛应用于协同过滤推荐算法的相似度计算。

3.4　性能比较

本节将在两个数据集上通过对比实验来验证本文方法的有效性。在实验过程中发现，各种方法在不同的数据集上获得的推荐效果不同，用户的邻域选择也会对推荐的性能产生较大的影响，因此为了充分显示各种方法的推荐性能，本文选择不同的邻域值进行实验。假设 k 表示最近邻数量，不同的值将会获得不同的推荐性能。

MovieLens-100k 数据集上不同方法的 *MAE* 比较和 *RMSE* 比较分别如图 1 和图 2 所示。

由图 1 可知，在 MovieLens-100k 数据集上所有方法的推荐性能在整体上随着 k 值的增大而变得更优，即 *MAE* 不断变小；同时，在任何 k 值情况下，本文提出的新方法的推荐性能一直优于其他方法，当 $k>70$ 时，新方法的 *MAE* 趋于稳定。

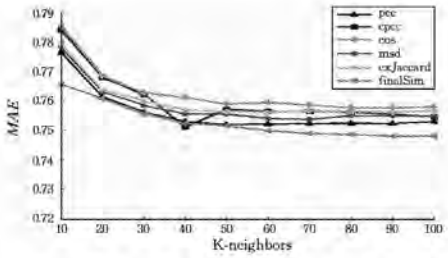

图 1 MovieLens-100k 数据集上不同方法的 *MAE* 比较

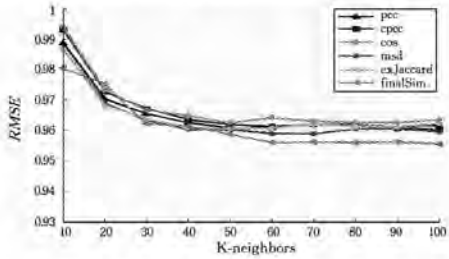

图 2 MovieLens-100k 数据集上不同方法的 *RMSE* 比较

由图 2 可知，在 MovieLens-100k 数据集上当 $k>40$ 时，本文提出的方法比其他方法得到了更小的 *RMSE*，推荐性能更好；当 $k>70$ 时新方法得到的 *RMSE* 趋于稳定。

图 3 给出了在 FilmTrust 数据集上不同方法的 *MAE* 指标的比较结果。由图 3 可知，所提出的新方法在 $k<50$ 时的推荐性能并未提高太多，CPCC 的性能更好；而当 $k>50$ 时，新方法的推荐性能明显优于 CPCC 方法，提高了推荐的准确度。

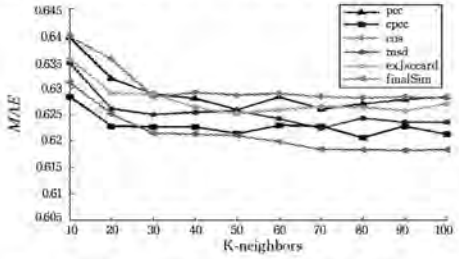

图 3 FilmTrust 数据集上不同方法的 *MAE* 比较

图 4 给出了在 FilmTrust 数据集上不同方法的 *RMSE* 指标的比较结果。由图 4 可知，当 $k<30$ 时，CPCC 的 *RMSE* 值一直保持最小，随着 k 值的增大，本文方法的推荐性能逐渐提升，在 $k=70$ 时达到最优，但 $k>70$ 后 *RMSE* 有小幅度的增加。

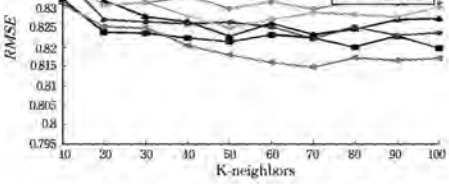

图 4 FilmTrust 数据集上不同方法的 *RMSE* 比较

由上述结果可知，当 $k>50$ 时，本文提出的方法较其他方法可以得到更好的推荐性能。在上述两个数据集上的实验结果充分验证了本文方法的有效性。

结束语 本文基于多方面因素计算用户之间的相似度，除了考虑用户的评分行为，即用户共同评分的项目，还考虑了用户评过分的项目之间的相关性，以及用户之间不同的评分偏好。现实世界中的评分数据集都较稀疏，通过充分挖掘数据中的潜在信息，完全可以建立更加精确的预测模型。本文发现用户的评分偏好也是一个重要的影响因素，因此将该因素融入模型中提高了推荐的精确度，并通过在两个常用数据集上的实验得到了验证。

用户或物品的属性信息都可以被充分利用，以提高计算相似度的准确度。随着社交网络的快速发展，用户产生的社交信息可以弥补评分数据稀疏导致的缺陷，因为朋友之间的兴趣通常比较相近，所以对相似度的计算将产生较大的影响。未来将会关注数据集中的异构信息和社交信息。

参 考 文 献

[1] CECHINEL C, SICILIA M Á, SÁNCHEZ-ALONSO S, et al. Evaluating collaborative filtering recommendations inside large learning object repositories[J]. Inf. Process. Manag., 2013, 49(1): 34-50.

[2] PAZZANI M J, BILLSUS D. Content-based recommendation systems[C]// The Adap. Web.. 2007: 325-341.

[3] BREESE J S, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. 1998: 43-52

[4] BURKE R. Integrating knowledge-based and collaborative-filtering recom-mender systems[C]// Proc. Work. AI Electron. Commer.. 1999: 69-72.

[5] LIU Z, QU W, LI H, et al. A hybrid collaborative filtering recommen-dation mechanism for P2P networks[J]. Future Generation Computer Systems, 2010, 26(8): 1409-1417.

[6] BREESE J S, HECKERMAN D, KADIE C. Emprical Analysis of Predictive algorithm for Collaborative filtering[J]. Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, 7(7): 43-52.

[7] DENG A L, ZHU Y Y, SHI B L. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Joumal of Software, 2003, 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.

[8] GOLDBERG K, ROEDER T, GUPTA G, et al. Eigentaste: a constant time collaborative filtering algorithm[J]. Information Retrieval, 2001, 4(2): 133-151.

[9] SARWAR B M, KARPIS G, KONSTAN J A, et al. Application of dimen-sionality reduction in recommender system-a case study [M]// ACM WebKDD Workshop. 2000

[10] BEDI P, KAUR H, MARWAHA S. Trust based recommender system for semantic Web[C]// Proc. of IJCAI'07. 2007: 2677-2682.

[11] MA H, KING I, LYU M R. Learning torecommend with social trust ensemble[C]// Proc. of SIGIR'09. Boston, MA, USA, 2009: 203-210.

[12] MA H, YANG H, LYU M R, et al. SoRec: Social recommendation using probabilistic matrix factorization[C]// Proceedings of

CIKM '08. Napa Valley, USA, 2008: 931-940.

[13] MASSA P, AVESANI P. Trust-aware recommender systems [C]// Proc. of RecSys '07. Minneapolis, MN, USA, 2007: 17-24.

[14] RESNICK P, IACOVOU N, SUCHAK M, et al. GroupLens: an open architecture for collaborative filtering of netnews [C]// Proceeding of the ACM Conference on Computer Supported Cooperative Work. 1994: 175-186

[15] SHI Y, LARSON M, HANJALIC A. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges[J]. ACM Computing Surveys (CSUR), 2014, 47(1): 1-45

[16] CACHEDA F, CARNEIRO V, FERNÁNDEZ D, et al. Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender system[J]. ACM Transactions Web, 2011, 5(1): 1-33.

[17] HUANG C G, YIN J, WANG J, et al. Uncertain neighbors' collaborative filtering recommendation algorithm[J]. Chinese Journal of Computers, 2010, 33(87): 1369-1377. (in Chinese)
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(87)：1369-1377.

[18] LUO X, OUYANG Y X, XIONG Z, et al. The effect of similarity support in K-nearest-neighborhood based collaborative filtering [J]. Chinese Journal of Computers, 2010, 33(8): 1437-1455. (in Chinese)
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)：1437-1455.

[19] XING C X, GAO F R, ZHAN S A, et al. A collaborative filtering recommendation algorithm in corporate with user interest change [J]. Journal of Computer Research and Development, 2007, 44(2): 296-301. (in Chinese)
邢春晓，高凤荣，战思南，等. 适应用户兴趣变化的协同过滤推荐算法[J]. 计算机研究与发展，2007，44(2)：296-301.

[20] STREHL A, GHOSH J, MOONEY R. Impact of similarity measures on web-page clustering[C]// Proceedings of the International Workshop on Artificial Intelligence for Web Search. 2000: 58-64.

[21] HERLOCKER J L, KONSTAN J A, BORCHERS A, et al. An algorithmic framework for performing collaborative filtering[C]// Proceedings of the Twenty Second Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 1999: 230-237.

[22] JAMALI M, ESTER M. Trustwalker: A random walk model for combining trust-based and item-based recommendation [C]// Proceedings of the fifteenth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2009: 397-406

[23] SHARDANAND U, MAES P. Social information filtering: algorithms for automating word of mouth[C]// Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1994: 210-217.

[24] POLATIDS N, GEORGIADIS C K. A multi-level collaborative filtering method that improves recommendations [J]. Expert Systems with Applications, 2016, 48: 100-110.

[25] LU J, SHAMBOUR Q, XU Y, et al. A web-based personalized business partner recommendation system using fuzzy semantic techniques[J]. Computational Intelligence, 2013, 29(1): 37-69.

[26] LUO H, NIU C, SHEN R, et al. A collaborative filtering framework based on both local user similarity and global user similarity[J]. Mach. Learn., 2008, 72(3): 231-245.

[27] LIU H, HU Z, MIAN A, et al. A new user similarity model to improve the accuracy of collaborative filtering[J]. Knowledge-Based Systems, 2014, 56(3): 156-166.

[28] WANG W, ZHANG G, LU J. Collaborative filtering with entropy-driven user similarity in recommender systems[J]. International Journal of Intelligent Systems, 2015, 30(8): 854-870.

[29] BHATTACHARYYA A. On a measure of divergence between two statistical populations defined by their probability distributions[J]. Bull. Calcutta Math. Soc., 1943, 35(1): 99-109.

[30] KAILATH T. The divergence and Bhattacharyya distance measures in signal selection[J]. IEEE Transactions Commun. Technol., 1967, 15(1): 52-60.

[31] NIELSEN F, BOLTZ S. The Burbea-Rao and Bhattacharyya centroids[J]. IEEE Transactions Inf. Theory, 2011, 57(8): 5455-5466.

[32] AHERNE F J, THACKER N A, ROCKETT P. The Bhattacharyya metric as an absolute similarity measure for frequency coded data[J]. Kybernetika, 1998, 34(4): 363-368.

[33] HUANG A. Similarity measures for text document clustering [C]// Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008). Christchurch, New Zealand, 2008: 49-56.

(上接第 41 页)

[7] DWORK. Differential Privacy[C]// Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP). 2016: 1-12.

[8] MCSHERRY F, MIRONOV I. Differentially private recommender systems: building privacy into the net[C]// ACM Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM, 2009: 627-636.

[9] ZHU T, LI G, REN Y, et al. Differential privacy for neighborhood-based Collaborative Filtering [C]// IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. IEEE, 2013: 752-759.

[10] YE M, YIN P, LEE W C. Location recommendation for location-based social networks[C]// ACM Sigspatial International Symposium on Advances in Geographic Information Systems, ACM-GIS 2010. USA, 2010: 458-461.

[11] WANG H, TERROVITIS M, MAMOULIS N. Location Recommendation in Location-based Social Networks using User Check-in Data[C]// ACM Sigspatial International Conference on Advances in Geographic Information Systems. 2013: 374-383.

[12] CHO E, MYERS S A, LESKOVEC J. Friendship and mobility: user movement in location-based social networks [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, CA, USA, 2011: 1082-1090.