文章编号:1007-5321(2017)02-0110-05 DOI:10.13190/j.jbupt.2017.02.019

基于 KL 散度的用户相似性协同过滤算法

王 永, 邓江洲
(重庆邮电大学 电子商务与现代物流重点实验室, 重庆 400065)

摘要: 大多数用户相似性算法在计算用户相似性时只考虑了用户间的共同评分项,而忽略了用户其他评分中可能隐藏的有价值信息. 为了准确评估用户间的相似性,提出了一种基于 KL 散度的用户相似性协同过滤算法. 该算法不仅利用了共同评分项,还考虑了其他非共同评分信息的影响. 该算法充分利用了用户的所有评分信息,提高了用户相似性度量的可靠性和准确性. 实验结果表明,该算法优于当前主流的用户相似性算法,且在没有共同评分信息的条件下,仍能有效地完成用户相似性度量,解决了对共同评分项的完全依赖问题,具有更好的适应性.

关 键 词: 协同过滤算法; 用户相似性; KL 散度; 共同评分信息; 数据稀疏
中图分类号: TP391 **文献标志码**: A

User Similarity Collaborative Filtering Algorithm Based on KL Divergence

WANG Yong, DENG Jiang-zhou
(Key Laboratory of Electronic Commerce and Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: User similarity based collaborative filtering algorithm is one of most widely used technologies. Most of user similarity algorithms only consider the co-rated items between two users, but ignore other ratings that probably hide valuable information. To evaluate user similarity accurately, a user similarity collaborative filtering algorithm based on Kullback - Leibles (KL) divergence was proposed. The proposed algorithm utilizes both the co-rated items and the influence of other no co-rated items. Since the algorithm makes full use of all rating information, it improves the accuracy and reliability of user similarity. Experiments show that the proposed algorithm outperforms other user similarities. Moreover, it can still measure the user similarity effectively, even if no co-rated items exist. Therefore, the presented algorithm solves the problem of full dependence on co-rated items and gains better flexibility.

Key words: collaborative filtering algorithm; user similarity; Kullback-Leibles divergence; co-rated information; data sparseness

协同过滤算法利用用户已有的偏好去预测未来对某产品的偏好,以满足用户的个性化需求. 数据稀疏性问题是该领域的常见问题之一,为解决此问题,许多相似性度量方法被提出[1-3]. 然而,这些算法普遍存在的问题是,只使用了共同评分信息,忽略了其他非共同评分信息的影响. 笔者借助 KL(Kullback-Leibles)散度提出了一种新的用户相似性协同过滤算法. 该算法既能有效处理共同评分项中的信息,还能充分利用其他的非共同评分项信息,具有更好的客观性与全面性,且能有效地应对数据集稀疏

收稿日期: 2016-11-16
基金项目: 国家自然科学基金项目(61472464); 国家社会科学基金项目(14CTQ026); 重庆市自然科学基金项目(cstc2015jcyjA10081)
作者简介: 王 永(1977—), 男, 教授, E-mail:wangyong_cqupt@163.com.

的问题，具有很好的应用价值.

1　预备知识

协同过滤推荐算法通常是利用已有的评分数据来预测用户对未评价项目的偏好程度. 已有的评分数据可表示为表 1 所示的 User-Item 评分矩阵，其中用户集合为 $U=\{u_1,u_2,\cdots,u_m\}$，项目计划为 $I=\{i_1,i_2,\cdots,i_n\}$，r_{uj}为用户 u 对项目 j 的评分.

表 1　User-Item 评分矩阵

U	i_1	i_2	…	i_n
u_1	r_{11}	r_{12}	…	–
u_2	–	r_{22}	…	r_{2n}
…	…	…	…	…
u_m	r_{m1}	r_{m2}	…	r_{mn}

值得注意的是，User-Item 评分矩阵通常是一个非常稀疏的矩阵. 如何设计相似性计算方法以解决评分矩阵的稀疏性问题，是该领域的研究热点之一.

1.1　用户相似性模型

用户相似性计算是协同过滤推荐算法中极其重要的部分，对推荐结果有至关重要的影响. 常见的具有代表性的用户相似性计算模型包括修正余弦相似性（ACOS，adjusted cosine similarity）、皮尔逊相关系数（PC，Pearson correlation coefficient）和约束皮尔逊相关系数（CPC，constrained Pearson's correlation）等，对应的计算式如下[1].

ACOS：

$$\mathrm{sim}(u,v)_{\mathrm{ACOS}}=\frac{\sum_{i\in I_u\cap I_v}(r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I_u}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_v}(r_{vi}-\bar{r}_v)^2}} \tag{1}$$

PC：

$$\mathrm{sim}(u,v)_{\mathrm{PC}}=\frac{\sum_{i\in I_u\cap I_v}(r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_u\cap I_v}(r_{vi}-\bar{r}_v)^2}} \tag{2}$$

CPC：

$$\mathrm{sim}(u,v)_{\mathrm{CPC}}=\frac{\sum_{i\in I_u\cap I_v}(r_{ui}-\bar{r}_{\mathrm{med}})(r_{vi}-\bar{r}_{\mathrm{med}})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{ui}-\bar{r}_{\mathrm{med}})^2}\sqrt{\sum_{i\in I_u\cap I_v}(r_{vi}-\bar{r}_{\mathrm{med}})^2}} \tag{3}$$

其中：I_u为用户 u 已评分的项目集合，$\bar{r}_u$ 为用户 u 对所有已评项目的评分均值，$\bar{r}_{\mathrm{med}}$为评分区间中值.

上述用户相似性方法在计算过程中只考虑了用户的共同评分项，而忽略了非共同评分的价值. 当共同评分项少或不存在时，这些方法将会出现较大的偏差或不可用，从而影响最后的推荐效果.

1.2　最近邻居集

最近邻居集 N_u是针对目标用户 u，计算用户 u 与其他用户间的相似性值，然后按照相似性值从大到小的筛选规则，选取出前 n 个用户. 这 n 个用户组成的集合就是用户 u 的最近邻居集.

1.3　评分预测规则

根据用户 u 的最近邻居集合 N_u，可得用户 u 对未评分项目 i 的预测评分 p_{ui}，其计算式为[1]

$$p_{ui}=\bar{r}_u+\frac{\sum_{v\in N_u}\mathrm{sim}(u,v)(r_{vi}-\bar{r}_v)}{\sum_{v\in N_u}|\ \mathrm{sim}(u,v)\ |} \tag{4}$$

2　基于 KL 散度的用户相似性协同过滤算法

2.1　用户相似性计算模型

传统方法在计算 2 用户间相似性时，通常只考虑了 2 用户的共同评分项. 在实际应用中，由于数据集的稀疏性会导致共同评分项十分稀少，从而造成用户间相似性测量的误差偏大，影响推荐效果. 笔者希望在计算用户相似性时能充分利用 2 个用户的所有评分信息，而不只是共同评分项. 为此，提出如下的用户相似性计算模型.

设用户 u,v 对项目 i,j 的评分分别为（r_{ui}，–）和（–，r_{vj}）. 由于用户间没有共同评分，采用 1.1 节中的公式无法计算用户 u 和 v 的相似性. 这样的情况在评分矩阵中广泛存在. 如果能利用非共同评分计算出用户的相似性，将极大地提升用户相似性计算式的应用范围. 通常用户 u 和 v 会对多个项目进行评分，因而在计算用户相似性时应综合考虑所有评分组合. 为此，引入的计算公式为

$$\mathrm{sim}(u,v)=\sum_{i\in I_u}\sum_{j\in I_v}\frac{(r_{ui}-\bar{r}_u)(r_{vj}-\bar{r}_v)}{\sigma_u\sigma_v} \tag{5}$$

其中：I_u为用户 u 的所有评分项集合；$\bar{r}_u$ 和 σ_u 分别为用户 u 的平均评分和标准差. 设 $s=\frac{(r_{ui}-\bar{r}_u)(r_{vj}-\bar{r}_v)}{\sigma_u\sigma_v}$为仅依靠 1 个评分对得到的用户相似性. s 的计算存在一个明显的局限：没有考虑项目 i 和 j 间的差异. 为了克服此局限性，笔者引入项

目相似性 $\mathrm{sim}(i,j)_{\mathrm{item}}$ 作为其权重对 s 的计算结果进行修正,则得到了如下的用户相似性计算模型.

$$\mathrm{sim}(u,v)_{\mathrm{final}} = \sum_{i\in I_u}\sum_{j\in I_v} \mathrm{sim}(i,j)_{\mathrm{item}} s \qquad (6)$$

由式(6)知,当 s 值较大,且项目 i 和 j 相似,则 $\mathrm{sim}(i,j)_{\mathrm{item}}s$ 值较大,计算的结果是增强用户 u 和 v 的相似性;相反,当 s 值较大,且项目 i 和 j 不相似时,则 $\mathrm{sim}(i,j)_{\mathrm{item}}$ 的值较小,导致 $\mathrm{sim}(i,j)_{\mathrm{item}}s$ 的值也较小,从而削弱用户 u 和 v 的相似性. 当 s 值较小时,可得类似的结果. 式(6)是所有可能的评分对的加权和,是一个累积的统计结果,不容易受局部评分的影响,因此具有更好的客观性和全面性.

此外,由式(6)可知,$\mathrm{sim}(i,j)_{\mathrm{item}}$ 对计算结果有重要影响. 在设计 $\mathrm{sim}(i,j)_{\mathrm{item}}$ 的计算式时希望能满足如下要求:1)不受共同评分项的限制,具有良好的适应性;2)可利用所有的评分信息,具有良好的全面性. 笔者采用基于 KL 散度的项目相似性来计算 $\mathrm{sim}(i,j)_{\mathrm{item}}$,可满足上述要求.

2.2 基于 KL 散度的项目相似性

1)KL 散度

KL 散度又称 KL 距离,从概率分布的角度去衡量 2 个变量间的距离[4-5]. 假设 ρ_1 和 ρ_2 分别为 2 个不同的概率密度函数,对于离散数据集 D,KL 距离定义为[4]

$$D(\rho_1 \| \rho_2) = \sum_{x\in D} \rho_1(x) \mathrm{lb}\frac{\rho_1(x)}{\rho_2(x)} \qquad (7)$$

其中 $\rho_1(x)>0$ 和 $\rho_2(x)>0$.

2)项目相似性计算

基于上述定义,在 Uses-Item 评分矩阵中,对任意 2 项目 i 和 j,将所有用户对它们的评分视作 2 个变量序列,可得项目 i 与 j 的 KL 距离 $D(i,j)$ 的计算式为[2]

$$D(i,j) = D(\rho_i \| \rho_j) = \sum_{r=1}^{r_{\max}} \rho_{ir} \mathrm{lb}\frac{\rho_{ir}}{\rho_{jr}} \qquad (8)$$

其中:ρ_i 为项目 i 的密度函数,$r_{\max}$ 为评分区间的最大值;$\rho_{ir} = \frac{\#r}{\#i}$ 为项目 i 中评分值为 r 的比率,$\#i$ 为所有用户对项目 i 评分的个数,$\#r$ 为项目 i 中值为 r 的评分个数.

根据 KL 距离,可得基于 KL 的全局项目相似性计算公式为[2]

$$L(i,j) = \mathrm{sim}(i,j)_{\mathrm{item}} = \frac{1}{1+D(i,j)} \qquad (9)$$

由于 KL 距离不具有对称性,而度量 2 项目间的距离时需要具有对称性,笔者采用式(10)代替式(9)中的 $D(i,j)$.

$$D_s(i,j) = (D(i,j) + D(j,i))/2 \qquad (10)$$

2.3 用户相似性计算

将式(6)中的项目相似性 $\mathrm{sim}(i,j)_{\mathrm{item}}$ 替换为式(9),可得最终的基于 KL 散度的用户相似性计算公式为

$$\mathrm{sim}(u,v)_{\mathrm{KLCF}} = \sum_{i\in I_u}\sum_{j\in I_v} L(i,j)\frac{(r_{ui}-\bar{r}_u)(r_{vj}-\bar{r}_v)}{\sigma_u \sigma_v} \qquad (11)$$

2.4 算法描述

利用 2.1~2.3 节中用户相似性计算的关键步骤和 1.3 节中的评分预测规则,得到基于 KL 散度的用户相似性协同过滤算法(KLCF, Kullback-Leibles collaborative filtering)如下.

输入:Uses-Item 评分矩阵.

输出:用户 u 对未评分项目的预测评分值.

① 利用式(8)和式(10)得到项目 i 和 j 的 KL 距离 $D_s(i,j)$;

② 利用式(9)计算项目 i 和 j 的 KL 相似性 $L(i,j)$;

③ 重复步骤①②,计算所有项目间的相似性;

④ 利用式(11)计算用户 u 和其他用户间的相似性,并筛选出前 n 个相似性值最高的用户作为用户 u 的最近邻居集合;

⑤ 利用式(4)得到最终的预测评分值.

3 算法分析

1)充分利用所有评分信息

传统的相似性度量方法,如余弦相似性等,是完全依赖于共同评分项的. 若用户间不存在共同评分项,则这类方法将无法进行相似性计算. 在稀疏数据集中,用户通常只对万千项目中的一小部分评分,所以用户间的共同评分项极度稀少. 例如 MovieLens 数据集中,2 用户间的共同评分数量平均仅占他们总评分数量的 4%. 若仅依靠共同评分项,则可利用的信息很少,容易造成较大的误差. 笔者的方法同时利用了共同评分信息和其他的非共同评分信息,解决了上述不足,且使算法具有更好的适应性. 由于笔者方法更充分的利用了所有评分项信息,因此提高了计算的准确性.

2)充分利用了 KL 散度优势

式(11)中项目相似性是基于 KL 散度设计的.

将评分矩阵中 2 个项目的所有评分视作 2 条序列时，由于评分范围有限（常为 5 级或 10 级），所以 2 条序列之间必然存在大量重叠点. KL 散度是利用评分值的概率分布去度量项目间的相似程度. 已有研究[5]表明，KL 散度的优势在于它能高效区分欧式距离难以区别的对象，尤其是当 2 个数据集中的数据出现大量重叠时，优势更为明显. 基于 KL 散度的项目相似性从统计角度使用了所有的评分信息，结果不容易受特殊情况影响，具有更好的客观性. 此外，该方法对用户所评项目的数量没有要求，即使在极为稀疏的数据集中也能很好的测量项目的相似性，具有很好的适应性.

3）利用绝对评分值区别不同用户

Jaccard 系数相似性方法[1]虽然也利用了用户的所有评分值，但它只考虑了共同评分项目的比例，而忽略了绝对评分值对计算结果的影响. 这导致 Jaccard 方法难以区分不同的用户，该结论在文献[1]中已有验证. 笔者的方法同时考虑了用户评分绝对值和它在项目所有评分中的比例，所以可有效克服 Jaccard 方法存在的不足，更好地区分用户.

4　实验结果与分析

4.1　数据集

采用公开数据集 MovieLens① 中最新公布的 ML-Lastest-Small 和 Yahoo Music② 作为笔者算法测试和验证的数据集. 各数据集的描述如表 2 所示，其中稀疏度 $\kappa = \frac{R}{MN} \times 100\%$.

表 2　实验数据集相关信息描述

数据集	用户数 M	项目数 N	评分数 R	稀疏度/%
MovieLens	706	8 570	100 023	1.7
Yahoo Music	15 400	1 000	365 704	2.3

为测试算法，将每个数据集划分为 2 部分，随机筛选出的 80% 数据作为训练集，余下的 20% 作为测试集.

4.2　评价指标

预测准确性是衡量协同过滤算法性能的一个重要参考标准，常用平均绝对误差（MAE，mean absolute error）来进行度量，计算式为

$$E = \frac{1}{m}\sum_{u=1}^{m}\frac{1}{n}\sum_{i=1}^{n}|r_{ui} - p_{ui}| \tag{12}$$

其中：m 为目标用户的个数，n 为用户 u 预测的项目数；r_{ui} 和 p_{ui} 分别为用户 u 对项目 i 的实际评分值和预测评分值.

推荐准确性是衡量推荐算法性能的另一重要参考标准，常用准确率 P、召回率 C 和 F_1 值来进行评估，对应计算式为[6]

$$P = \frac{1}{m}\sum_{u=1}^{m}\frac{|R_{u,\mathrm{p}} \cap R_{u,\mathrm{a}}|}{|R_{u,\mathrm{p}}|} \tag{13}$$

$$C = \frac{1}{m}\sum_{u=1}^{m}\frac{|R_{u,\mathrm{p}} \cap R_{u,\mathrm{a}}|}{|R_{u,\mathrm{a}}|} \tag{14}$$

$$F_1 = \frac{2PC}{P+C} \tag{15}$$

其中 $R_{u,\mathrm{p}}$ 和 $R_{u,\mathrm{a}}$ 分别为预测推荐的项目集合和真实推荐的项目集合. F_1 值为 P 和 C 的综合评价指标，F_1 值越大，推荐效果越好.

4.3　结果分析

为了与笔者方法进行对比，在相同的数据集中，对协同过滤方法皮尔森相关系数（PCC，Pearson correlation coefficient）[1]，约束的皮尔森相关系数（CPCC，constrained Pearson correlation coefficient）[1]，平均平方差（MSD，mean squared difference）[3]，接近－影响－普及（PIP，proximity-impact-popularity）[1]，新的启发式相似性模型（NHSM，new heuristic similarity model）[1]和 Bhattacharyya 系数协同过滤（BCF，Bhattacharyya coefficient in collaborative filtering）[3]进行了测试. 最终得到的实验结果如图 1、图 2 所示.

4.3.1　MAE 测试结果及比较

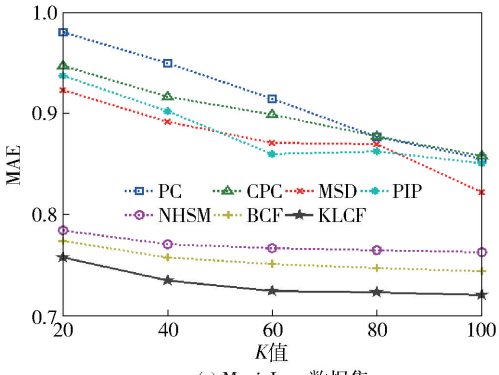
MAE 反映的是预测评分值与用户实际评分值之间的平均差异程度. 图 1（a）中，笔者提出 KLCF 算法的 MAE 值总体范围为 $0.718 \leqslant E \leqslant 0.745$，对于最接近的 BCF 算法，精确度提高了约 2%；图 1（b）中，当 K 值为 100 时，KLCF 的 MAE 值最小，为 0.986. 图 1 整体反映出 KLCF 算法的 MAE 值最小，具有良好的预测精度，且明显优于其他算法.

4.3.2　F_1 值比较

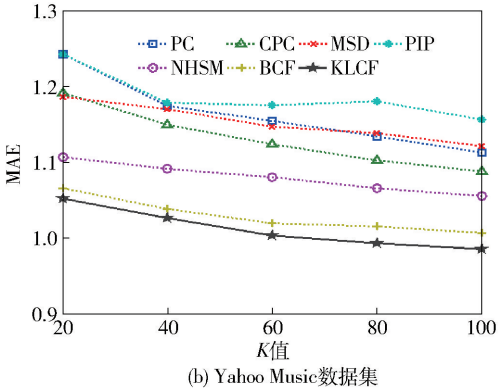
在此处的实验中，将预测值大于用户平均评分的项目作为推荐项目，并以此为基础计算 F_1 值. 图 2（a）中，KLCF 算法的 F_1 值比 BCF 算法提高了 6.6%；图 2（b）中，当 K 值大于 40 时，只有 KLCF 的

① http://www.grouplens.org

② http://research.yahoo.com/Academic_Relations

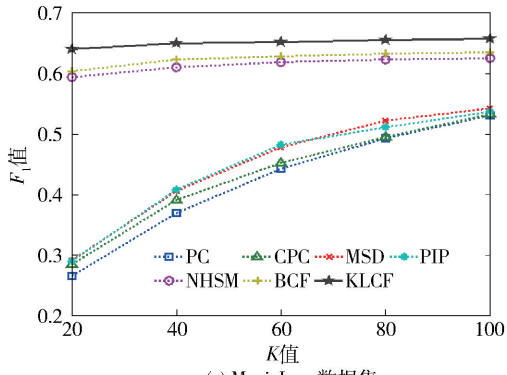


(a) MovieLens数据集

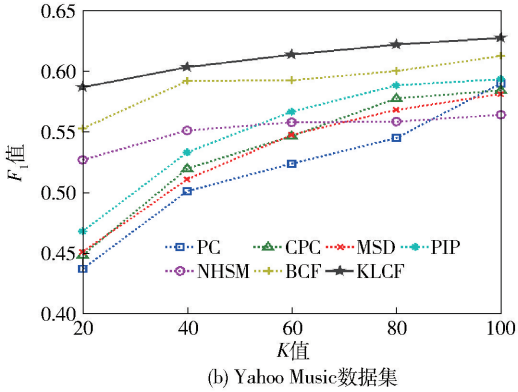


(b) Yahoo Music数据集

图1 MAE测试结果



(a) MovieLens数据集



(b) Yahoo Music数据集

图2 F_1值结果比较

F_1值超过0.6. 图2中的结果表明,KLCF算法在2个数据集上均有良好的推荐效果,且优于其他算法.

5 结束语

为解决用户相似性计算依赖于共同评分项的问题,提出了一种基于KL散度的用户相似性协同过滤算法. 该算法同时利用了共同评分信息和非共同评分信息,克服了已有方法受制于共同评分项的限制,对稀疏数据集具有更好的适应性. 在计算项目相似性时,基于KL散度从所有评分信息的概率分布角度进行测量,由于充分利用了所有的信息,故增强了算法的可靠性和准确性. 数据实验及其对比的结果表明,该算法具有良好的性能,且优于已有的常用协同过滤算法,具有很好的应用潜力.

参考文献:

[1] Liu Haifeng, Hu Zheng, Mian Ahmad, et al. A new user similarity model to improve the accuracy of collaborative filtering[J]. Knowledge Based Systems, 2014, 56(11): 156-166.

[2] 王永, 邓江洲, 邓永恒, 等. 基于项目概率分布的协同过滤推荐算法[J]. 现代图书情报技术, 2016, 32(6): 73-79.
Wang Yong, Deng Jiangzhou, Deng Yongheng, et al. A collaborative filtering recommendation algorithm based on item probability distribution[J]. New Technology of Library and Information Service, 2016, 32(6): 73-79.

[3] Patra B K, Launonen R, Ollikainen V, et al. A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data[J]. Knowledge-Based Systems, 2015, 82(3): 163-177.

[4] Kullback S, Leibler R A. On information and sufficiency [J]. The Annals of Mathematical Statistics, 1951(22): 79-86.

[5] Huang Anna. Similarity measures for text document clustering[C]//the New Zealand Computer Science Research Student Conference(NZCSRSC). New Zealand: [s. n.], 2008: 49-56.

[6] Zhang Jing, Peng Qinke, Sun Shiquan, et al. Collaborative filtering recommendation algorithm based on uses preference derived from item domain features[J]. Physica A, 2014, 396(2): 66-76.