Operations Research and Fuzziology 运筹与模糊学, 2022, 12(1), 119-124
Published Online February 2022 in Hans. http://www.hanspub.org/journal/orf
https://doi.org/10.12677/orf.2022.121012

Hans汉斯

推荐算法的相似度计算综述

黄向春，赵芬霞，安建业

天津商业大学理学院，天津

收稿日期：2022年1月12日；录用日期：2022年2月14日；发布日期：2022年2月21日

摘　要

相似度的计算作为推荐算法中的核心内容，合适的相似度计算方法对推荐算法的推荐效果有着重要的影响。总结了推荐算法中常用的相似度计算方法，并对这几种相似度计算方法的局限性和特点做了对比分析，最后对相似度计算方法的改进做了简单总结。

关键词

推荐算法，相似度，协同过滤

Review on Similarity Calculation of Recommendation Algorithms

Xiangchun Huang, Fenxia Zhao, Jianye An

School of Science, Tianjin University of Commerce, Tianjin

Received: Jan. 12th, 2022; accepted: Feb. 14th, 2022; published: Feb. 21st, 2022

Abstract

The calculation of similarity is the core content of the recommendation algorithm, and the appropriate similarity calculation method has an important influence on the recommendation effect of the recommendation algorithm. This paper summarizes the similarity calculation methods commonly used in the recommendation algorithm, and compares and analyzes the limitations and characteristics of these similarity calculation methods, and finally summarizes the improvement of similarity calculation methods.

Keywords

Recommendation Algorithm, Similarity, Collaborative Filtering

文章引用: 黄向春, 赵芬霞, 安建业. 推荐算法的相似度计算综述[J]. 运筹与模糊学, 2022, 12(1): 119-124.
DOI: 10.12677/orf.2022.121012

1. 引言

随着互联网的发展和大数据时代的到来，信息过载问题日益严重，推荐算法作为解决信息过载问题重要的技术，在电影、新闻、电子商务、音乐等领域的发展中起到了重要的作用[1]。推荐算法的核心内容是计算用户或项目之间的相似度，通过相似度的计算找到用户感兴趣的物品推荐给用户。因此，相似度的计算在推荐算法中具有重要的意义。

2. 推荐算法中常用的相似度计算方法

用来计算用户或项目之间相似性的方法有很多，常用的相似度计算方法有 Jaccard 相似系数、余弦相似性、Pearson 相似性、距离相似性等。

2.1. Jaccard 相似系数

杰卡德系数是衡量两个集合间相似性的常用公式，两个集合U 和V 的交集元素在U 和V 的并集中所占的比例，称为两个集合的杰卡德相似系数，用符号 $J(U,V)$ 表示[2]。对应计算公式如下：

$$\text{Jaccard}(U,V)=\frac{|U\cap V|}{|U\cup V|} \tag{1}$$

从杰卡德系数的计算公式可知， $J(U,N)$ 的取值范围为[0, 1]该系数越大，相似性越高。

2.2. 余弦相似性

几何中的夹角余弦是用来衡量两个向量方向的差异。在二维空间中，向量 $\boldsymbol{u}(x_1,y_1)$ 与向量 $\boldsymbol{v}(x_2,y_2)$ 的夹角余弦公式为：

$$\cos(\theta)=\frac{x_1x_2+y_1y_2}{\sqrt{x_1^2+y_1^2}\sqrt{x_2^2+y_2^2}} \tag{2}$$

在推荐算法中通常使用余弦相似性进行用户(或物品)的相似度计算，以衡量用户(或物品)之间的差异。假设推荐的主体是用户，那么使用余弦相似性对用户 u,v 都评分过的项目进行描述，并确定其相似性[3]。假设用两条向量表示两名用户 u,v 对 n 个项目的评分，分别表示为 $\boldsymbol{u}=(r_{u1},r_{u2},r_{u3},\cdots,r_{un})$，$\boldsymbol{v}=(r_{v1},r_{v2},r_{v3},\cdots,r_{vn})$，则用户 u,v 之间的相似度为：

$$\text{sim}(u,v)=\cos(\boldsymbol{u},\boldsymbol{v})=\frac{\boldsymbol{u}\cdot\boldsymbol{v}}{|\boldsymbol{u}|\cdot|\boldsymbol{v}|}=\frac{\sum_{i=1}^{n}r_{ui}r_{vi}}{\sqrt{\sum_{i=1}^{n}r_{ui}^2}\sqrt{\sum_{i=1}^{n}r_{vi}^2}} \tag{3}$$

2.3. Pearson 相关系数

Pearson 相关系数是衡量两个随机变量之间线性相关程度的统计量，相关系数的取值范围为[−1, 1]，相关系数的绝对值越大，表明两个变量之间的相关度越高。随机变量 U, V 的相关系数公式为：

$$\rho_{UV}=\frac{Cov(U,V)}{\sqrt{D(U)}\sqrt{D(V)}}=\frac{E\left((U-EU)(V-EV)\right)}{\sqrt{D(U)}\sqrt{D(V)}} \tag{4}$$

Pearson 相关系数是相似度计算过程中使用最为广泛计算方法。相对于夹角余弦公式，Pearson 相关系数对变量进行了去中心化处理，其好处是减少变量个体的数值差异对变量间相似度的影响[4]。假设用两条向量表示两名用户 u,v 对 n 个项目的评分，分别表示为 $\boldsymbol{u}=(r_{u1},r_{u2},r_{u3},\cdots,r_{un})$，$\boldsymbol{v}=(r_{v1},r_{v2},r_{v3},\cdots,r_{vn})$，则用户 u,v 之间的相似度为：

$$\mathrm{sim}(u,v)=\frac{\sum_{i=1}^{n}(r_{ui}-\overline{r}_u)(r_{vi}-\overline{r}_v)}{\sqrt{\sum_{i=1}^{n}(r_{ui}-\overline{r}_u)^2}\sqrt{\sum_{i=1}^{n}(r_{vi}-\overline{r}_v)^2}} \tag{5}$$

2.4. 距离相似度计算方法

2.4.1. 欧式距离

欧氏距离也叫欧几里得距离，指 n 维空间中两个点的真实距离。在二维空间中，向量 $\boldsymbol{u}(x_1,y_1)$ 与向量 $\boldsymbol{v}(x_2,y_2)$ 的欧氏距离公式为：

$$d_{uv}=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \tag{6}$$

欧式距离先将特征数据标准化，然后计算距离。欧氏距离能够体现个体数值特征的绝对差异，一般用于需要从维度的数值大小中体现差异的相关度分析[5]。然而由于其是基于各维度特征计算相似度，所以如果特征刻度指标不同，其计算结果可能会失效。对于用户 u,v 使用欧氏距离计算其相似度，则用户 u,v 之间的相似度为：

$$\mathrm{sim}(u,v)=\sqrt{\sum_{i=1}^{n}(r_{ui}-r_{vi})^2}. \tag{7}$$

2.4.2. 明可夫斯基距离

该距离公式为多个距离公式的扩展，是概括描述。欧几里得距离、曼哈顿距离、马哈拉诺比斯距离等是其特例，p 值为 2 则该公式退化为欧几里得距离[6]。对于用户 u,v 的明可夫斯基距离公式计算其相似度为：

$$\mathrm{sim}(u,v)=\left(\sum_{i=1}^{n}|r_{ui}-r_{vi}|^p\right)^{\frac{1}{p}} \tag{8}$$

2.4.3. 切比雪夫距离

切比雪夫距离起源于国际象棋中国王的走法，在二维空间中，向量 $\boldsymbol{u}(x_1,y_1)$ 与向量 $\boldsymbol{v}(x_2,y_2)$ 的欧氏距离公式为：

$$d_{uv}=\max\left(|x_1-x_2|,|y_1-y_2|\right) \tag{9}$$

对于用户 u,v 的切比雪夫距离公式计算其相似度为：

$$\mathrm{sim}(u,v)=\max\left(\sum_{i=1}^{n}|r_{ui}-r_{vi}|\right) \tag{10}$$

上文提到的相似度计算公式在推荐算法中存在的局限性及特点可以总结为以下几个方面如表 1 所示：

Table 1. Comparison of similarity calculation methods
表 1. 相似度计算方法对比

相似性计算方法	局限性	特点
Jaccard 相似系数	对于稀疏数据，共同评分项目非常稀少时相似度计算不准确；样本的特征值为非二进制，无法使用；容易受热门项目的影响。	值域[0, 1]，计算复杂度较低。适用于样本的特征值为二进制。平行的评分向量下不受影响。
余弦相似性	对于稀疏数据，共同评分项目非常稀少时相似度计算不准确；存在平行的评分向量，无法直接对其相似度关系进行描述。	值域[−1, 1]，对向量进行了归一化处理，解决了向量个体间存在度量标准不统一问题产生的计算偏差；相比于距离相似性能够很好的对向量间的相似度值进行了量化。
Pearson 相似性	对于稀疏数据，共同评分项目非常稀少时相似度计算不准确；存在平行的评分向量，无法直接对其相似度关系进行描述。	值域[−1, 1]，相比于余弦相似度对变量进行了均值化(或去中心化)处理，减少变量个体的数值差异对变量间相似度的影响。
距离相似性	对于稀疏数据，共同评分项目非常稀少时相似度计算不准确；特征刻度指标不同，其计算结果可能会失效。	值域[0, ∞]；从向量间的绝对距离区分差异，对向量各个维度内的数值特征非常敏感；用于需要从维度的数值大小中体现差异的相关度分析。

3. 相似度计算方法的改进

3.1. 基于余弦相似度的改进

用户间在相似度计算过程中，由于用户评分习惯的不同往往存在明显的差异性，而先前使用的余弦相似度方法没有考虑到用户间的评分区别，可以通过减掉用户全部评分的平均值进行改进，去除用户评价尺度差异性带来的影响。改进的余弦相似度公式如下：

$$\mathrm{sim}(u,v)=\frac{\sum_{i=1}^{n}\left(r_{ui}-\overline{r}_u\right)\left(r_{vi}-\overline{r}_v\right)}{\sqrt{\sum_{i=1}^{n}\left(r_{ui}-\overline{r}_u\right)^2}\sqrt{\sum_{i=1}^{n}\left(r_{vi}-\overline{r}_v\right)^2}} \tag{11}$$

Chen 等[7]从用户评分角度出发，将改进的余弦相似度用于计算不同用户间的项目评级尺度差异，提出了一种改进的基于优化用户相似度的 CF 算法，该算法推荐精确度显著提高。

3.2. 基于惩罚项的相似度改进

对于活跃用户与热门物品相似度计算的问题，他认为活跃用户对物品相似度的贡献应该小于不活跃的用户，因此在计算时要降低活跃用户对相似度权重的影响，John S.Breese 等[8]提出了 IUF (Inverse User Frequence)方法，引入用户活跃度对数的倒数的参数，即增加$\frac{1}{\log\left(1+\left|N(u)\right|\right)}$项，改进后的用户相似度的计算公式：

$$\mathrm{sim}(i,j)=\frac{\sum u\in N(i)\cap N(j)\frac{1}{\log\left(1+\left|N(u)\right|\right)}}{\sqrt{\left|N(i)\right|\left|N(j)\right|}} \tag{12}$$

同样的可以通过增加$\frac{1}{\log\left(1+\left|N\left(i\right)\right|\right)}$项，惩罚用户$u,v$共同兴趣列表中热门物品对他们相似度的影响。项亮[9]引入热门商品与该商品的几何平均值以降低热门商品与其他商品的相似度，可以通过在分母上加大对热门物品的惩罚公式如下：

$$\mathrm{sim}\left(i,j\right)=\frac{\left|N\left(i\right)\cap N\left(j\right)\right|}{\left|N\left(i\right)\right|^{1-\alpha}\left|N\left(j\right)\right|^{\alpha}} \tag{13}$$

其中$\alpha\in\left[0.5,1\right]$通过提高$\alpha$就可以惩罚热门物品$j$。对于改进后的算法 ItemCF-IUF 与 temCF 算法对比其结果如表 2 所示：

Table 2. Comparison between ItemCF-IUF algorithm and temCF algorithm in MovieLens data set
表 2. MovieLens 数据集中 ItemCF-IUF 算法与 temCF 算法对比

	准确率	召回率	覆盖率	流行度
temCF	22.28%	10.76%	18.84%	7.254526
ItemCF-IUF	22.29%	10.77%	19.70%	7.217326

ItemCF-IUF 算法与 temCF 算法在准确率和召回率上差别不大，ItemCF-IUF 算法的覆盖率明显高于 temCF 算法。推荐结果的流行度得到了降低，表明加入惩罚项的相似度改进提高了 temCF 算法的推荐效果。

3.3. 基于时间衰减因子的相似度改进

用户近期的行为最能反映出用户的当前兴趣，董立岩等[10]在相似度矩阵的计算过程中融入时间衰减因子，公式如下：

$$\mathrm{sim}\left(i,j\right)=\frac{\sum u\in N\left(i\right)\cap N\left(j\right)\frac{f\left(\left|t_{ui}-t_{uj}\right|\right)}{\log\left(1+\left|N\left(u\right)\right|\right)}}{\sqrt{\left|N\left(i\right)\right|\left|N\left(j\right)\right|}} \tag{14}$$

衰减项中t_{ui},t_{uj}为用户u对物品i和j产生浏览行为的时间。其时间衰减函数公式如下：

$$f\left(\left|t_{ui}-t_{uj}\right|\right)=\frac{1}{1+\alpha\left|t_{ui}-t_{uj}\right|} \tag{15}$$

α为时间衰减因子的影响系数，用户兴趣变化越快，α的值越大，反之越小。实验表明改进算法使推荐结果更具时效性。

3.4. 基于融合多种相似度的改进

在相似度计算中可以根据样本数据的特征将不同的相似度计算方法相融合。蒋宗礼，李慧[11]等将信任度引入到协同过滤算法中与用户相似度相融合，在相似度计算过程中选择余弦相似度和 Jaccard 相似度相结合的方式计算用户间的相似度，以避免余弦相似度基于用户共同评价项目所带来的严重缺陷，计算公式如下所示：

$$\mathrm{sim}\left(u,v\right)=\cos\mathrm{sim}\left(u,v\right)\cdot Jac\left(u,v\right) \tag{16}$$

方惠[12]等在传统相似度矩阵计算中引入时间衰减函数和物品惩罚因子，得到改进相似度矩阵，来计

算项目之间的相似度。实验表明，改进后的算法能有效提高系统推荐的准确性。Azene Zenebe [13]等在相似度计算时引用模糊集合，用模糊集合的方法去计算项目之间的相似度。

4. 结论

本文对推荐算法中常用的相似度计算方法做了总结，并比较分析了几种相似度计算方法的局限性和特点，最后对相似度计算方法的改进做了简单的归纳。分别展示了对公式进行调整的基于余弦的相似度改进；减低活跃用户和热门商品流行度的加入惩罚项的相似度改进；根据用户兴趣模型加入时间因子的相似度改进；融合多种相似度计算方法的相似度计算改进。未来将对模糊数加入相似度的计算的方法进行研究。

基金项目

国家社科青年项目(20CTJ011)。

参考文献

[1] Adomavicius, G. and Kwon, Y. (2014) Optimization-Based Approaches for Maximizing Aggregate Recommendation Diversity. *Informs Journal on Computing*, **26**, 351-369. https://doi.org/10.1287/ijoc.2013.0570

[2] 尹毫, 焦文彬, 史广军, 等. 加入惩罚因子的基于物品的协同过滤算法[J]. 科研信息化技术与应用, 2019, 10(2): 10-19.

[3] Dhawan, S., Singh, K. and Jyoti (2015) High Rating Recent Preferences Based Recommendation System. *Procedia Computer Science*, **70**, 259-264. https://doi.org/10.1016/j.procs.2015.10.085

[4] Ahn, H.J. (2008) A New Similarity Measure for Collaborative Filtering to Alleviate the New User Cold-Starting Problem. *Information Sciences*, **178**, 37-51. https://doi.org/10.1016/j.ins.2007.07.024

[5] Toori, S. and Esmaeily, A. (2017) A Novel 3D Intelligent Fuzzy Algorithm Based on Minkowski-Clustering. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **XLII-4/W4**, 293-297. https://doi.org/10.5194/isprs-archives-XLII-4-W4-293-2017

[6] Liberti, L. and Lavor, C. (2017) Euclidean Distance Geometry. https://doi.org/10.1007/978-3-319-60792-4

[7] Chen, H., Li, Z.K. and Hu, W. (2016) An Improved Collaborative Recommendation Algorithm Based on Optimized User Similarity. *The Journal of Supercomputing*, **72**, 2565-2578. https://doi.org/10.1007/s11227-015-1518-5

[8] Breese, J.S. (1998) Emprical Analysis of Predictive Alogorithms for Collaborative Filtering. Morgan Kaufmann Publishers.

[9] 项亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012.

[10] 董立岩, 王越群, 贺嘉楠, 等. 基于时间衰减的协同过滤推荐算法[J]. 吉林大学学报(工学版), 2017, 47(4): 1268-1272.

[11] 李慧, 蒋宗礼. 融合用户相似度与信任度的协同过滤推荐算法[J]. 软件导刊, 2017, 16(6): 28-31.

[12] 方惠, 李民, 邓秀辉, 余开朝. 改进物品相似度计算的协同过滤算法[J]. 软件导刊, 2021, 20(9): 89-91.

[13] Zenebe, A. and Norcio, A.F. (2009) Representation, Similarity Measures and Aggregation Methods Using Fuzzy Sets for Content-Based Recommender Systems. *Fuzzy Sets and Systems*, **160**, 76-94. https://doi.org/10.1016/j.fss.2008.03.017