

协同过滤算法中相似度计算问题研究

樊艳清¹, 梁宏宇², 纪佳琪²

(1. 河北民族师范学院 信息中心, 河北 承德 067000;

2. 河北民族师范学院 数学与计算机科学学院, 河北 承德 067000)

摘要:大数据时代由于信息过载问题使人们无法在海量数据中快速精准地获取有效信息。为了解决个性化信息呈现问题,推荐系统应运而生。在众多推荐算法中,协同过滤通过分析用户的历史行为信息,能够得到良好的推荐结果,成为推荐系统中使用最普遍的算法。协同过滤算法中相似度的计算方式直接影响着推荐结果。针对目前缺乏综合评价不同相似度对推荐结果评价指标影响的相关研究,深入全面分析研究了余弦相似度、调整的余弦相似度、皮尔森相关系数、欧几里德相似度、谷本系数5种不同相似度的计算方法,并分析了对准确率、召回率、平均调和数和覆盖率4种不同评价指标的影响。通过在真实数据集上的实验结果,给出了不同相似度计算方式的适用场景和优缺点。

关键词:协同过滤;推荐系统;相似度;评分矩阵;大数据

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2020)08-0091-06

doi:10.3969/j.issn.1673-629X.2020.08.015

Research on Similarity Calculation in Collaborative Filtering Algorithm

FAN Yan-qing¹, LIANG Hong-yu², JI Jia-qi²

(1. Information Center, Hebei Normal University for Nationalities, Chengde 067000, China;

2. School of Math and Computer Science, Hebei Normal University for Nationalities, Chengde 067000, China)

Abstract: In the era of big data, people cannot get effective information quickly and accurately in massive data due to information overload. In order to solve the problem of personalized information presentation, the recommender system emerge at the historic moment. Among different recommendation algorithms, collaborative filtering can obtain ideal recommendation results by analyzing the historical behavior of users, making it the most commonly used algorithm in the recommendation system. The similarity calculation method of collaborative filtering algorithm directly affects the recommendation results. A comprehensive research on five similarity calculation methods is carried out since it is lack of comprehensive evaluation on the effect of varied similarity calculation methods on recommendation results. Four different evaluating indicator such as precision, recall, harmonic mean and coverage are analyzed. With the experimental results on the real datasets, the application domains and scenarios, advantages and disadvantages of different similarity calculation methods are proposed.

Key words: collaborative filtering; recommendation system; similarity; rating matrix; big data

0 引言

随着大数据时代的到来,信息过载问题越来越严重^[1-2],现有的技术已不能很好地处理日益加速增长的数据,数据的增长率已远大于人们的处理能力。在海量的文章、声音、视频中人们无法有效地查询到需要的信息,技术的壁垒成为人们在海量数据中获取有价值信息的障碍。虽然传统的搜索引擎技术能够解决人们在海量数据中搜索有效信息的问题,但是搜索结果

依赖于用户输入的关键字且不能针对不同用户的喜好给出个性化的结果,为此推荐系统应运而生^[3-4]。

推荐系统根据实现方式的不同又可分为基于内容的推荐系统、协同过滤推荐系统、混合推荐系统和基于深度学习的推荐系统。其中协同过滤是推荐系统众多算法中使用最广泛、推荐效果较好的一种算法^[5-6]。协同过滤算法并不需要获取用户信息和物品的属性信息,只需要获取用户对物品的评分即可,因此数据获取

收稿日期:2019-09-08

修回日期:2020-01-08

基金项目:河北省高等学校科学技术研究项目(Z2019053);2019年度引进留学人员资助项目(C20190179);2019年度河北省引进国外智力项目(2019YX009B)

作者简介:樊艳清(1987-),女,研究方向为数据挖掘;梁宏宇,高级工程师,研究方向为大数据处理。

简单,不需要抽取用户或物品特征,算法实现也相对简单。根据应用场景不同,协同过滤算法可划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^[7-8]。基于用户的协同过滤算法的主要思想是:根据用户对物品的评分,计算用户间的相似度,然后推荐相似用户喜好而自己又没购买的物品^[9]。但是这种方法的可扩展性不好,随着系统中用户数量的不断增加,用户间的相似度计算将越来越困难。基于物品的协同过滤算法的主要思想是:根据用户对物品的评分,计算物品间的相似度,然后推荐给用户和他们之前购买物品相似的物品^[7,10],由于系统中物品一般不会过多,因此可扩展性好,计算速度快。但是无论是基于用户的还是基于物品的协同过滤算法,相似度的计算都是一个重要的中间环节,其采用不同的相似度计算方式,将会对最终的推荐结果产生较大影响。

然而,虽然目前很多研究给出了不同相似度计算方法^[5,11-12],但是却很少有全面深入分析不同相似度计算方式对推荐结果产生影响的研究,也缺少从不同评价指标进行的对比评测。因此无法很好地指导工业实践。为此,文中总结了基于物品的协同过滤算法中各种相似度的计算方式,并从准确率、召回率、平均调和数和覆盖率这些指标进行了评测,并根据评测结果给出了理论上的解释说明。

1 相似度计算

文中以基于物品的协同过滤算法实现为基础,即计算物品与物品之间的相似度从而得到物品相似度矩阵。基于物品的协同过滤算法中最关键的步骤是如何计算物品间的相似度,然后选取最相似的 k 个物品。相似度计算的基本思想是找到所有用户对两个物品 i 和 j 的共同评分项,然后使用某种相似度计算方法计算物品 i 和 j 之间的相似度 s_{ij} 。例如,所有用户对物品 i 的评分为 $[1, ?, 2, 3, ?]$ ($?$ 号表示没有评分),所有用户对物品 j 的评分为 $[2, 5, 4, ?, 1]$,则需要计算向量 $[1, 2]$ 和向量 $[2, 4]$ 之间的相似度。文中提出的相似度的度量方式有^[13-15]:

(1) 余弦相似度 (cosine similarity); (2) 调整的余弦相似度 (adjusted cosine similarity); (3) 皮尔逊相关系数 (Pearson correlation coefficient); (4) 欧几里德相似度 (Euclidean similarity); (5) 谷本系数 (Tanimoto coefficient)。

为了论述清晰,首先给出文中用到的数学符号定义。设 \mathbf{R} 表示用户评分矩阵,在 \mathbf{R} 中每一行表示用户所有物品的评分,每一列表示某一物品被所有用户的评分。 r_{ij} 表示 \mathbf{R} 中的单元格,即第 i 行第 j 列,表示用户 i 对物品 j 的评分。假设 \mathbf{R} 中含有 3 个用户对 4 个物品

的评分,则 \mathbf{R} 可以表示为:

$$\mathbf{R} = \begin{matrix} & \begin{matrix} I_1 & I_2 & I_3 & I_4 \end{matrix} \\ \begin{matrix} U_1 \\ U_2 \\ U_3 \end{matrix} & \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \end{bmatrix} \end{matrix}$$

为了书写方便,使用 \vec{x}_i 表示 \mathbf{R} 的第 i 行向量,使用 \vec{y}_j 表示 \mathbf{R} 的第 j 列向量。例如 $\vec{x}_i = \{r_{11}, r_{12}, r_{13}, r_{14}\}$, $\vec{y}_j = \{r_{11}, r_{21}, r_{31}\}$ 。

1.1 余弦相似度

此时,两个待计算相似度的物品被看作 m 维的两个向量。余弦相似度通过计算两个向量间的夹角余弦值来度量相似度,其取值范围为 $[0, 1]$ 。夹角越小,余弦值越接近 1,相似度越高。反之,夹角越大,余弦值越接近 0,相似度越低。其计算公式如下:

$$\text{sim}(\vec{y}_i, \vec{y}_j) = \cos(\vec{y}_i, \vec{y}_j) = \frac{\vec{y}_i \cdot \vec{y}_j}{\|\vec{y}_i\| \times \|\vec{y}_j\|} = \frac{\sum_{u=1}^n r_{ui} + r_{uj}}{\sqrt{\sum_{u=1}^n r_{ui}^2} \sqrt{\sum_{u=1}^n r_{uj}^2}} \quad (1)$$

1.2 调整的余弦相似度

基于用户的协同过滤与基于物品的协同过滤在计算相似度时的不同处是:基于用户的协同过滤是沿着评分矩阵 \mathbf{R} 的行进行的,而基于物品的评分矩阵是沿着评分矩阵的列进行的,每一对共同评分项对应的都是不同的用户,因此使用余弦相似度计算物品间相似度有一个重大缺陷:没有考虑不同用户对不同物品的评分尺度。而且余弦相似度从夹角(或方向)上度量了差异,对维度的数值并不敏感。例如,两个用户对两个物品的评分分别为 $(1, 2)$ 和 $(4, 5)$,使用余弦相似度计算两个物品的相似度结果都是 0.98,说明两个物品很相似,但从评分上显然可以看出这两个物品是不相似的。为了修正这种不合理性,需要在所有维度上减去一个均值,则调整的余弦相似度定义如下:

$$\text{sim}(\vec{y}_i, \vec{y}_j) = \frac{\sum_{u=1}^n (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u=1}^n (r_{ui} - \bar{r}_u)^2} \times \sqrt{\sum_{u=1}^n (r_{uj} - \bar{r}_u)^2}} \quad (2)$$

其中, \bar{r}_u 表示用户 u 对所有物品评分的平均值。

1.3 皮尔森相关系数

皮尔森相关系数描述了 2 个向量的线性相关性,其取值范围为 $[-1, 1]$,越接近 1 表示越线性正相关,越接近 -1 表示越线性负相关,越接近 0 表示线性不相关。其定义为 2 个向量的协方差与 2 个向量标准差乘

积的商,具体表示为:

$$\begin{aligned} \text{sim}(\vec{y}_i, \vec{y}_j) &= \frac{\sum_{u=1}^n (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u=1}^n (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u=1}^n (r_{uj} - \bar{r}_j)^2}} \\ &= \frac{\sum_{u=1}^n (r_{ui} \times r_{uj}) - \frac{\sum_{u=1}^n r_{ui} \sum_{u=1}^n r_{uj}}{N}}{\sqrt{(\sum_{u=1}^n r_{ui}^2 - \frac{(\sum_{u=1}^n r_{ui})^2}{N})(\sum_{u=1}^n r_{uj}^2 - \frac{(\sum_{u=1}^n r_{uj})^2}{N})}} \end{aligned} \quad (3)$$

上述2个公式是等价的,其中 \bar{r}_i 表示所有用户对物品*i*的平均评分, N 表示向量的维度。

1.4 欧几里德相似度

欧几里德一般是用来度量距离的,数值越大,其距离越远^[16]。因此可以取其倒数,这样距离越远,其倒数越接近0,表示其相似度越低,数学定义如下:

$$\text{sim}(\vec{y}_i, \vec{y}_j) = \frac{1}{1 + \sqrt{\sum_{u=1}^n (r_{ui} - r_{uj})^2}} \quad (4)$$

公式中分母加1的主要原因是防止分母为0的情况出现,从公式中可以看出欧几里德相似度的取值范围为(0,1]。由于欧几里德相似度需要参加计算的向量单位是一致的,因此,在应用该公式前,需要对参与计算的向量做标准化处理,公式如下:

$$X = \frac{X - \bar{X}}{\sigma} \quad (5)$$

其中, X 表示参与计算的向量, \bar{X} 表示 X 的均值, σ 表示 X 的方差。

1.5 谷本系数

谷本系数又称广义的杰卡德(Jaccard)相似系数^[17],它的实质是2个物品交集与并集的比值。传统的杰卡德相似度只能计算只含0和1的向量,而谷本系数把其扩展到了实数范围内,其定义如下:

$$\begin{aligned} \text{sim}(\vec{y}_i, \vec{y}_j) &= \frac{\vec{y}_i \cdot \vec{y}_j}{\|\vec{y}_i\| + \|\vec{y}_j\| - \vec{y}_i \cdot \vec{y}_j} = \\ &= \frac{\sum_{u=1}^n r_{ui} \times r_{uj}}{\sqrt{\sum_{u=1}^n r_{ui}^2} + \sqrt{\sum_{u=1}^n r_{uj}^2} - \sum_{u=1}^n r_{ui} \times r_{uj}} \end{aligned} \quad (6)$$

2 基于物品的协同过滤算法

本节先介绍基于物品的协同过滤算法。与基于用户的协同过滤算法不同,基于物品的协同过滤关注的是用户评分的物品间的相似度。其主要思想是:计算

物品*i*与其他物品间的相似度,并选择*k*个最相似的物品 $\{i_1, i_2, \dots, i_k\}$,物品*i*与其他*k*个物品间的相似度可以记为 $\{s_{i1}, s_{i2}, \dots, s_{ik}\}$ 。找到最相似的*k*个物品后,通过计算待推荐用户*u*对该物品的加权平均值就可以得到用户*u*对物品*i*的喜好程度。因此基于物品的协同过滤算法主要是两个过程:一是相似度的计算,二是喜好程度(用户对物品的预测打分)的计算。

图1展示了基于物品的协同过滤算法的主要过程:

(1)图1左侧是用户评分矩阵 R ,每一行代表一个用户,每一列代表一个物品,每一单元格代表某一用户对某一物品的评分值。其评分值根据用户的喜好程度从1至5不等(不同的数据集该处取值可能不同)。如果用户尚未对某物品评分,则用一问号表示,也就是要预测的评分值。

(2)根据用户评分矩阵 R ,利用第1部分介绍的相似度计算方法,可以计算得到两两物品间的相似度,从而得到物品间相似度矩阵。显然采用不同的相似度计算方式,将得到不同的物品相似度矩阵。由于该矩阵是对称矩阵(因为物品 i_1 和物品 i_2 的相似度与物品 i_2 和物品 i_1 的相似度是相同的),因此在计算时只需计算不含对角线的上半角矩阵即可(对角线上的值全为1,即自身与自身的相似度为1)。

(3)计算用户对物品的喜好程度,由式(7)完成用户*u*对未购买的物品*i*进行评分预测。其原理是物品*i*被其他用户评分的平均值加上用户*u*对与*i*相似的*k*个物品的加权评分值。

$$P_{u,i} = \bar{r}_i + \frac{\sum_{j \in s(n)} \text{sim}(i,j) \times (r_{u,j} - \bar{r}_j)}{\sum_{j \in s(n)} \text{sim}(i,j)} \quad (7)$$

其中, $P_{u,i}$ 表示用户*u*对物品*i*的预测评分, $s(n)$ 表示与物品*i*最相似的*k*个物品的集合,即物品*i*的近邻。

(4)根据对预测物品的评分值由高到低进行排序。

(5)选取*n*个预测评分最高的物品作为最终的推荐结果。

还有一种与上述方法相似的方法称为回归方法,这种方法使用回归模型来近似评分而不再做权重的相加。

在实践中,使用余弦相似度或皮尔逊相似度有可能得出一些错误结果,比如某两个向量间距离很远(使用欧几里德距离计算),但是计算出来的相似度却很高。因此使用所谓的相似度计算可能导致不好的结果。但文中不考虑这种极端情况的出现,只做一个朴素的假设,即计算得出的相似度高就认为两个物品间的相似度高。

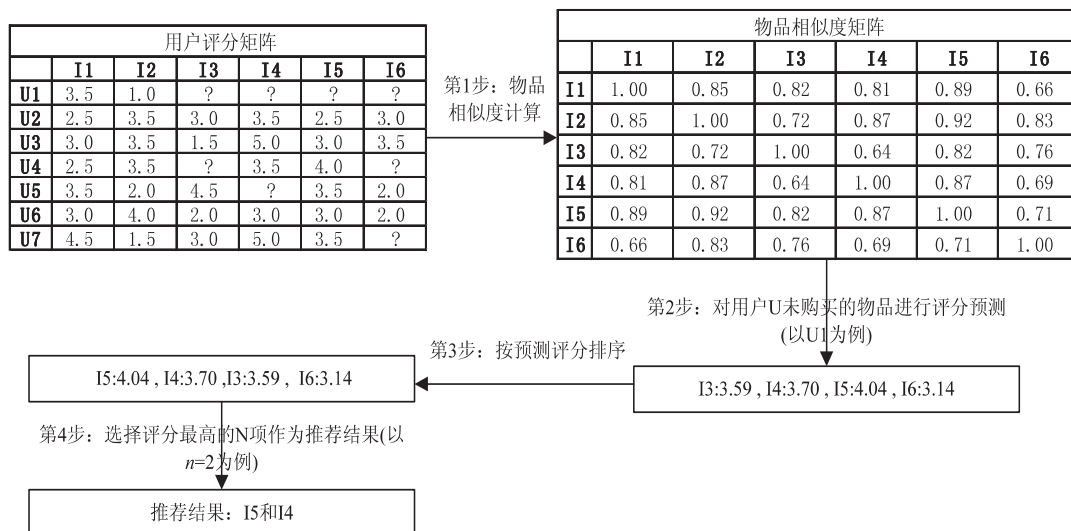


图 1 基于物品的协同过滤算法流程

3 实验

3.1 实验环境

为验证不同相似度计算对推荐结果的影响,使用 Java 语言对所有算法进行了实现。使用的硬件环境为: Intel(R) Core(TM) i5-6200U CPU @ 2.30 GHz 2.40 GHz, 内存 8 G, 250 G 固态硬盘; 软件环境为: Win10 操作系统, JDK1.7, Eclipse 3.5。

在实验中,使用 5 折交叉验证对推荐性能进行评价,即数据集被随机分成互不相关的 5 份,在每一次迭代中做 5 次训练,使用 4 份作为训练集,使用 1 份作为验证集,最后 5 次结果的平均值作为最终结果。

3.2 数据集

文中选取 MovieLens ml-100k 数据集,该数据集包含了 1 000 个用户对 1 700 部电影的共 10 万条评分,每条评分分值范围都是 1 至 5,分值越高表示用户对该电影越喜欢。该数据集的提供者在发布前已经对其进行了数据清洗,保证每个用户至少有 20 条评分。实验前对数据集进行了随机抽样,其中 80% 作为训练集,20% 作为测试集。

3.3 评价指标

不同相似度计算方法对推荐结果影响的评价从以下几个方面考量:

(1) 精确度 (precision): 是广泛用于推荐系统中的一个度量指标。其反映的是推荐给用户物品中是用户感兴趣的占总推荐物品数的比例,即推荐出的物品有多少是准确的。设 $P(u)$ 是根据训练集得到的用户 u 的推荐结果列表, $T(u)$ 是用户 u 在测试集上的评分列表,则形式化定义如下:

$$\text{Precision} = \frac{\sum_{u \in U} |P(u) \cap T(u)|}{\sum_{u \in U} |P(u)|} \quad (8)$$

(2) 召回率 (recall): 也称查全率。其反映的是推荐给用户物品中用户感兴趣的物品占用户所有感兴趣物品的比例,即用户感兴趣的物品中有多少被推荐出来了。其形式化定义如下:

$$\text{Recall} = \frac{\sum_{u \in U} |P(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (9)$$

(3) 调和平均数 (harmonic mean): 精确度和召回率是 2 个相互对立的评价指标,例如如果系统把所有物品都作为推荐物品,显然这时精确度很低,但是召回率却最高为 1;反之如果系统推荐的物品都在测试集中出现,则此时精确度最高为 1,但是召回率却不高。从定义中可以看出精确度用来度量推荐的准确程度,而召回率用来度量推荐的查全程度。为了综合这两个指标,提出了调和平均数这个度量方式:

$$F_1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (10)$$

(4) 覆盖率: 覆盖率反映了系统给不同用户推荐物品的集合(该集合内的物品是不重复的)占系统总物品数的比例。这个值越高说明系统能够尽可能多地把每个物品都推荐出来。其定义如下:

$$\text{Coverage} = \frac{\sum_{u \in U} |P(u)|}{|I|} \quad (11)$$

其中, $|I|$ 表示系统中物品总数。

3.4 结果分析

在实验中,固定选择近邻的个数为 50,推荐列表的长度为 20。同时为了提高实验的准确性,采用 5 折交叉验证方法,并取平均值作为最终的实验结果。其实验数据如表 1 所示。

表1 5种不同相似度计算方式在4种评价指标上的实验数据

指标	Pearson	Cosine	Adjust cosine	Euclidean	Tanimoto
Precision	0.273 4	0.264 4	0.268 0	0.270 9	0.144 2
Recall	0.258 7	0.250 2	0.253 5	0.256 2	0.136 4
F_1	0.265 9	0.257 1	0.260 6	0.263 3	0.140 2
Coverage	0.341 6	0.391 7	0.326 5	0.326 5	0.060 0

为了对上述数据进行分析与解释,使用了5幅不同的图形进行描述。图2显示了5种不同相似度对推荐精确度的影响,结果表明除了使用谷本系数时推荐精确度较低外,其余4种方式精确度非常接近,但皮尔森相关系数相对准确度最高。但余弦相似度和皮尔森相似度也有其固有的缺点,具体可以概括为如下四个方面:(1)扁平值问题:如果所有的评分值都一致,如(1,1,1),使用皮尔逊相似度会由于其分母为0而无法计算;使用余弦相似度的结果会都是1。(2)反向值问题:如果两个用户对物品的评分都是相反的,则皮尔逊相似度的值总是-1。(3)单值问题:如果两个用户只对一个物品有评分,则皮尔逊相关系数无法计算,余弦相似度的值总是1。(4)交叉值问题:如果2个用户仅对2个物品评分并且2个向量相互交叉,如(1,3)和(3,1),皮尔逊相关系数总为-1。

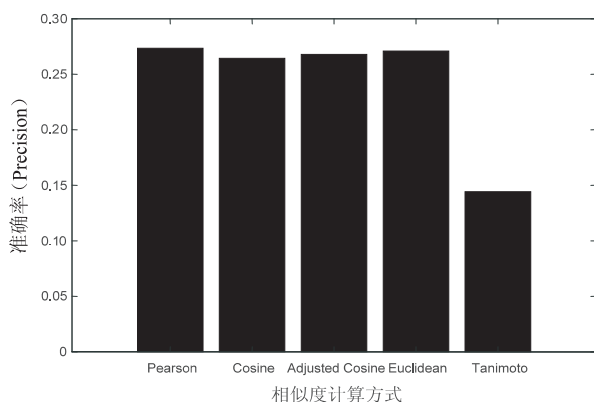


图2 相似度计算方式对精确度的影响

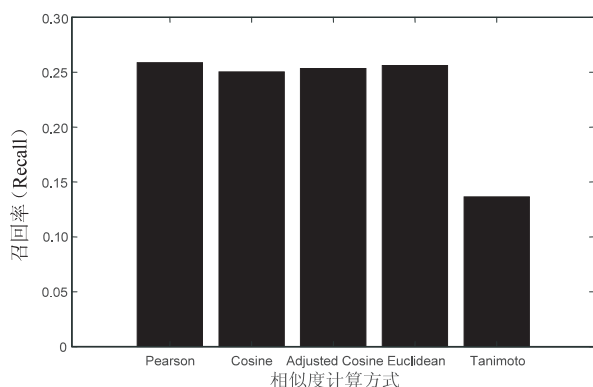


图3 相似度计算方式对召回率的影响

从图3可以看出,除谷本系数外,其余4者的召回率依然非常接近,比较而言皮尔森系数和欧几里德相似度表现略佳。皮尔森相似度与欧几里德相似度的最

大区别在于前者更重视数据集的整体性,因为皮尔森相似度计算的是相对距离,而欧几里德距离计算的是绝对距离。就实际情况来说,使用皮尔森相似度可以适应不同量纲的数据集,而欧几里德距离需要针对数据最归一化,即使数据有相同的量纲。

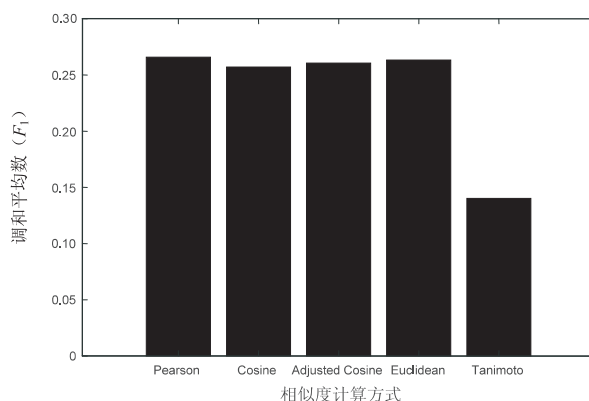


图4 相似度计算方式对调和平均数的影响

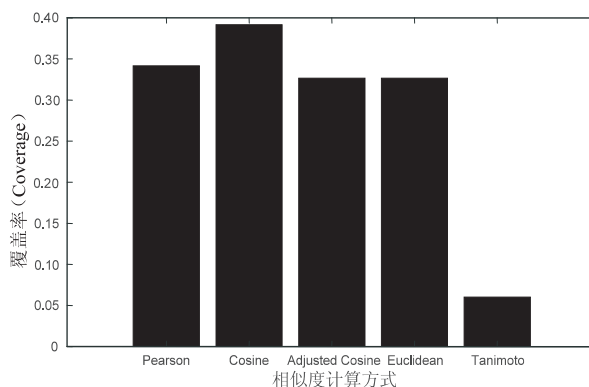


图5 相似度计算方式对覆盖率的影响

图4中调和平均数是精确度与召回率的综合考量,更加综合地反映了对推荐结果的影响。不难看出皮尔森相关系数和杰卡德相似度相对表现最优异。

图5反映了推荐给用户的物品数占总物品数的比例,此时使用余弦相似度要好于其他相似度。

综上所述,皮尔森相似系数在精确度、召回率、调和平均数指标中表现最佳,余弦相似度在覆盖率指标上表现良好,而谷本系数无论在哪个指标上都远远落后于其他相似度计算方法。

4 结束语

协同过滤算法由于其良好的推荐效果得到了广泛的应用。由于协同过滤算法的核心是相似度的计算方

式,不同的相似度计算方式会对结果造成一定的影响。由于目前缺乏不同相似度计算方式对推荐结果能够产生哪些影响的研究,因此文中详细研究了 5 种不同相似度计算方法对基于物品的协同过滤推荐结果的影响,并通过不同的评价指标如准确率、召回率、调和平均数和覆盖率进行了对比分析。下一步将研究其他协同过滤算法中相似度计算对推荐结果的影响,并基于不同数据集进行对比研究。

参考文献:

- [1] HU R, DOU W, LIU J. Clubcf: a clustering-based collaborative filtering approach for big data application [J]. IEEE Transactions on Emerging Topics in Computing, 2014, 2(3): 302-313.
- [2] KARYDI E, MARGARITIS K. Parallel and distributed collaborative filtering: a survey [J]. ACM Computing Surveys, 2016, 49(2): 1-41.
- [3] LEMIRE D, MACLACHLAN A. Slope one predictors for online rating-based collaborative filtering [C]//Proceedings of the 2005 SIAM international conference on data mining. USA: SIAM, 2005.
- [4] 张应辉, 司彩霞. 基于用户偏好和项目特征的协同过滤推荐算法 [J]. 计算机技术与发展, 2017, 27(1): 16-19.
- [5] YANG Z, WU B, ZHENG K, et al. A survey of collaborative filtering-based recommender systems for mobile internet applications [J]. IEEE Access, 2016, 4: 3273-3287.
- [6] HUANG Z, ZENG D, CHEN H. A comparison of collaborative-filtering recommendation algorithms for e-commerce [J]. IEEE Intelligent Systems, 2007, 22(5): 68-78.
- [7] SARWAR B, KARYPIS G, KONSTAN, et al. Item-based collaborative filtering recommendation algorithms [C]//Proceedings of the 10th international conference on world wide web. USA: ACM, 2001.
- [8] HE C, PARRA D, VERBERT K. Interactive recommender systems: a survey of the state of the art and future research challenges and opportunities [J]. Expert Systems with Applications, 2016, 56: 9-27.
- [9] LIU H, KONG X, BAI X, et al. Context-based collaborative filtering for citation recommendation [J]. IEEE Access, 2015, 3: 1695-1703.
- [10] BU J, SHEN X, XU B, et al. Improving collaborative recommendation via user-item subgroups [J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(9): 2363-2375.
- [11] GUO G, ZHANG J, YORKE-SMITH N. A novel recommendation model regularized with user trust and item ratings [J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(7): 1607-1620.
- [12] CHRISTAKOPOULOU E, KARYPIS G. Local item-item models for top-n recommendation [C]//Proceedings of the 10th ACM conference on recommender systems. USA: ACM, 2016.
- [13] 荆常春, 李亚茹, 郭强, 等. 协同过滤推荐算法稳定性研究 [J]. 软件导刊, 2017, 16(2): 39-42.
- [14] 谭学清, 张磊, 黄翠翠, 等. 融合领域专家信任与相似度的协同过滤推荐算法研究 [J]. 现代图书情报技术, 2016(7): 101-109.
- [15] 陈功平, 王红. 改进 Pearson 相关系数的个性化推荐算法 [J]. 山东农业大学学报: 自然科学版, 2016, 47(6): 940-944.
- [16] 王国霞. 基于用户引力的协同过滤推荐算法 [J]. 计算机应用研究, 2016, 33(11): 3329-3333.
- [17] 李斌, 张博, 刘学军, 等. 基于 Jaccard 相似度和位置行为的协同过滤推荐算法 [J]. 计算机科学, 2016, 43(12): 200-205.