计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
48(Suppl.): 180-188, 2011

一种基于多因素的引文推荐方法

石 杰 申德荣 聂铁铮 寇 月 于 戈
(东北大学信息科学与工程学院 沈阳 110004)
(shijie_neu@163.com)

A Citation Recommendation Method Based on Multiple Factors

Shi Jie, ShenDerong, NieTiezheng, KouYue, and Yu Ge
(*College of Information Science and Engineering, Northeastern University, Shenyang* 110004)

Abstract With propagation speed of information increasing, the number of available access to scientific literature is also increasing rapidly. It is a very difficult task for users to browse thousands of query results to get what they need. Citation recommendation is one way to solve this problem. In this paper we get the recommendation citation collection through the available properties (author, year, the reference relationship, etc.) of citations. First we generate a citation reference graph based on the reference relationship of citations. Next, we define a set of rules according to the same author, the common reference and so on. Using these rules, we calculate the weight of the reference edge to express the strength of that association. Then we use a cluster algorithm to cluster closely linked citations. According to the cluster results, we get the Citations to satisfy users' needs.

Key words citation network; clustering; citation recommendation

摘 要 随着信息传播速度的快速提升,可供查阅的科技文献数量也在迅速增加.用户想要在上千条引文查询结果中找到自己需要的结果是一件很困难的事情.结果推荐是解决这个问题的方法之一.通过利用引文可获得的属性(作者、年份、引用关系等)获得需要推荐的引文集合.首先根据引文的引用关系生成一个引文引用图,然后根据同作者、共同引用等定义一系列规则,通过计算给引用边赋权值表示联系的强弱.给出一个聚类算法对联系紧密的引文进行聚类.根据聚类的结果找出用户需要的相关引文.

关键词 引文网络,聚类,引文推荐

中图法分类号 TP391

随着信息时代的到来,科学研究的深度和广度都呈现出了快速增长的趋势.网络的发展、信息传播速度的提升,使得这些研究的成果能够迅速地传播.如此数量庞大的引文集合使得用户的一次搜索通常会返回上千条的结果,用户想发现感兴趣的引文是很困难的一件事.

结果推荐是解决这个问题的一种方法.结果推荐可以根据用户日志等信息,对用户的兴趣进行分析,产生预测,将用户可能感兴趣的项目推荐给用户.根据分析对象的不同,可以将推荐的思想分为基于用户的推荐和基于项目的推荐.基于用户的推荐是通过找到与用户喜好相似的用户群,根据这个用户群的喜好产生预测.基于项目的推荐是通过计算用户已知的感兴趣的项目与其他项目之间的相似性,将相似性最高的项目推荐给用户.

我们将结果推荐的思想应用到引文推荐中,希

收稿日期:2011-07-15
基金项目:国家自然科学基金项目(60973021,61003060);国家"八六三"高技术研究发展计划基金项目(2008AA01Z146)

望将与目标引文关联性强的引文推荐给用户.结果推荐中项目概念在引文推荐中就是引文.

每一篇文献不是独立存在的,新的文献中的成果通常是对旧文献成果的继承和发展.这种引文之间的借鉴和参考关系就形成了引文网络.引文网络描述了知识的发展和研究主题的转移趋势,通过分析引文网络可以发现隐藏的引文之间的联系.这种联系要比分析题目或关键字等字符串相似性的计算要准确得多.目前对引文网络的分析大概分为以下两种类型:1)通过对引文的引用和被引用次数进行统计,利用统计结果评价引文的质量;2)对引文网络反映出的主题作相似性研究,揭示学科的发展趋势.

我们将引文网络应用于引文的推荐中,通过分析引文网络找出关联关系强的引文.Vazquez[1]的实验证明了引文网络具有一定的聚集倾向,这也使得我们通过分析引文网络发现相关引文的想法成为可能.

本文通过建立引文网络,利用聚类的手段,综合考虑了引文之间的引用关系、同作者关系、引用耦合等因素对引文之间关联关系的强度的影响,目的在于寻找关联程度较高的引文,最终产生推荐集合.

1　相关工作

常用的推荐算法分为基于用户和基于项目两种.Shardanand 等人[2]提出了基于用户推荐的算法,目标是找到与目标用户兴趣相似的用户群,并将符合这些用户兴趣的内容推荐给目标用户.该算法的缺点是随着用户的增多计算呈线性增长,性能会变得越来越差,而且不能对推荐结果提供很好的解释.另外,在初始阶段,用户的评分矩阵非常稀疏,该算法并不能提供很好的推荐结果.

Sarwar 等人[3]提出了基于项目推荐的算法,该算法假设大部分用户对同一项目的评分是相似的,基于这种假设可以产生目标项目的最近邻集合,从而产生推荐.Sarwar 等人的算法虽然解决了用户数量增长导致算法性能变差的问题:1)但是依然存在着两个问题,第一,需要用户参与评分,这在实际应用中是不现实的.实际应用中评分矩阵是非常稀疏的,通常只有用户总量的 1/100;2)冷开始问题,即初始状态时没有用户的评分,不能准确对结果进行推荐.

文献[4-6]根据 Sarwar 等人的基于项目推荐的思想,对这种算法作出了改进.不再依赖用户对项目的评分,而是利用项目本身具有的属性计算项目之间的相似程度.这些算法虽然避免了用户的评分,但是在计算相似度时,计算的都是语义的相似程度.这种做法存在一些问题:1)由于抽取技术的原因,抽取的项目关键字存在错误;2)语义分析的结果不是十分准确,有时,语义分析出的相似结果事实上关联性不是很强.现在越来越多的技术考虑从项目获取更多的信息来提高计算相似度的准确率[7].

一些学者将推荐的思想应用到引文的领域.文献[8]根据引文的摘要对用户作出推荐.文献[9]根据上下文的信息对引文作出推荐.

我们希望通过分析引文,并利用引文的作者、发表时间和引用关系等可获得的引文属性,计算引文之间的关联性.

已有的引文分析法有文献偶合法和同引用分析法两种.Kesser[10]提出了文献偶合法中,衡量两篇论文之间的相关性的标准是这两篇论文共同引用的论文的数目.Small[11]提出的同引用分析法中,认为两篇论文如果被同一篇论文引用,那么它们是相关的,同引用的论文数量越多这两篇论文的相关性越强.

本文的主要贡献主要在以下几个方面:

1) 将基于用户和基于项目的思想结合起来,在初期用户评分稀疏的情况下,利用项目之间的相似度进行推荐,在系统运行一段时间之后,根据用户的评价对之前计算的相似度进行调整,使结果更准确;

2) 在计算项目相似度时,根据引文的引用关系等属性计算而不是传统的语义相似度,这样可以避免抽取和语义分析得不准确在计算用户评分时使用用户行为代替用户评分,解决用户评分矩阵十分稀疏的问题;

3) 传统的引文分析在分析引用关系时,只考虑两篇引文的相关程度,我们找出的是一个联系紧密的引文的集合;

4) 我们在分析引用关系的同时考虑了同作者等其他的因素,使相似度的计算更加准确.

2　引文引用图的创建

首先,我们将引文的引用关系用图结构来表示,之后,基于引文引用图进行聚类实现引文推荐.

2.1　引文引用图的创建

引文引用图创建的基本规则:论文引用图 $G=$

(V,E)是一个有向图. 图上的任意点 $s \in V$ 代表一篇引文. 如果引文 e 引用了引文 v,我们用边$(s,v) \in E$ 来表示这个引用关系.

我们选取 17 篇引文根据上述规则生成引文引用图,如图 1 所示. 这 17 篇引文如表 1 所示(这张图仅为了方便算法的讲解,实际生成的图将比这个复杂很多). 下面我们将以图 1 为例,说明如何在引用关系图上找出那些关联强度高的引文.

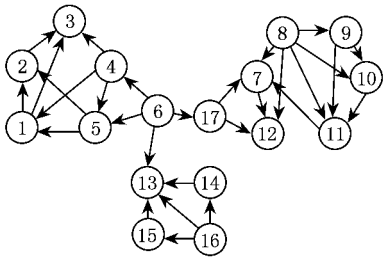

图 1 引文引用图

表 1 引文列表

ID	作者	题目	发表年份
1	Li X, Han J, and Gonzalez H	High-Dimensional OLAP: A Minimal Cubing Approach	2004
2	Wang W, Lu H, Feng J, and Yu J X	Condensed Cube: AnEffective Approach to Reducing Data Cube Size	2002
3	Harinarayan V, Rajaraman A, and Ullman J D	Implementing Data Cubes Efficiently	1997
4	Xin D, Han J, Li X	Star-cubing: computing iceberg cubes by top-down and bottom-upintegration	2007
5	Xin D, Han J, Shao Z	C-cubing: efficient computation of closed cubes by aggregation-based checking.	2006
6	Xin Dong · Alon Halevy · Cong Yu	Data integration with uncertainty	2009
7	Aslam J, Pelekhov K, and Rus D	A practical clustering algorithm for static and dynamic information organization	1999
8	Koudas N, Saha A, Srivastava D, and Venkatasubramanian S	Metric functional dependencies	2009
9	Legany C, Juhasz S, and Babos A	Cluster validity measurement techniques	2006
10	Bilenko M and Mooney R J	Adaptive duplicate detection using learnable string similarity measures	2003
11	Tung H, Ng R T, Lakshmanan L V S, and Han J	Constraint-based clustering in large databases	2001
12	Florescu D, Koller D, Levy A	Using probabilistic information in data integration.	1997
13	Abiteboul S, Duschka O	Complexity of answering queries using materialized views.	1998
14	Bernstein P A, Green T J, Melnik S, Nash A	Implementing mapping composition	2006
15	Fagin R, Kolaitis P G, Popa L	Data exchange: getting to the core. ACM Trans. Database Syst	2005
16	Antova L, Koch C, Olteanu D	World-set decompositions: Expressiveness and efficient algorithms	2007
17	Halevy A Y, Rajaraman A, Ordille J J	Data integration: the teenage years	2006

2.2 各种联系对引文的关联的影响

在前面生成的引文引用图中,每条引用边的重要性是一样的. 实际上有些边在引用图中的地位会更重要一些. 比如同一作者或合作者写的两篇文章之间的联系要比普通两篇有引用关系的引文的联系更强一些. 为了解决这个问题,我们通过加权的方法来提升这些边的重要性.

我们认为有两种引用关系会使得两篇存在引用关系的引文联系更强.

情况 1. 引文 a 引用了引文 b,而且 a 和 b 的作者相同或者部分相同. 这种情况通常是引文 a 是对引文 b 的进一步研究和补充. 因此,这样的两篇引文的联系要比普通的引用联系更强.

情况 2. 如图 1 中编号为 1,2,3 的引文中,引文 2 引用了引文 3,引文 1 引用了 2,3. 通常是引文 1 的作者在阅读引文 2 时,发现了引文 3 的一些观点

对其有帮助,所以阅读并引用了引文 3. 因此,引文 1 与引文 3 的联系要比普通引用更强.

考虑到上述两种情况的存在,我们对引用图进行加权操作. 边(a,b)的权值记为 $l(a,b)$,具体定义为

$$l(a,b)=1-\alpha\cdot\omega(a,b)-\beta\cdot\phi(a,b),$$

其中,边的默认权值为 1,$\omega(a,b)$,$\phi(a,b)$分别表示上述两种情况对引文联系的影响,α,β 为影响系数. 在这里我们设置 $\alpha=0.9$,$\beta=0.6$.

只有在情况 1 不存在时我们才会考虑情况 2. 即当 $\alpha\omega(a,b)>0$ 时,$\beta=0$. 因为同作者的关系比引用更能说明文章的相关度. 下面我们将分别叙述这两种影响的计算方法.

$\omega(a,b)$表示 a,b 两篇引文的共同作者对 a,b 两篇引文的引用边重要性的影响,具体计算定义为

$$\omega(a,b)=Au(a,b)\cdot y(a,b),$$

其中,

$$Au(a,b)=\frac{author(a)\cap author(b)+\lambda}{AVG(Count(author(a),author(b)))+\lambda},$$

$$y(a,b)=\begin{cases}1, & |year(a)-year(b)|\leqslant 2,\\ \dfrac{1}{e^{|year(a)-year(b)|-3}}, & |year(a)-year(b)|>2,\end{cases}$$

$Au(a,b)$为 a,b 两篇引文的共同作者,在 a,b 两篇引文的所有作者中占的比例. 在分子分母同时添加 λ,是为了防止引文 a,b 的作者数量差太多而影响计算结果. 在这里,我们设置 λ 的值为 a,b 两篇引文的作者数的平均数.

$y(a,b)$为 a,b 两篇引文发表的间隔时间. 作者相同或部分相同的两篇引文 a,b,它们发表的时间间隔越短联系性会越强. 后者很有可能是前者的扩展与补充. 由于引文的关联性会随着引文发表时间间隔的增长而迅速下降,所以我们用$\frac{1}{e^x}$来描述这种趋势.

$\phi(a,b)$为引用参考文献的引用对引用边重要性的影响,具体定义为

$$\phi(a,b)\begin{cases}\lg\sum_{i=1}^{n}Cit(c_i,b), & \sum_{i=1}^{n}Cit(c_i,b)<10,\\ 1, & \sum_{i=1}^{n}Cit(c_i,b)\geqslant 10,\end{cases}$$

其中,c_i 为 a 的参考文献. $Cit(c,b)$表示 c,b 是否存在引用关系. 如果 c 引用了 b,则 $Cit(c,b)=1$,否则 $Cit(c,b)=0$.

a 的参考文献中,引用 b 的越多说明 b 与 a 的研究方向的相关性更强,或者 a 延续并发展了一些 b 的观点.

为此,我们根据相关因素在原引文引用图上加上权重.

例如:假设 4,5 是图中的两篇引文,则

$$\omega_{(4,5)}=Au_{(4,5)}\cdot y_{(4,5)}=Au_{(4,5)}=\frac{author_{(4)}\cap author_{(5)}+\lambda}{AVG_{(}Count_{(}author_{(4)},author_{(5)}))+\lambda}\cdot 1=\frac{2+3.5}{3.5+3.5}=0.78,$$

$$l_{(4,5)}=1-\alpha\cdot\omega_{(4,5)}-\beta\cdot\phi(4,5)=1-0.9\times 0.78-0=0.33.$$

最终经过计算,边(4,5)的权值应为 0.33. 根据相关因素影响计算规则,得到如图 2 所示的加权后的引文引用图. 没有标明权值的边为普通的引用边,权值为 1.

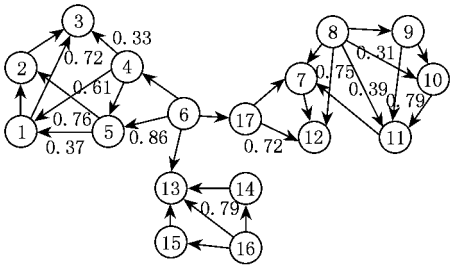

图 2　加权后的引文引用图

3　聚　　类

本节中我们利用之前生成的引文引用图对相关的引文进行聚类,发现那些互相之间有着紧密联系的引文.

3.1　BHC 算法

我们按照基于层次的聚类思想,提出了一种应用在引文引用图上的基本的聚类算法(basic hierarchical clustering, BHC),BHC 算法分为 3 个部分:

1) 初始化. 将每个引文划分为一个类.

2) 计算引文之间的距离,满足条件的进行合并.

3) 持续 2)直到稳定,即没有新产生的类.

在聚类时,我们会用到两种距离:两篇引文之间的距离以及引文与类之间的距离.

① 引文之间的距离

我们用 Dijkstra 的算法计算图上两点的最小距离. 引文的引用是存在时序关系的,即只能由时间点

靠后的引文引用时间点靠前的引文,因此引文引用图 $G(V,E)$是不存在回路的. 这就会导致一个问题:如果 a 到 b 是可达的,那么 b 到 a 必然是不可达的. 如图 3 所示,使用 Dijkstra 算出引文 6 到引文 3 的距离为 1.33,引文 3 到引文 6 距离为∞. 如果用距离表示两篇引文的关联强度,引文 3 到引文 6 距离为∞,说明两篇引文没有任何联系,这种说法显然是不合适的.

基于以上原因,我们将两点距离定义为

$$Distance(a,b)=\begin{cases}D(a,b), & D(a,b)<D(b,a),\\ \alpha\cdot D(b,a), & D(a,b)>D(b,a),\end{cases}$$

其中 $Distance(a,b)$表示 a,b 两篇引文之间的距离. $D(a,b)$表示用 Dijkstra 算出的 a,b 两点之间的距离. 如果 $D(a,b)>D(b,a)$那么 a,b 必然是不可达的. a,b 的距离用 $D(b,a)$乘以一个系数 α 表示. 这里我们设置 $\alpha=1.3$. 乘以一个系数,是因为 a 引用 b,a 对于 b 的关联强度要比 b 对于 a 的关联强度强一些.

② 引文与类之间的距离

我们用引文与类中所有引文的平均距离表示引文与类之间的距离.

$$Distance(a,C(u))=\frac{\sum_{i=1}^{n}Distance(a,u_i)}{n},$$

其中,$Distance(a,C(u))$表示引文 a 与类 $C(u)$之间的距离,u_i 为类中的引文,n 为类 $C(u)$包含的引文个数.

算法运行结束后,聚类结果如图 3 所示:

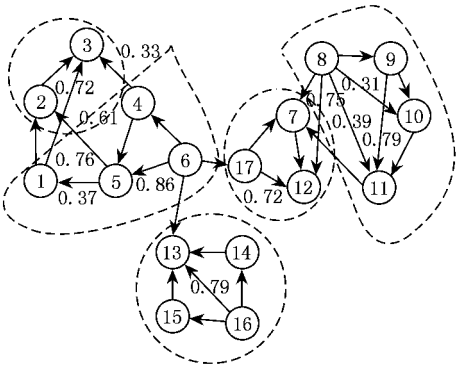

图 3 BHC 算法的聚类结果

3.2 *K*-BHC 算法

由于 BHC 聚类结果不准确,因此我们作出几点修改,使它能更贴近理想的聚类结果. 该改进后的算法称 *K*-BHC 算法.

1) 改进思想

因为我们研究的引文推荐集合中不可能包含所有的引文,而事实上也没有必要包含所有的引文. 但我们希望引文推荐集合中包含的引文质量和参考价值很高. 通常,科学文献是有衰老过程的,发表时间很长的论文的参考价值会很低. 而发表了很长时间的引文即使被引用,通常也只是引用其中的一些概念,因此,引文推荐集合中不应该包含这样的引文.

以上这种情况会使一些引文引用的参考文献不在引文推荐集合中而导致这些引文的引用边数量减少. 我们是根据引用边寻找相关性强的引文的,而引用边数量较少的引文无法准确计算它与其他引文之间的相关程度. 为此,我们引入引用边稀少引文概念.

定义 1. 若一篇引文 60%以上的引用引文不在推荐集合中. 在这里我们称之为引用边稀少引文(rare reference edge citation, REC).

REC 是无法通过计算引用边的长度进行准确聚类的,为了解决这个问题,我们引入由 small 提出来的同被引概念[11]. 同被引是指两篇论文共同被后来的一篇或多篇论文引用的现象. 同被引的频次越高两篇论文的相关性越强. 利用这个概念我们将聚类算法进行修改. 对于 REC,我们利用同被引频次的概念对其进行聚类.

针对以上思想,我们对算法 BHC 作出如下修改:在初始化时,将每个引文划分为一个类. 将 REC 依照同被引频次聚类,同被引频次高的 REC 聚为一类. 引入 REC 规则后的 BHC 算法聚类结果如图 4 所示:

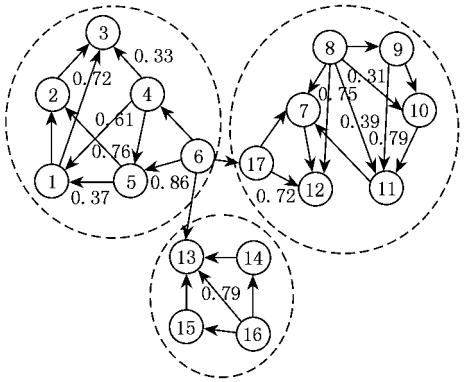

图 4 引入 REC 规则后 BHC 算法聚类结果

在图 4 的聚类结果中,因为 6 与 17 虽然满足平均距离的限制,但是与本类中联系的边很少. 如 6 只有 4,5 两个点有引用关系. 我们希望最后生成的聚

类满足的条件是类中各个点之间的联系都非常紧密.也就是说我们希望得到的聚类在图中表现为稠密子图.要想满足这一点只用平均距离进行限制是不可能的.我们在限制平均距离的同时添加一条新的限制,新加入的点必须至少与类中 K 个点有联系(存在引用边).在这里我们将 K 设置为聚类中点的数量的 1/2.

进一步,将 BHC 算法进行改进,提出 K-BHC 算法.K-BHC 是基于 BHC 算法提出的一个适合对引用图进行聚类的算法,并根据聚类的结果对算法作出一些改进,使之能更贴近理想结果.

2) K-BHC 算法

K-BHC 算法中的 3 个部分描述如下.

① 初始化.将 REC 依照同被引频次聚类,剩下的引文每个引文作为一个类.

② 计算引文之间的距离,将同时满足平均聚类与最小 K 联系的引文聚为一类.

③ 持续 2)直到稳定,即没有新产生的类.

算法 1 描述了聚类的过程.算法的输入是引文引用图,算法的输出是图中点的聚类.行①～⑦为算法的初始化过程.首先找出属于 REC 的引文,将引用频次高的 REC 聚为一 类,其他的每个引文单独作为一类.从行⑧开始为迭代阶段.将同时满足平均距离及至少与 K 个点有联系的两个类聚成一个新的类.当没有新类产生时算法结束.

算法 1. K-BHC 算法.

```
Input:G(V,E)
Output: clusters of vertex u
①   for each vertex u∈V[G]
②   if(REC(u))
③   for each vertex t∈V[G]
④     if(REC(t) && hsmall(u,t))
⑤       new C=combine(u,t)
⑥   for each vertex u∈V[G]-t∈C
⑦     do C(u)←vertex u
⑧   for i=1 to N
⑨     for j=i to N
⑩       temp=Distance(C(i),C(j))
⑪       if(temp<mindistance)
⑫         temp=mindistance
⑬           N1←i
⑭           N2←j
⑮ if ( min < threshold && Cit ( C ( N1 ), C(N2))>K)
⑯         new C=combine(C(N1),C(N2))
⑰   while(have new node)
⑱   for i=1 to M
⑲     temp=Distance(new C,C(k))
⑳       if(temp<mindistance)
㉑         temp=mindistance
㉒  if ( min < threshold&&Cit ( C ( N1 ), C (N2))>K)
㉓         new C=combine(new node,C(k))
```

图 5 为 K-BHC 的初始化结果,图 6 为 K-BHC 的聚类结果:

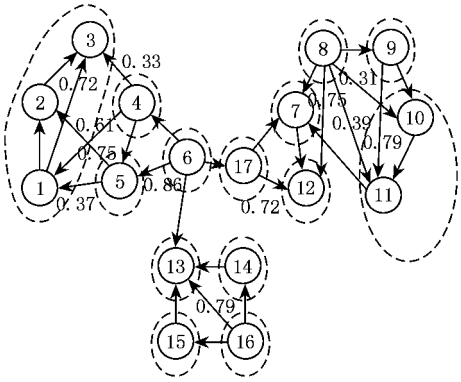

图 5　K-BHC 的初始化结果

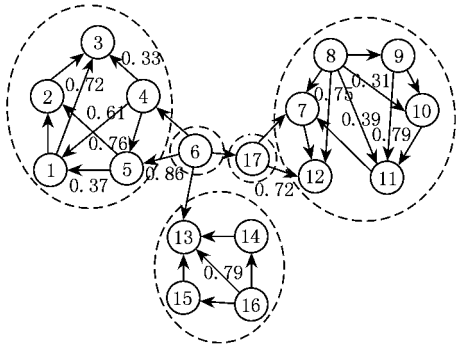

图 6　K-BHC 的聚类结果

4　产生相关引文

本节我们将利用第 3 节完成的聚类结果来产生目标引文的相关引文.

4.1　相关引文

我们在产生相关引文时会遇到两种情况:1)目标引文在一个类中(如图 5 中的引文 4),那么这个聚类中的引文就是 a 的相关引文.这种情况是比较简单的,由于一个类中的引文相互之间的关联性很

强,所以直接将这个类里的引文作为相关引文既可.我们重点研究的是第2)种情况.2)如果 a 不在一个聚类中(如图6中的引文6)那么需要把它附近的聚类中的引文作为它的相关引文.通常这样的聚类不止一个.这就涉及到采用哪个聚类的问题.

4.2 聚类评价

在前面描述的问题中,我们希望根据 a 推荐的是这样的一个聚类,离 a 的距离越近越好,类的质量越高越好.所以我们的评价得分应该与聚类和 a 之间的距离成反比,聚类的质量成正比.

聚类和 a 之间的距离只需要在图上计算一个点与一个聚类的平均距离即可.下面我们讨论如何评价一个聚类的好坏.我们从用户和聚类本身两个方面来评价希望通过用户评分评价一个项目的好坏是不现实的,因为只有极少数的用户才会对项目作出评价.因此,我们使用用户的行为代替用户的评分来对项目作出评价.首先,我们定义几个有意义的用户的行为:a 下载了推荐的文献;b 将推荐的文献保存了书签;c 浏览推荐文献页面超过一定时间;d 浏览推荐文献少于一定时间;e 没有对推荐文献作任何事.这5个项目量化成评分分别是4,3,2,1,0分.

我们希望根据与当前用户行为相似的用户的评分来推测当前用户对当前聚类的评分(之前我们已将用户的操作量化为评分),下面我们将用户对引文的操作称为评分).我们采用用户对聚类中引文的平均评分作为聚类的评分.

首先我们计算两个用户之间的相似程度.我们采用比较常见的余弦距离计算两个用户之间的相似程度:

$$Sim(a,b)=\frac{\sum_{C\in C_a\cap C_b}(S_{a,C}-S_a)(S_{b,C}-S_b)}{\sqrt{\sum_{C\in C_a\cap C_b}(S_{a,C}-S_a)^2}\cdot\sqrt{\sum_{C\in C_a\cap C_b}(S_{b,C}-S_b)^2}},$$

其中,$Sim(a,b)$ 表示用户 a,b 之间的相似程度;C_a 和 C_b 分别表示用户 a 和用户 b 评过分的聚类的集合;$S_{a,C}$ 和 $S_{b,C}$ 分别表示用户 a 和用户 b 对聚类 C 的评分;$S_{a,C}$ 和 $S_{b,C}$ 分别表示用户 a 和用户 b 对聚类 C 的平均评分.

我们将满足 $Sim(a,b)>\alpha$ 的用户 b 的集合作为 a 的相似用户的集合 $M(a)$. α 的具体设定根据实际需要相似用户的数量大小确定.

下面我们根据相似用户推测当前用户对聚类的评分.

$$Score_u(C,a)=\frac{\sum_{b\in M(a)}Sim(a,b)(S_{b,C}-S_b)}{\sum_{b\in M(a)}Sim(a,b)},$$

$Score_u(C,a)$ 表示用户 a 对于聚类 C 的预测得分.

评价一篇引文的好坏,引用次数是一个很重要的标准,通常被引用次数高的引文质量较高,我们将这一观点引入到评价聚类质量中,将一个聚类的平均被引用次数(图中点的平均入度)作为评价聚类质量的标准.

$$Score_i(C,t)=\frac{\sum_{i\in C}d_{in}(i)}{n\cdot Distance(t,C)},$$

其中,$Scorei(C,t)$ 表示类 C 对于引文 t 的得分. $d_{in}(i)$ 表示引文 i 的入度,n 表示类 C 中引文的数量.

最终我们综合考虑用户和项目本身两方面,产生对类的质量的评价.

$$Score(C,a,t)=\frac{Score_i(C,a)+Score_i(C,t)}{2},\tag{1}$$

其中 $Score(C,a,t)$ 表示类 C 对于目标引文 a、当前用户 u 的得分.

我们的基于多因素的引文推荐算法(MFR)如下:

1) 产生引文引用图;
2) 计算引文图边的权值;
3) 按照 K-BHC 算法进行聚类;
4) 根据式(1)计算出得分高的类推荐给用户.

5 实验与分析

5.1 数据集

为了验证算法的性能,我们各引文数据源网站下载了1 000篇引文作为推荐集合.在实验室的引文搜索软件实现了推荐功能.选取80个评价了10篇以上引文的用户作为训练集合,20个用户作为验证集合.

5.2 度 量

本文的实验采用平均绝对偏差(mean absoloute error,MAE)作为度量算法好坏的标准. MAE 越小推荐质量越高[12].

5.3 算法比较

为了验证本文提出的多因素推荐算法(MFR)的有效性,将与传统的基于用户的推荐算法(UCFR)以及文献[12]提出的基于项目的推荐算法(ICFR)

进行了比较.

我们分两种情况进行比较,保持项目数量的不变,增加用户的数量以及保持用户的数量的不变而增加项目的数量. 图 7 和图 8 分别为两种情况下的实验结果:

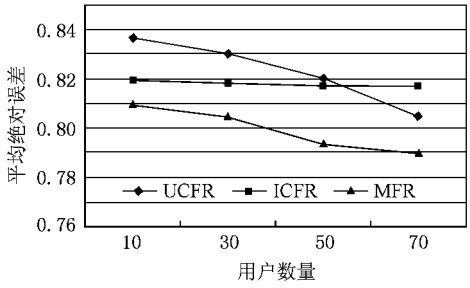

图 7　项目数量不变的实验结果

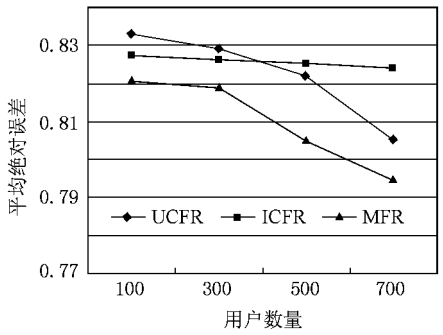

图 8　用户数量不变的实验结果

图 7 为保持项目数量不变逐渐增加用户数量的实验结果. 坐标的横轴为用户的数量,坐标的纵轴平均绝对误差,平均绝对误差的数值越小说明算法的效果越好.

在初始阶段,由于用户数量很少,所以基于用户的推荐算法(UCFR)性能最低,随着用户数量的增加,基于用户的推荐算法准确率逐渐升高. 基于项目的推荐算法(ICFR)基本不受用户数量增加的影响. 本文提出的推荐算法(MFR)在初期,依靠引文自身的属性计算相似程度,不受用户数量少的影响. 随着用户数量的增加,根据用户的行为对计算结果进行调整,所以会随着用户数量的增加提高准确率.

图 8 为保持用户数量不变逐渐增加用户数量的实验结果. 坐标的横轴为用户的数量,坐标的纵轴平均绝对误差,平均绝对误差的数值越小说明算法的效果越好. 在初始阶段,由于项目数量很少,所以基于项目的推荐算法(UCFR)性能最低,随着项目数量的增加,基于项目的推荐算法准确率逐渐升高. 基于用户的推荐算法(ICFR)基本不受项目数量增加的影响. 本文提出的推荐算法(MFR)在初期,依靠用户的行为修正项目之间相似度的计算结果. 随着项目的增加,引用图会变的更加有规律性,聚类的效果会更好,所以性能会随着项目数量的增加而增加.

由图 7 和 8 可以看出,本文提出的算法(MFR)在用户或项目较少的情况下,仍然可以作出推荐,不存在冷开始问题. 且随着用户数量的增加,根据用户的行为,可以修正计算结果,吸取了基于用户的算法的优点.

6　总　　结

本文提出的推荐算法在计算引文相似度时,并不依赖用户的评分,而是根据引文本身的属性进行计算,克服了冷开始的问题. 考虑了多种因素对引文之间相关性的影响,所以计算相似性的准确率要比单纯考虑语义相似度要高.

随着用户的增多,可以根据用户的行为对计算结果进行调整,使结果更靠近正确结果.

本文寻找的是关联程度较高的引文的集合,而不是独立的找出一篇篇引文,这样发现的引文关联程度更高.

参　考　文　献

[1] Vazquez A. Statistics of citation network. 2001. [2011-05-10]. http://www. sns. ias. edu/~ vazquez/publications/citation. pdf

[2] Shardanand U, Maes P. Social information filtering: Algorithm for automating word of mouth //Proc of CHI 1995. New York: ACM, 1995: 210-217

[3] Sarwar B, Karypis G, Konstan J. Item-based collaborative filtering recommendation algorithm //Proc of WWW 2001. New York: ACM, 2001: 285-295

[4] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 2003, 7(1): 76-80

[5] Sun Xiaohua, Kong Fansheng, Ye Song. A comparison of several algorithms for collaborative filtering in startup stage //Proc of Networking Sensing and Control 2005. Berlin: Springer, 2005: 25-28

[6] Gong Songjie, Ye Hongwu. Joining user clustering and item based collaborative filtering in personalized recommendation services //Proc of IIS 2009. Piscataway, NJ: IEEE, 2009: 149-151

[7] Bhattacharya I, Getoor L. Iterative record linkage for cleaning and integration //Proc of DMKD 2004. New York: ACM, 2004: 11-18

[8] Basu C, Hirsh H, Cohen W, et al. Technical paper recommendation: A study in combining multiple information sources. Artifical Intelligence Research, 2001, 14(1): 231-252

[9] He Qi, Pei Jian, Kifer D, et al. Context-aware citation recommendation //Proc of WWW 2010. New York: ACM, 2010: 421-430

[10] Kessler M M. Bibliographic coupling between scientific papers. Arnerican Documentation, 1963, 14(1): 10-25

[11] Small H. Co-citation in the scientific literature: A new measure of the relationship between two documents. American Society for Information Science, 1973, 24(4): 265-269

[12] Wang Jun P. de Vries A, Reinders M. Unifying user-based and item-based collaborative filtering approaches by similarity fusion //Proc of SIGIR 2006. New York: ACM, 2006: 501-508

石 杰 男,1985年生,硕士研究生,主要研究方向为Web数据管理.

申德荣 女,1964年生,教授,博士生导师,主要研究方向为Web数据管理,网格计算及数据空间等(shenderong@ise.neu.edu.cn).

聂铁铮 男,1980年生,博士,讲师,主要研究方向为数据集成(nietiezheng@ise.neu.edu.cn).

寇 月 女,1980年生,博士,讲师,主要研究方向为实体识别(kouyue@ise.neu.edu.cn).

于 戈 男,1962年生,教授,博士生导师,主要研究方向为数据库、数据挖掘、数据仓库等(yuge@ise.neu.edu.cn).